INTERNATIONAL HEALTH TERMINOLOGY
STANDARDS DEVELOPMENT ORGANISATION

SNOMED CT® Technical Implementation Guide

January 2015 International Release

(US English)

This document may have been updated since you received it.
The latest versions of IHTSDO documents are available online.
Latest PDF version: www.snomed.org/tig.pdf
Latest web browsable version: www.snomed.org/tig
Directory of available documents: www.snomed.org/doc

Contents

Chapter 1

1 Preface

The purpose of this guide is to meet the needs of people who require an authoritative point of technical reference and advice to support their involvement in designing, developing, acquiring or deploying software applications that uses *SNOMED Clinical Terms*.

SNOMED Clinical Terms continues to evolve to further enhance its ability to represent clinical information and there is a growing body of knowledge about how best to use it. As a result future releases of the guide may include additional detail, revised advice and notes on significant changes to specifications.

This guide is available in two forms:

- **WebHelp (HTML)**: (file suffix .html)
  - A hyper-linked version viewable in a standard web-browser.
  - This version includes searching and glossary lookups.
  - The web-based version is most effectively when used online. Some features may not work on a local version of this resource.
- **Adobe Acrobat**: (file suffix .pdf)
  - A browsable and printable version arranged for page layout rather than in separate topics.
  - The text content is identical to the HTML version but there are some difference to navigation and cross references resulting from page oriented formatting.
  - This version is searchable.

A version of each of the above is available configured for the US English and GB English. Note that the PDF versions are formatted for different paper sizes (US - Letter, GB - A4).

1.1 Notation used in this document

The following notation is used in this User Guide to represent key types of *SNOMED CT* information:

SNOMED CT concept names are generally represented using the *Fully Specified Name* in mixed case formatted as in the following example:

Example: | Peribronchial pneumonia (disorder) |

SNOMED CT attribute names are preceded and followed by a vertical bar. In some cases, to make them stand out they are presented in all capital letters as in the following example, though this is not to be considered a standard form for rendering *attribute names*:

Example: | FINDING SITE |

1.2 Document Properties

Table 1:

Title:	*SNOMED CT* Technical Implementation Guide
Date & **Version**	See cover for version and date
Creating Author:	David Markwell
Subject:	SNOMED CT, Technical, Implementation Guide

Table 2: Amendment History

Version	Date	Editor	Comments
1.00	2002-02-31		Initial draft of TIG provided with first *SNOMED CT release* published by College of American Pathologists (CAP).
2 to 5	2002-07-31 to 2007-01-31		Updates published by CAP on a six-monthly or annual basis with SNOMED CT International Release.
6 to 11	2007-07-31 to 2011-07-31		Updates published by IHTSDO on a six-monthly or annual basis with SNOMED CT International Release. Including move to online HTML version.
12 to 15	2012-01-31 to 2013-07-31	DMA	Regular six-monthly updates including introduction of Release Format 2 and gradual removal of Release Format 1 specific material to a separate RF1 Guide. During this period changes to the documents were substantially informed by comments and corrections posted to the IHTSDO Document Issue Tracker.
16.00	2014-01-31	DMA	Further updates based on tracker comments. Update of concept model section to align more closely with Editorial Guide. Further major updates to remove some residual RF1 based language and guidance to increase the clarity and consistency of RF2 guidance.

1.3 Status

This guide contains parts and sections which differ in terms of the authority and status of their content. Each section of the guide is marked to indicate its publication type and status using the symbols shown in *Table 3* and *Table 4*.

Table 3: Document Types

Type Name and *Description*	Draft	Review	Current
Standard A document or other resource that is intended to be authoritative. This includes specifications of *SNOMED CT* content and *release files*. Normative requirements for particular functions are also standards.			
Guidance A document or other resource that is intended to provide advice or suggest possible approaches to particular requirement or subject area.			

Table 4: Document Status

Status Name and *Description*	Standard	Guidance
Current Indicates that the document or resource is considered to be up-to-date and complete for the current release of *SNOMED CT* (indicated by an explicitly stated version date or by the publication date).		
Review Indicates that the document or resource has been released for review and comments from *SNOMED CT* users and other stakeholders. It is intended to be complete but has not been formally approved as a final version.		
Draft Indicates that the document or resource is a draft version. It may be incomplete and has not been approved in a final version.		

This edition of the document is configured to use US English .

The PDF version of this draft is formatted to be printed on US Letter paper.

Note: This is one of a several large documents that are regularly revised by the *IHTSDO*. Therefore, for the sake of the environment, please think carefully before deciding to print the entire document.

1.4 Referencing and Commenting

This document contains a way to reference topics a way that is not dependent on changes to the structure of the document as new versions are released including additional topics. These references are web addresses that will point to the latest version of and topic in the document.

If you are using the PDF version of the document there are three icons to the right of each title which provide useful information and relevant links.

- The icon indicates the status of the topic (see *Status*).
- The icon provides a link to the web address to access and reference this topic online. Please use this reference to identify or share references to the topic as section and page numbers change between versions.
- The icon links directly to a page where you can submit comments or report errors about this topic. The comment tracker is an online resource that requires you to login to an IHTSDO CollabNet account. If you do not have an account, there is an option to create an account available on the login page.

If you are using the online web version of this document then there is a single bookmark icon which, when clicked, opens a small form with an easy copy and paste option for access to the topic reference and button to click to take you direct to the comment tracker.

1.5 Additional information

Further information about *SNOMED CT* is available by contacting *IHTSDO*:

IHTSDO Contact Details:

Web:

- *www.ihtsdo.org*

Email:

- *support@ihtsdo.org*

Address:

- IHTSDO
- Gammeltorv 4, 1.
- 1457 Copenhagen K
- Denmark
-
- Tel: +45 3644 8736
- Fax: +45 4444 8736

1.6 Inventory of Documentation

The following *SNOMED CT* documentation is made available to accompany the *International Release* of *SNOMED CT* from the International Health Terminology Standards Development Organization (*IHTSDO*). In the following listing hyperlinks are provided which will be maintained to point to the latest version of each of these documents.

- A list of documents, including a wider range of versions, is available from: *www.snomed.org/doc*.

***SNOMED CT* Starter Guide**

- On line HTML version: - *not currently available in HTML - please download one of the PDF versions*
- PDF version US English Letter page size: *www.snomed.org/snomedct_starter.pdf*
- PDF version UK English A4 page size: *www.snomed.org/snomedct_starter_gb.pdf*

The Starter Guide is the ideal place to start learning about SNOMED CT. It covers a wide range of topics related to SNOMED CT at a fairly high-level but with sufficient detail to be useful to most readers.

Note:

The Starter Guide replaces the previously published User Guide.

***SNOMED CT* Technical Implementation Guide (TIG)**

- On line HTML version: *www.snomed.org/tig*
- PDF version US English Letter page size: *www.snomed.org/tig.pdf*
- PDF version UK English A4 page size: *www.snomed.org/tig_gb.pdf*

The TIG is intended for *SNOMED CT* implementers, such as software designers. The TIG assumes information technology and software development experience. Clinical knowledge is not required, although some background is helpful to understand the application context and needs.

The TIG contains guidelines and advice about the design of applications using *SNOMED CT*, and covers topics such as *Terminology services*, entering and storing information, and migration of legacy information.

***SNOMED CT* Editorial Guide**

- On line HTML version: *www.snomed.org/eg*
- PDF version US English Letter page size: *www.snomed.org/eg.pdf*
- PDF version UK English A4 page size: *www.snomed.org/eg_gb.pdf*

The Editorial Guide is intended for clinical personnel, business directors, software product managers, and project leaders; information technology experience, though not necessary, can be helpful.

The Editorial Guide is intended to explain *SNOMED CT*'s capabilities and uses from a content perspective. It explains the content and *concept model*, and the principles used to edit the terminology.

***IHTSDO* Glossary (DRAFT)**

- On line HTML version: *www.snomed.org/glossary*
- PDF version US English Letter page size: *www.snomed.org/glossary.pdf*
- PDF version UK English A4 page size: *www.snomed.org/glossary_gb.pdf*

***SNOMED CT* ICD-10 Mapping Specification**

- PDF version US English Letter page size: *www.snomed.org/icd10map.pdf*

This document describes the mapping use cases and technical procedures applied to the co-development of a *SNOMED CT* to ICD-10 map by *IHTSDO* and the World Health Organization (WHO). It provides guidance on the intended purposes and practical use of the mapping files produced from this development.

SNOMED CT Release Format 1 **Guide**

- On line HTML version: *www.snomed.org/rf1*
- PDF version US English Letter page size: *www.snomed.org/rf1.pdf*
- PDF version UK English A4 page size: *www.snomed.org/rf1_gb.pdf*

The RF1 Guide provides technical information relevant to those using the original *SNOMED CT Release Format*. Although this format was replaced by RF2 in January 2012, the old format is being maintained for a transitional period.

SNOMED CT **Non-Human** ***Refset*** **Guide**

- PDF version US English Letter page size: *www.snomed.org/guide/non_human_rs.pdf*

A guide to use of the "Non-Human" Simple *Reference Set* that contains *concepts* and terms that are only used in veterinary medicine. This guide applies to content that is dated July 2013 or before. As of the January 2014 International Release, non-human concepts have been moved to an extension.

SNOMED CT **Developer Toolkit Guide**

- PDF version US English Letter page size: *www.snomed.org/guide/toolkit.pdf*

A guide to use of value-added files and scripts that are provided as a toolkit available as part of the *SNOMED CT International Release*.

Additional Documentation: The following materials previously published in separate documents are now integrated as part of the Technical Implementation Guide.

- *Technical Reference Guide*
- *Namespace Identifier Guide*
- *Namespace Identifier Registry*
- *File Naming Convention*
- *RF2 Data Structures Specification*
- *RF2 Reference Set Specifications*
- *RF2 Update Guide*
- *Stated Relationships Guide*
- *Canonical Table Guide* (previously included in RF1)

1.7 Where is the Glossary?

Some versions of documents may contain a glossary section. However, we are also developing a separate *IHTSDO* Glossary document which is currently available in a draft form. The intention is to move toward using this single common resource make it easier to ensure consistency across the *IHTSDO* community.

The current version of the *IHTSDO* Glossary is available as follows:

- On line HTML version: *www.ihtsdo.org/glossary*
- PDF version US English Letter page size: *www.ihtsdo.org/glossary.pdf*
- PDF version UK English A4 page size: *www.ihtsdo.org/glossary_gb.pdf*
- You can create links that query the glossary use the following web address pattern "www.ihtsdo.org/define/*word or phrase*". The following examples can be tested and you can include these types of reference in your documents to make it easy to refer to the *IHTSDO* glossary definitions:
  - *www.ihtsdo.org/define/ihtsdo*
  - *www.ihtsdo.org/define/snomed ct*
  - *www.ihtsdo.org/define/affiliate licence*

1.8 Copyright Notice

Chapter 2

2 Overview of the guide

The overall structure of the guide is not intended as a suggested reading order but to provide a predictable location for each broad category of information. Thus as the guidance is extended and revised new sections will appear within the relevant locations rather than as separate documents.

The guide is divided into several parts each of which focuses on a particular aspect of technical implementation. The sections are interdependent and are extensively cross-linked. These links are used to avoid duplication and aid consistent maintenance.

- ***Structure and Content Guide (4)*** :
  - This part of the guide provides reference material on the technical design of *SNOMED CT*. These design features provide the foundation for the services described in subsequent parts of the guide.
  - The design features include: *SNOMED CT components* (*concepts*, *description* and *relationships*), extensibility mechanisms (*reference sets*), content *extensions* and *expressions* that are used to represent information within a electronic record.
- ***Release File Specifications (5)*** :
  - This part of the guide includes detail *Descriptions* of the files used to distribute *SNOMED CT* content to licensees;
  - The specification are an important source of technical reference for those developing and maintaining applications that provide access to *SNOMED CT* content.
- ***Concept Model Guide (6)*** :
  - This part of the guide describes the ways in which technical design of *SNOMED CT* is populated with content. The *Concept Model* specifies the main hierarchies in which *concepts* are arranged and the types of the *relationships* that are permitted between them.
  - This *Concept Model* is directly relevant to implementation because it determines the types of clinical ideas that can be expressed using *SNOMED CT* and the ways in which these ideas can be refined to represent more detailed information.
- *Terminology Services Guide (7)* :
  - This part of the guide describes the types of services required to access and make use of *SNOMED CT*. It also provides practical advice on effective ways to deliver these services based on practice experience.
  - These services include: importing distribution files, determining the *status* and properties of selected components, searching for *terms*, navigating hierarchies, testing and using *relationships* between *concepts*, working with references sets to determine *language* acceptability, membership of *value sets*, *maps* to other classification and additional *annotations* and metadata.
- ***Record Services Guide (8)*** :
  - This part of the guide describes the types of services required to use *SNOMED CT*, to represent instances of clinical information in *electronic health records*, knowledge resources, decision support algorithms and data retrieval specifications.

- These services include, entry of *expressions* (including *postcoordinated refinements*), storage of *expressions*, communication and selective retrieval of information that uses *SNOMED CT expressions* to represent clinical ideas.
- As part of the consideration of storage, communication and retrieval, this part of the guide also discusses the integration of the terminology with a well-designed information model. It is now widely recognized that this is crucial element in design and development of a *SNOMED CT enabled application*.
- *Record services* are dependent on *Terminology services* and these two sets of services may be tightly integrated. Alternatively, an application that delivers *record services* may use a third party *terminology server* to reduce the required development.

- ***Change Management Guide (9)*** :
  - This part of the guide addresses requirements that arise from changes to the content, structure and use of *SNOMED CT*.
  - The first significant change management challenge relates to *migration* from other coding schemes or from a less structured electronic record system. Decisions must be made about retaining or converting records, queries and protocols originally created using a terminology other than *SNOMED CT*.
  - Each release of *SNOMED CT* introduces some changes to content. From time to time there will also be changes designed to increase the expressivity of the *Concept Model*. Occasionally there may also be additional technical artifacts or specification developed to meet emerging requirements.
  - As systems evolve and as the content and structure of *SNOMED CT* are enhanced there is a continuing requirement to address to manage changes smoothly and without loss of information or functionality.
- ***Extension Services Guide (10)*** :
  - This part of the guide describes additional services which some advanced users or implementers may require to allow them to create or maintain *Extensions* for use in a particular country, organization or specialty.
  - The most common of these requirements will be to support the creation and maintenance of specialized *Reference sets*. Uses for *Reference Sets* include representation of *value sets*, marking *descriptions* to indicate acceptability of *terms* in a specific *language* or specialty, alternative hierarachies, *cross mapping* to classifications and *annotations*.

2.1 Who Should Read This Guide?

The guide can be used in various ways to assist the design, evaluation, operational implementation and use of various types of software applications that use *SNOMED CT*. The intended audience includes systems developers, health informatics specialists, purchasers, and system integrators.

2.1.1 Software designers and developers

- Software designers and developers should use this guide:
  - To enhance their technical understanding of *SNOMED CT* and the value it offers to their applications;
  - As a point of reference when designing a *SNOMED CT enabled application* and when planning and undertaking the required development.
- Designers and developers of fully integrated applications should use the guide:
  - As a checklist of *SNOMED CT* services necessary to meet the needs of their users;
  - For advice on how to implement the required services in ways that make the best use of *SNOMED CT* and which avoid known pitfalls.
- Designers and developers of *terminology servers* should use the guide:

  - As a checklist when deciding which *SNOMED CT* services their server should offer;
  - For advice on ways to implement the required services in ways that make the best use of *SNOMED CT* and avoid known pitfalls;
  - As a point of reference when describing the functionality of their server.

- Designers and developers of applications that use *Terminology services* should use the guide:
  - As a checklist of *SNOMED CT* services necessary to meet the needs of their users;
  - To assist consideration of whether to use a *terminology server*;
  - As a point of reference when reviewing the functionality of *terminology servers*.

2.1.2 Health informatics specialists, analysts, purchasers and integrators

- Health informatics specialists, analysts, purchasers and integrators should use this guide:
  - To enhance their technical understanding of *SNOMED CT* and the value it offers to their organization ;
  - As a point of reference when specifying, procuring and evaluating *SNOMED CT enabled applications*.
- Health informatics specialists analyzing the needs of users and organizations should use this guide:
  - As a checklist of *SNOMED CT* services necessary to meet the needs of their users;
  - For advice on known pitfalls when implementing clinical terminologies;
  - To assist decisions on technical approaches to design and implementation of applications that use *SNOMED CT*.
- Purchasers of healthcare information systems should use this guide:
  - As a checklist when specifying procurement requirements for applications that use *SNOMED CT*;
  - As a starting point for the evaluation of the *SNOMED CT* related technical features of the available systems.
- Healthcare information systems integrators should use this guide:
  - As a checklist for confirming the claimed functionality of *SNOMED CT enabled applications*;
  - For advice on alternative approaches to integration of *SNOMED CT* related services into a wider information system.
- Information systems departments and project teams should use this guide:
  - As a checklist for the *SNOMED CT* related functionality needed to meet the requirements of their users;
  - For advice on alternative approaches to delivery and maintenance of *SNOMED CT* related functionality as part of an operational information system.

2.2 Important Notices

Note: The *IHTSDO* supplies *SNOMED CT* as a set of *release files* that are designed to be loaded into healthcare software applications such as *Electronic Health Records*. This guide describes services that should be provided by software applications that implement *SNOMED CT*.

Note: The *IHTSDO* does not create or market healthcare software applications but seeks to promote implementation and innovation by promoting a market place in which *SNOMED CT* is equally accessible to all software developers, vendors and health service providers.

Note: This guide refers to files that are included in the *International Release* of *SNOMED CT* provided to licensees by the *IHTSDO*. It also refers to additional files that are included in *SNOMED CT Extensions* provided by *IHTSDO Members* and *Affiliates*. Details of the licensing arrangements for *SNOMED CT* and contact details for *IHTSDO Members* are available from the *IHTSDO* web site:

- *www.ihtsdo.org*

2.3 Additional information and feedback

Further information about *SNOMED CT* is available on the Internet at:

- *www.ihtsdo.org*

Please send feedback by email to:

- *support@ihtsdo.org*

or contact the International Health Terminology Standards Development Organisation at:

- IHTSDO
- Gammeltorv 4, 1.
- 1457 Copenhagen K
- Denmark
-
- Tel: +45 3644 8736
- Fax: +45 4444 8736

Chapter 3

3 SNOMED CT implementation

This part of the guide introduces the rationale for implementing *SNOMED CT*. It identifies some of the types of software application that benefit from the features of *SNOMED CT*. It sets out some broad parameters that determine the extent to which an application can make use of particular aspects of *SNOMED CT* and outlines some approaches to delivering the required services.

3.1 Motivation for Implementation

SNOMED Clinical Terms (*SNOMED CT*) is widely recognized as the leading global clinical terminology for use in *electronic health records*. It is maintained and developed by an International body (the *IHTSDO*) which has a growing community of Members and Affiliates. It is available free for use in *IHTSDO Member* countries and can also be used in other countries based on openly published licensing terms that are designed to be affordable. *IHTSDO* policies allow for the open involvement of its Members and *Affiliate Licensees* in the development of content and the design of future enhancements.

The features of *SNOMED CT* include:

- A broad scope that covers most of the clinical *concepts* used in patient centered clinical records;
- Ability to express different levels of clinical detail in patient record entries by using *expressions* containing one or more *concept identifiers*;
- *Relationships* between *concepts* that enable consistent retrieval of a common form of clinical information for many different purposes;
- Extensible design allowing graceful, evolutionary enhancement and addition of national, local or specialty content within a coherent standard structure;
- A *reference set* mechanism to support representation of *language / dialect* variants, *value sets*, alternative hierarchies and mapping to classifications;
- *Component* permanence with history tracking;
- Good compliance with the essential features for future clinical terminologies as identified by *JJ Cimino in his peer acclaimed 1998 paper*.

SNOMED CT is designed to enable effective representation of clinical information in *electronic health records*. While there are other potential uses for *SNOMED CT*, the potential benefits are greatest where it is implemented as a part of a *Clinical Information System* centered on the delivery of health care services to individuals and populations.

The benefits actually realized by implementation depend on the technical design of applications and the way they integrate *SNOMED CT* with other essential elements. These technical issues are addressed in this guide. Another critical success factor is a process for managing implementation across an organization , region or country. Although the guide does not address broader issues of operational implementation within an organisation, it does provide a key source of reference for those specifying the practical details of a plan for large scale implementation of *SNOMED CT*.

3.1.1 Benefits for electronic health records

Implementation of *SNOMED CT*, as part of a well-designed *Clinical Information System*, is the key to unlock many of the potential benefits of *electronic health records*.

SNOMED CT enables consistent representation of clinical information within *electronic health records*. Its content and design allow most types of clinical information to be represented at levels of detail appropriate to a wide range of different use cases. The hierarchical and defining *relationships* of *SNOMED CT* facilitate effective meaning-based retrieval and reuse of this information. By using these *relationships*, a *SNOMED CT enabled application* can *query electronic health records* to extract, analyze and aggregate relevant data recorded in different settings and at different levels of detail.

Many of the benefits of *electronic health records* require an effective retrieval and reuse of clinical information. These include:

- Enhancing the care of individual patients:
  - Display of appropriate information to enable clinical staff to assess the condition and needs of patients;
  - Decision support tools that help to guide safe, appropriate and effective patient care;
  - Communicating, sharing and maintaining information in ways that enable different members of the health care team to access and use relevant information collected at different places and times.
- Enhancing the care of populations of patients:
  - Epidemiology monitoring and reporting;
  - Research into the causes of diseases;
  - Research into the effectiveness of different approaches to disease management and treatment.
- Supporting cost-effective delivery of care:
  - Using decision support to minimize the risk of costly errors in treatment;
  - Reducing duplication of investigation and interventions through effective access to shared information about the patient;
  - Auditing the delivery of clinical services; with more opportunity to analyze outliers and exceptions in the pattern of care delivery;
  - Planning future service delivery based on emerging health trends, perceived priorities and changes in clinical understanding.

Delivering these benefits depends on consistent representation of the various types of information that are represented in a health record. It must be possible to represent this information at different levels of detail and it must be possible to *query* this information from various perspectives and at different levels of detail. To meet these requirements *electronic health records* need a well-maintained terminology that meets the criteria specified in *Desiderata for Controlled Medical Vocabularies in the Twenty-First Century* (*Cimino JJ in Methods Inf Med. 1998 Nov;37(4-5):394-403*). *SNOMED CT* addresses these requirements and additional practical requirements for an implementable, globally applicable but locally extensible, multilingual solution.

3.1.2 Benefits for knowledge representation

Implementation of *SNOMED CT* within a knowledge resource, such as an electronic reference book, clinical guideline, decision support protocol, facilitates effective access from, or integration with, *Clinical Information Systems*.

The use of *SNOMED CT* in *electronic health records* enables consistent processable representation of clinical information. Potential uses of this information include linkage to knowledge sources to assist its understanding and interpretation.

Developers of decision support protocols, care pathways or data analysis packages can benefit by using *SNOMED CT* to represent requirements for clinical information collection and processing. This allows direct translation of the protocol into queries that can be applied directly to a *SNOMED CT* enabled *electronic health record*.

Publishers of knowledge based resources can benefit by tagging their information using *SNOMED CT*. These tags can be used to index information by *concept* rather than by *keywords*. As a result, relevant information can be identified by users during interaction with an *electronic health record*. For example, when selecting a particular item during data entry or review potentially relevant articles can be listed and/or displayed.

SNOMED CT also offers benefits during the development of knowledge resources. Tagging information using *SNOMED CT* while authoring knowledge artifacts may identify potential ambiguities that would otherwise be overlooked.

3.1.3 Benefits of an open global approach

Implementation of *SNOMED CT* offers the benefit of a global approach to the requirements for clinical terminology.

Any country or large organization that is developing or deploying *electronic health records* needs to consider the requirements for consistent representation of clinical information. One element of the solution is usually a coding scheme, controlled vocabulary or terminology. The breadth or scope and depth of detail in clinical records means that the set of codes or *terms* required is large and grows rapidly as additional disciplines and specialties become involved. Similarly the interdependency of *terms* used in different domains leads to a significant level of complexity.

Developing and maintaining a terminology that adequately addresses clinical requirements is a substantial task. A global approach has significant benefits by enabling economies of scale for National bodies and health care service providers.

A global approach also encourages common solutions to some of the challenges posed by requirements for consistent representation of complex information. The resulting reduction in divergence provides a more secure foundation for implementers who wish to deploy their applications in many countries.

Implementing a global clinical terminology also enables applications to be deployed in other countries without needing to switch between terminologies. It also allows use of other standards and materials that incorporate or are designed for use with that terminology. The ability to integrate components and standards based on a common terminology is a major advance over solutions that depend on a local or proprietary code system.

A global clinical terminology also provides a foundation for communication and sharing of information. The information communicated may include clinical records used to support delivery of health care to a mobile population. It may also include aggregations of records used for epidemiology and multi -center research.

3.1.4 Benefits of extensibility and configurability

Implementation of *SNOMED CT* allows common approaches to be applied to extend and configure the terminology for use in a particular environment.

Most clinical *concepts* are relevant in all countries, organizations and specialties but some *concepts* are relevant only to a particular environment. *SNOMED CT* allows national, local or organizational requirements to be addressed by separately maintained *SNOMED CT Extensions*. *SNOMED CT enabled implementations* can benefit from the content in these *Extensions* without the need for any additional software development because *Extensions* have exactly the same structure as the *International Release*.

SNOMED CT covers a broad domain to depth of detail appropriate to a range of health care disciplines and clinical specialties. As a result, it has an extensive content, different parts of which are needed in particular environments. The *SNOMED CT* design includes the *Reference Set* mechanism which provides a standard way to refer to a set of *SNOMED CT components*. *Reference Sets* can be used to configure different views of *SNOMED CT* by constraining searches or representing short lists of terms for a data entry field. They can also be used to meet other requirements including checking that a *concept* id falls within a permitted set of values for a field in a data structure or message (e.g. to represent an *HL7 value set*).

- Organizations implementing *SNOMED CT* benefit from *Reference Sets* because they allow requirements for use of particular terms and *concepts* to be represented in a form that can be applied to any *SNOMED CT enabled application*. This allows *Reference Sets* to be shared throughout and between organizations , even when different software is used to meet local or departmental requirements.
- Software developers and vendors benefit because *Reference Sets* provide a common, machine processable representation of requirements for different patterns of use of *SNOMED CT*. This simplifies local configuration and enhances interoperability with other *SNOMED CT enabled applications*.

3.2 Implementation Types

SNOMED CT itself is only a part of the solution to addressing the requirements for effective electronic clinical records. A terminology on its own "does" nothing unless it is implemented as part of an application and used. Implementation of *SNOMED CT* requires software applications that exploit its features to meet the real and perceived needs of users.

The "users" of *SNOMED CT* include:

- Those who specify, commission and configure software for use in a particular clinical environment;
- End-users who enter or retrieve clinical information.

As illustrated by *Figure 1*, users experience *SNOMED CT* through application software which delivers services to access and apply *SNOMED CT*. The ways in which applications apply the features of *SNOMED CT* to address user requirements determine the extent to which the potential benefits are realized .

The following sections summarize some of the types of implementation that may be needed to meet different requirements. Some types of application do not need to support or use all *SNOMED CT* features. However, there are some overarching requirements for consistency between implementation used within a given organization , country or region. Even where requirements are limited, care should be taken to ensure that *SNOMED CT enabled applications* are aligned with good practice and with agreed policies applicable to the situations in which they are used.

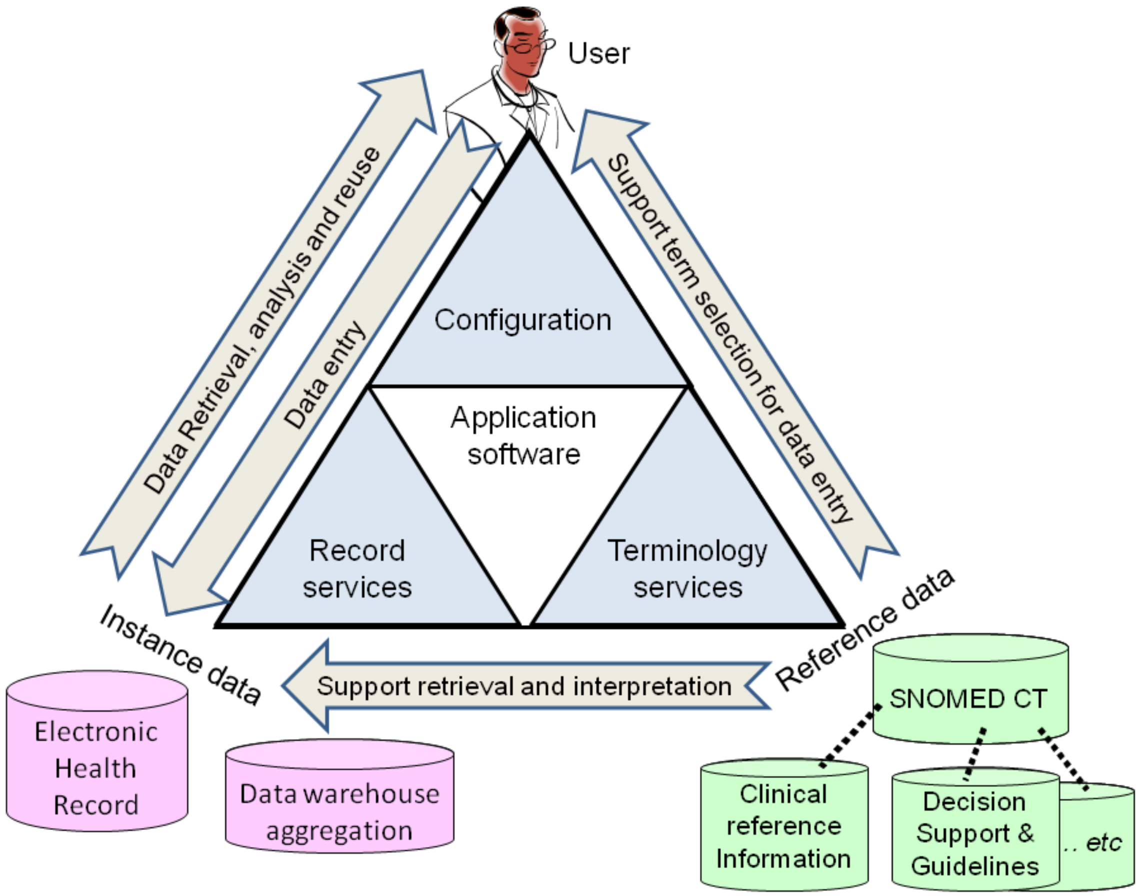


Figure 1: Relationship between users application software and SNOMED CT

3.2.1 Implementation Types - Clinical records

A *SNOMED CT* enabled clinical record application uses *SNOMED CT expressions* to represent clinical information in the records of individual patients.

Clinical record applications include specialized departmental systems, organization-wide systems and systems that integrate multiple systems to deliver a distributed *electronic health record* or a collection of widely accessible summary records.

A *SNOMED CT* enabled clinical record application needs to provide *record services* including entry, storage, retrieval and communication of *SNOMED CT expressions*. These *record services* depend on *terminology services* including the ability to search for *concepts* and to interpret stored *SNOMED CT expressions*.

A wide range of types of information can be represented at different levels of detail using *SNOMED CT expressions*. The types of information and level of detail that are used may vary depending on user requirement or may be limited by the design of the application. Differences in the required level of expressivity influence the range of *record services* that need to be supported.

The way that *SNOMED CT expressions* are represented within a record structure affects the range of services that are required to deliver the potential benefits of implementation. The value of the rich expressivity of *SNOMED CT* may be enhanced or diminished by the way the record structure relates *SNOMED CT expressions* to surrounding contextual information. For example, if a record structure permits similar or related information to be recorded in several ways a query to retrieve that information will need to consider all these possibilities. Retrieval is simpler if similar information is recorded in a consistent way - irrespective of the way it was entered. This issues are discussed in detail in the *Record Services Guide (8)*.

3.2.2 Implementation Types - Knowledge representation

A *SNOMED CT* enabled knowledge representation uses *SNOMED CT expressions* to represent or tag resources that represent clinical knowledge. Examples of resources that can be *SNOMED CT* enabled include electronic reference books, clinical guidelines, care pathways, decision support protocols and requirements for analysis and audit.

There are various ways in which *SNOMED CT expressions* can be used in a knowledge resource. These can be divided into two broad categories:

- Use of *SNOMED CT expressions* as an integral part of a structured representation of knowledge:
  - For example, a decision support rule that tests for the existence of a record of a particular type of finding represented using a *SNOMED CT expression*.
- Use of *SNOMED CT expressions* to tag or index a knowledge resource:
  - For example, a reference book in which textual *descriptions* of indications, contraindications and side effects of particular treatments are tagged with *SNOMED CT expression* that can be used to allow context-sensitive retrieval of relevant information.

There are two distinct but interrelated aspects to *SNOMED CT* knowledge representation.

- Applying *SNOMED CT expressions* to the resource:
  - The form of representation to be used must be specified in a way that takes account of the ways in which the resource is to be used and accessed.
  - The knowledge authoring environment must allow the specified representation to be applied consistently. This requires use of *Terminology services* that allow searching and selection of *concepts*. Depending on the level of detail required, there may also be a requirement to support the construction of *postcoordinated expressions*.
- Enabling appropriate access to and use of the resource:
  - The types of access required depend on the intended functionality.
  - The most basic level of functionality involves using *SNOMED CT expressions* as a *concept* -based index. By taking account of the *SNOMED CT subtype hierarchy* and defining *relationships*, a *concept* -based index can provide more relevant results than a simple *term* based search.
  - More sophisticated uses such as clinical decision support require *SNOMED CT expressions* in the knowledge resource to be used to generate queries that can be applied to information stored in an *electronic health record*.
  - The provider of a *SNOMED CT* enabled knowledge resource may provide a specification that allows software developed by other organizations to interrogate it and provide the required level of functionality. Alternatively, the knowledge authoring organization may also develop and provide software that delivers the intended functionality.

3.2.3 Implementation Types - Aggregation and analysis

SNOMED CT enabled aggregation and analysis systems use *SNOMED CT* to enable effective aggregation and analysis of information derived from clinical record systems.

SNOMED CT enables consistent processable representation of clinical information. As well as presenting opportunities for analysis of information within an individual clinical record system, this can be used to support analysis of a broader substrate of aggregated data.

There are two types of approach that be employed to enable analysis of aggregate data.

- A *SNOMED CT* enabled data warehouse:

- The content and structure of data required from individual *Clinical Information Systems* is specified. The specified structure must include details of the required representation of data including *SNOMED CT expressions*.
- The required data is extracted and uploaded to a database designed for the purpose of large scale analysis. Usually the extract and upload will need to be repeated or updated at specified intervals.
- The central database is structured to optimize common types of queries taking account of the *SNOMED CT expressions* and the *relationships* between referenced *concepts* asserted in *SNOMED CT* content.
- A *query* interface is provided to allow common types of question to be expressed against the central database.
- Queries are run taking account of the *relationships* between *concepts* to provide comprehensive and accurate results (minimizing the risks of false negatives or false positives).
- The results of queries are presented where relevant using *SNOMED CT expressions* as processable labels to enable further analysis.

- A common *query* specification supported by clinical record systems:
  - A common *reference information model* including *SNOMED CT expressions* is specified. This is used as a common *model of meaning* against which queries are evaluated.
  - Each clinical record system provider implements this common *model of meaning* as a *view* of the information stored in their *electronic health record*.
  - A *query* interface is provided to allow common types of question to be expressed against the common *model of meaning*.
  - Queries are distributed and run on individual systems and the results are returned to a central system for aggregate reporting.
  - The results of queries are presented where relevant using *SNOMED CT expressions* as processable labels to enable further analysis.

In practice , there is significant overlap between these two approaches. A data warehouse approach can benefit from a common approach to specifying the information extraction requirements. This allows changes to the specification without re-engineering the contributing clinical record systems. A common *query* specification approach requires a central element to manage distribution of queries and aggregation of results.

Irrespective of the approach taken, *SNOMED CT* enabled aggregation and analysis is most effective where the representation of information in the contributing clinical record systems is consistent with a common view. However, it is possible to aggregate information from diverse systems if the limits imposed by differences are understood. It is even possible for a *SNOMED CT* aggregation and analysis system to be applied information that was not originally encoded using *SNOMED CT*. An extraction and aggregation interface that includes mapping from another coding system may produce information of adequate quality and consistency for many purposes. Data derived by tagging textual records using *natural language processing* may also meet requirements that are not safety-critical.

3.2.4 Terminology tools

SNOMED CT enabled terminology tools provide access to *SNOMED CT* content. On their own they are not practical end-user implementations but they enable the development and review of *SNOMED CT*. They may also deliver services that can be used by end-user implementations.

3.2.4.1 Implementation Types - Terminology browser

A *SNOMED CT* enabled *browser* allows the content and structure of *SNOMED CT* to be explored and reviewed.

A typical *SNOMED CT* enabled *browser* can locate *concepts* and *descriptions* by *Identifiers* and by searching the text of *description terms*. Various views of located *concepts* may be displayed including the set of related *descriptions*, the hierarchical *relationships* and other defining *relationships*.

A terminology *browser* may be:

- A stand-alone tool.
- Part of a more extensive implementation.
- Accessible via an *Application Programming Interface* (*API*):
    - This may allow the *browser* to be used by client applications to select *SNOMED CT expressions*;
    - It may be part of a *terminology server* which provides a wider range of *Terminology services*.

3.2.4.2 Implementation Types - Terminology server

A *SNOMED CT enabled terminology server* is a software application that provides programmatic access to *SNOMED CT components*. These services are made available through a documented *Application Programming Interface* (*API*) which can be used by many different client applications.

A *SNOMED CT enabled terminology server* must be able to *import SNOMED CT release files* and provide some or all of the services described in the *Terminology Services Guide (7)*. All *terminology servers* must support a basic minimum set of functions including *Foundation Terminology Services* and access to *Reference sets and other metadata*.

A *terminology server* may provide *user interface* services, such as a set of screen controls to support term selection. Alternatively, while the *API* should support searches, the *user interface* representation of the results of a search may be left to client applications. Where *user interface* controls are provided by the server, these controls may also be packaged in an integrated form as a *terminology browser*.

A *SNOMED CT enabled terminology server* may also provide services that support the use of other terminologies. In this case, it may conform to a standard specification such as *Common Terminology Services 2* (*CTS2*).

3.2.4.3 Implementation Types - Terminology development and maintenance tools

SNOMED CT development and maintenance requires tools which are able to create and update *SNOMED CT* content.

Development and maintenance tools may either be general purpose or may focus on specific requirements (e.g. *Reference Sets* to support *language*, *mapping* or development of *value sets*).

The process of maintenance needs to track changes and manage conflicts between edits made by different authors. In the case of content development, the tools must also ensure that *concept* definitions conform to the *SNOMED CT Concept Model*. At regular intervals the tools need to generate a consistent set of quality assured *release files*.

The *IHTSDO Workbench* is a set of software tools designed to support the development, maintenance, and use of *SNOMED CT*. Its key role is to facilitate the maintenance of the *SNOMED CT International Release* and the National *Extensions* developed by *IHTSDO Members*. However, the future scope of use may extend to other organizations and to health information systems around the world. The *Workbench* is owned by the *IHTSDO* and is available under an Open Source license agreement.

3.3 Implementation Levels

SNOMED CT can be implemented in a wide range of clinical record applications. These include systems developed for use with other code systems that have been adapted to support *SNOMED CT* as well as systems designed with the assumption that *SNOMED CT* would serve as the primary terminology. The *SNOMED CT* features that applications support and use may vary, partly due to differences in user requirements and partly due to development priorities. Against this background of variability, it is reasonable to ask what is a *SNOMED CT implementation* or what is a good *SNOMED CT implementation*. While there is not a single or simple answer to these questions, this section identifies some key dimensions which determine the capability of *SNOMED CT* enabled clinical record systems.

Each of the following sections describes a dimension and outlines a spectrum of capabilities ranging from absence of support (Level 0) to full support (Level 2). A mixture of Level 0 and Level 1 capabilities are likely to be found in existing systems that have been adapted to work with *SNOMED CT*. A system specifically developed to work with *SNOMED CT* should be expected to have capabilities that are at least at the high end of the Level 1 spectrum and should ideally have Level 2 capabilities.

The specification of different levels is not intended to suggest a step-by-step development path. Those needing to rapidly *SNOMED CT* enable an existing clinical record system are recommended to follow a two stage approach.

1. Design, develop and deploy a revision to the current system to support Level 1 capabilities that meet known short or medium term requirements:
    - The level achieved in this stage will depend on customer requirements and the design limitation of the existing system.
2. Design and develop a new or substantially revised system (including revised record structures) to support a mixture of high-end Level 1 and Level 2 capabilities:
    - The level at which this development is target should be one that meets anticipated medium to long term requirements;
    - Even if the initial target of the work is limited to the high-end of Level 1, the design should be sufficiently flexible to enable Level 2 capabilities to be added when required.

Developers who do not require a rapid deployment based on a revision of an existing systems are recommended to skip the first step and proceed to design and develop a flexible solution that utilizes the key strengths of *SNOMED CT*.

Each of the following sections describes one dimension that contributes to the overall implementation level. It is important to recognize that:

- This is not a formal scoring scheme:
    - Some dimensions are more significant than others;
    - The significance of reaching a particular level depends on the nature of the application and the user requirements it seeks to address.
- Many of the dimensions are inherently interdependent:
    - For example, Level 2 data entry capabilities are not compatible with Level 1 data storage.

3.3.1 Implementation Level - Scope of use

A clinical record system may use *SNOMED CT expressions* to represent some or all of the types of information outlined in the list below. The types of information for which *SNOMED CT* can be used may be limited by the structure used to store the *electronic health record*. The significance of these limitations depends upon the intended use of the clinical record system.

- Level 0: No support for *SNOMED CT expressions*.
- Level 1: Support for use of *SNOMED CT* limited to particular types of clinical data:
    - Addressing the requirements for a particular type of use;
    - Addressing a set of requirements specified by a particular organization .
- Level 2: Support for consistent use of *SNOMED CT* across a broad scope of information types:
    - Providing a general purpose approach to the use of *SNOMED CT* within an *electronic health record*
    - Allowing configuration to vary the scope of coverage to meet specific requirements.

The following check-list identifies some of the *electronic health record* elements in which *SNOMED CT expressions* might be used. The list is not complete but it covers many of the areas in which use of *SNOMED CT* has been discussed in *IHTSDO* working groups. It is intended to assist consideration of the areas in which *SNOMED CT* should be used to meet the needs of users and organizations . The inclusion of an item in this list does not imply that the *SNOMED CT International Release* provides comprehensive content to populate that part of the record.

1. Disorders, diagnoses and problems:

- Problem list entries;
- Admission diagnosis;
- Discharge diagnosis;
- Provisional or working diagnosis;
- Differential diagnosis.

2. Symptoms:

- Presenting symptoms;
- History of current condition;
- Other symptoms.

3. Allergies and adverse reactions:

- Adverse reaction events;
- Allergies and other propensities to adverse reactions.

4. Procedures:

- Operative procedures.
- Diagnostic procedures.
- Medications:
    - Current medication;
    - Prescriptions;
    - Dispensing records;
    - Drug charts.
- Other therapeutic procedures:
    - Other therapy requests;
    - Other therapy delivery and outcomes.

5. History:

- Medical and surgical past history;
- Medication history;
- Family history.

6. Examination findings:

- Vital signs;
- Clinical examination findings.

7. Investigation information:

- Laboratory investigations:
    - Laboratory investigation requests;
    - Laboratory investigation procedures;
    - Laboratory investigation results.

- Diagnostic imaging:
  - Diagnostic imaging requests;
  - Diagnostic imaging procedures;
  - Diagnostic imaging results.
- Other investigations:
  - Other investigation requests;
  - Other investigation procedures;
  - Other investigation result.

8. Other types of clinical information:

- Planned actions;
- Risk, goal and expected outcomes;
- Scale based assessments;
- Progress notes.

9. Administrative information:

- Admission, transfer and discharge events.

10. Other values:

- Body sites, structures and locations;
- Organisms;
- Substances (other than drugs);
- Pharmaceutical and biological products (drugs).

3.3.2 Implementation Level - Record structure

The logical model underlying the structure of the record has a direct effect on the ability of a *SNOMED CT* enabled clinical record system to take advantage of the features of *SNOMED CT*. An application may use an optimized proprietary internal representation of the *electronic health record*. However, consistent use of *SNOMED CT* across a range of applications requires a common reference model to which proprietary structures are mapped. In addition to this, the ways in which *SNOMED CT expressions* are used within a common *reference information model* need to be constrained to improve predictability and minimize ambiguity.

- Level 0: A proprietary structure that is neither aligned with nor mapped to a standard *reference information model*:
  - Low: Text only record with no use of clinical codes;
  - High: Structured record supporting use of clinical codes.
- Level 1: A structure that is aligned with or mapped to a standard *reference information model*:
  - Low: Proprietary structure mapped to a standard model to support limited messaging requirements. Supports the use of *SNOMED CT* coding within that structure.
  - High: Structure aligned with a standard *reference information model* that supports that supports use of *SNOMED CT* coding.
  - Examples of standard *reference information models* include:
    - The *HL7* Version 3 *Reference Information Model* (RIM);
    - The *CEN TC251* Health informatics - *Electronic health record* communication - Part 1: Reference model (*EN13606*).

- Level 2: An aligned or mapped structure in which *SNOMED CT expressions* are used in accordance with agreed guidelines for use of a standard *reference information model*:
    - In Level 2 *SNOMED CT* is used in accordance with *terminology binding* guidance to minimize the semantic gaps and overlaps between the terminology and the information model. Without constraints, these gaps and overlaps lead to inconsistent representation of similar data and thus limit the effective reuse of information.
    - Example of agreed guidelines for using use of *SNOMED CT expression* in particular reference models include:
        - The *HL7TermInfo* DSTU - Guide to the Use of *SNOMED CT* in *HL7* Version 3;
        - Guidance on *terminology binding* developed by the *UK NHS* Logical Record Architecture for use in an *EN13606* based logical model.

3.3.3 Implementation Level - Expression storage

Support for storing *precoordinated* and *postcoordinated SNOMED CT expressions* determines the extent to which *SNOMED CT* can be used to represent detailed information within an *electronic health record*.

- Level 0: No support for storage of *SNOMED CT expressions*
- Level 1: Support for storage of *precoordinated SNOMED CT expressions*:
    - Support for storage of a *precoordinated expression* implies the ability to store a representation of a *concept identifier* as part of each item for which *SNOMED CT* is used:
        - The *concept identifier* may be represented as a 64-bit *integer* or as an 18-digit *string*;
        - Other internal representations may be used provided they can be resolved to the appropriate *Identifier* for display, communication or processing.
- Level 2: Support for storage of *postcoordinated SNOMED CT expressions*:
    - Support for storage of *postcoordinated expression* implies the ability to store a representation that captures the logical model of a *postcoordinated expression*:
        - The simplest representation of a *postcoordinated expression* is the *SNOMED CT compositional grammar*. Due to the open-ended nature of *postcoordinated expressions*, this representation results in a string of variable length with no clear-cut maximum length.
        - The guide discusses alternative representations including the use of *expression* reference table which enables use of a fixed length reference within the records. This approach uses a *UUID* which can be represented as a 128-bit *integer* or as a hexadecimal *string* (see Storing expressions) .
    - This level has variants depending on the extent of support for *postcoordinated expression* storage:
        - Low: Storage of *postcoordinated expressions* limited to specific fields in the record structure;
        - High: Full support for storage of *postcoordinated expression* allowing any valid *expression* to be stored and retrieved.

3.3.4 Implementation Level - Data entry

The categorization in this section in based on the extent to which the system enables entry of *SNOMED CT expressions*. In addition, this section indicates the importance of a well-designed user-interface.

- Level 0: No support for entry of *SNOMED CT expressions*.

.

- Level 1: Support for *precoordinated SNOMED CT expression* entry:

    - Low: Access limited to fixed set of *SNOMED CT concepts*;
    - Medium: Access to full content of *SNOMED CT*;
    - High: Access to full content of *SNOMED CT* with configurable *value sets* matched to user requirements.

- Level 2: Support for *postcoordinated expression* entry:
    - Low: Access to limited *postcoordination* (matching data storage restrictions);
    - Medium: Access to full range of *postcoordination* supported by the *Concept Model*;
    - High: Access to *postcoordination* with configurable *constraint* matched to user requirements.

Another important data entry issue is the ease of use which depends on the usability, relevance and performance of searches. Where *postcoordinated* data entry is supported the approach to selecting or constructing *postcoordinated expressions* is also significant.

An attempt to categorize specific approaches to the user-interface is subjective as alternative *user interfaces* may be appropriate to different uses. However, for most environments a flexible range of configurable *SNOMED CT* aware user-interface tools is likely to offer a better user experience than reliance on a one-size fits all *browser* or search engine.

3.3.5 Implementation Level - Data retrieval

A major strength of *SNOMED CT* is its ability to support meaning based selective retrieval. The extent to which this feature is used by a clinical record system determines the value of entering and storing the data.

- Level 0: No native support for *SNOMED CT* enabled data retrieval:
    - This level has variants depending on whether it can map code in exported data to *SNOMED CT expressions*:
        - Low: No support for *SNOMED CT* based analysis;
        - High: Support for extracting a specified set of locally coded data and mapping the local codes to appropriate *SNOMED CT expression* for central aggregation and analysis.
- Level 1: Support for retrieval of *precoordinated SNOMED CT expressions* :
    - This level has a spectrum of variants depending on the level of support for the following features:
        - *Query* expressivity: The ability to express *query* predicates that explicitly include or exclude *subtypes* of specifically identified *concepts*;
        - Subsumption testing: Use of *SNOMED CT subtype hierarchy* to interpret and evaluate queries;
        - *Concept Equivalence*: The ability to retrieve equivalent information even if it is represented in different structures within the record;
        - Context awareness: The ability to take account of contextual information, derived from the record structure and/or the *SNOMED expression*, when interpreting and evaluating queries;
        - Performance: The ability to interpret and evaluate queries within an appropriate period of time and without causing deterioration in other system functions.
- Level 2: Support for retrieval of *postcoordinated SNOMED CT expressions* :
    - This level has a spectrum of variants depending on the level of support for the following additional aspects of the features specified for Level 1:
        - *Query* expressivity: The ability to represent *postcoordinated* predicates in a *query*;
        - Subsumption testing: Use of *defining characteristics* and *normal form transformations* (or a *description logic classifier*) to determine whether *expressions* are subsumed by *query* predicates;
        - *Equivalence*: Use of *defining characteristics* and *normal form transformations* (or a *description logic classifier*) to determine *equivalence* between different *postcoordinated expressions* and in different structures within the record;

- Context awareness: The ability to take account of contextual information derived from the record structure and/or *postcoordinated SNOMED expressions*, when interpreting and evaluating queries;
- Performance: The ability to interpret and evaluate queries that support *postcoordinated* representations within an appropriate period of time and without causing deterioration in other system functions.

3.3.6 Implementation Level - Communication

The ability to send and received *SNOMED CT expressions* in messages or other communication is partially dependent on data entry, storage and retrieval capabilities. However, some types of communication may be supported by mapping or human-readable renderings even in the absence of internal support for *SNOMED CT*.

- Level 0: Mapping based support for communication of *SNOMED CT expressions* :
  - Inbound communications containing *SNOMED CT expressions* :
    - Low: Not supported.
    - Medium: Rendered as human-readable text. Unless the inbound message also contains the *term* text, this requires access to some *SNOMED CT* enable *Terminology services* to lookup and display the relevant *terms*.
    - High: Mapped to an internal coding scheme or classification. This may be feasible to support specific use cases but not for the full scope of clinical information.
  - Outbound communication containing *SNOMED CT expressions* :
    - Low: Not supported;
    - Medium: Supported for a few specific types of clinical data in the existing system by mapping to from an existing code system to *SNOMED CT*;
    - High: Supported for most clinical data in the existing system by mapping to from an existing code system to *SNOMED CT*.
- Level 1: Native support for communication of *precoordinated SNOMED CT expressions* :
  - Inbound communications containing *precoordinated SNOMED CT expressions* :
    - Low: Supported for some types of information but constrained by data entry and *expression* storage capabilities;
    - High: Supported for most types of information.
  - Outbound communications containing *precoordinated SNOMED CT expressions* :
    - Low: Supported but limited by data entry and storage and retrieval capabilities;
    - High: Supported for most types of information.
- Level 2: Native support for communication of *postcoordinated SNOMED CT expressions* :
  - Inbound communications containing *postcoordinated SNOMED CT expressions* :
    - Low: Support limited to particular attributes (e.g. |laterality|, |causative agent|) in *postcoordinated expression*;
    - Medium: Support for general *postcoordination* applied to some types of information;
    - High: Able to receive, process and store any valid *postcoordinated expression*.
  - Outbound communications containing *postcoordinated SNOMED CT expressions* :
    - Low: Support limited to particular attributes (e.g. |laterality|, |causative agent|) in *postcoordinated expression*;

- Medium: Support for outbound communication of any *postcoordinated expression* that can be entered or stored in the system;
- High: Support for outbound communication of any valid *postcoordinated expression*.

3.4 Implementation Services

When designing or implementing a *SNOMED CT enabled application*, the first step is to assess the range of services necessary to meet user requirements. The two main categories of services required by applications are *terminology services* that only interact with the terminology and *record services* which apply the terminology to instance data. These services are described in separate sections of this guide

The *Terminology Services Guide (7)* describes services that access *SNOMED CT* reference data. These services are summarized in *Figure 2*.

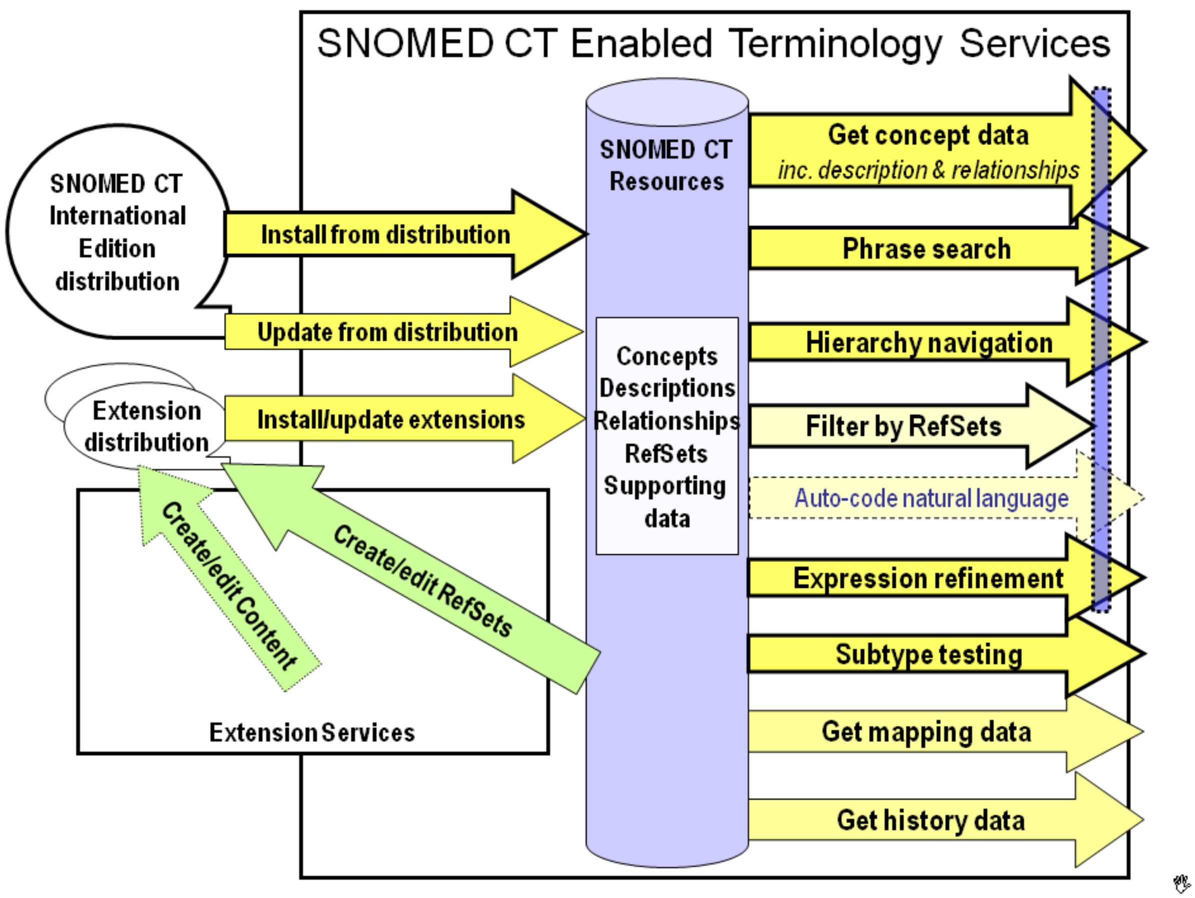


Figure 2: SNOMED CT Enabled Terminology services

The *Record services guide (8)* describes services that apply *SNOMED CT* to represent information in a clinical record. These services are summarized in *Figure 3*.

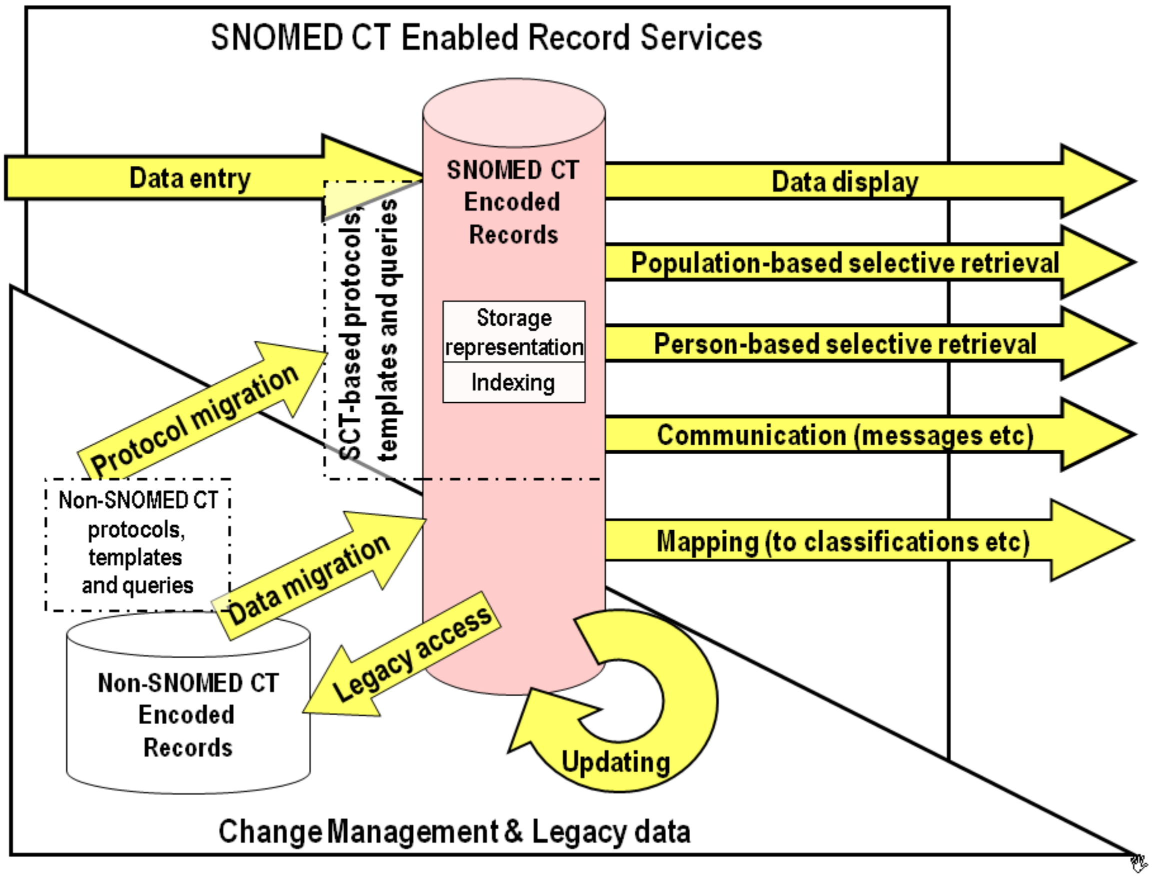


Figure 3: SNOMED CT Enabled Record Services

3.4.1 Service architecture

A *SNOMED CT enabled application* may be completely self-contained, delivering all the required services as part of a single development. Alternatively, service delivery may be modularized so that separately developed reusable modules are used to meet specific sets of requirements.

A distinction can be made between functions that only require interaction with terminology resources (*terminology services*) and functions that involve using the terminology as part of an application such as an *electronic health record* (*record services*).

Terminology services can be generalized , so that they are independent of the way the terminology is used in a particular clinical record application. *Terminology services* include support for the following types of function.

- Read-only functions:
  - Importing and updating a local terminology repository with a *SNOMED CT release*;
  - Determining the properties or an identified *component*;
  - Text or pattern searches for *Descriptions* that include a matching term;
  - Displaying a part of the *concept* hierarchy;
  - Determining whether a *SNOMED CT concept* or *expression* is equivalent to or a *subtype* of another *concept* or *expression*;
  - Locating the *map* from a particular *SNOMED CT concept* to a code in another scheme or classification.
- Authoring and maintenance functions:

- Enabling the creation and maintenance of core *SNOMED CT components* to facilitate production of the *SNOMED CT International Release* and *Extensions* to *SNOMED CT*;
- Enabling the creation and maintenance of derivative such as *reference sets* to customize and enhance the effective use of *SNOMED CT*.

Record services are intimately related to ways in which information is entered, stored and retrieved by a particular application. Therefore, while these services interact with *terminology services* they are usually specific to a particular application or to a family of applications with a common underlying record design. *Record services* include support for the following types of function:

- *User interface* functions that:
  - Enable entry of information using *SNOMED CT expressions* where these are relevant;
  - Display of previously entered information, with appropriate rendering of *SNOMED CT expressions*;
  - Enable design of protocols that guide data entry to encourage efficient and consistent use of *SNOMED CT*;
  - Enable specification of queries that include appropriate use of *SNOMED CT* to meet requirements for selective retrieval.
- Application server functions that:
  - Store *SNOMED CT expressions* as part of the individual record entries (or in other types of instance data);
  - Communicate data including *SNOMED CT expressions* in ways dictated by standards and local specifications;
  - Apply queries to efficiently, accurately and precisely retrieve information taking account of the data structure of the application and the logical *Relationships* between *SNOMED CT expressions*.

These two sets of services can be developed and provided separately. This approach allows *record service* to access required *terminology services* through an *Application Programming Interface* (*API*). The guide does not specify an *API* but, by making a clear distinction between *terminology services* and *record services*, it identifies the functions that such an interface should support.

Self-contained and modular approaches offer different profiles of advantages, some of which are summarized below.

- A modular approach offers the following advantages:
  - Rapid development of *SNOMED CT* related functionality, focused on meeting the requirements of users of a specific software application.
  - Opportunities to choose between different *terminology servers* to deliver a cost-effective solution.
  - Simplifies future migration to enhanced or more cost-effective solutions by separately identifying reusable and replaceable modules.
  - Allows several applications used by a single organization to use a single *terminology server*. This has several advantages:
    - Reduction of maintenance and support cost associated with installing each release of *SNOMED CT*;
    - Guaranteed alignment of *SNOMED CT releases* between applications that share the server;
    - Consistency of the *user interface* and technical characteristics of different applications with respect to their access to *SNOMED CT*.
- A fully integrated approach offers the following advantages:
  - Independence of third party development;
  - Customized access to *SNOMED CT* tailored to the needs of particular application users.

The approach chosen depends on a careful consideration taking into account the cost and functionality of available *components*. Commercial and technical concerns about dependence on third-party *components* may be a valid reason for in-house development of all the required services. However, even where all the development is undertaken within a single organization , separation of terminology and *record services* into separate *components* may offer a more robust approach, allowing future extensibility and migration at lower cost.

Chapter 4

4 Structure and Content Guide

This part of the guide covers the features of *SNOMED CT* that need to be understood by those implementing *SNOMED CT* in software applications. These features include the components, *derivatives* and supporting materials that are distributed as part of each *SNOMED CT Release*. In addition, the guide addresses the ways in which these components may be referenced to represent instances of clinical information in clinical records and other types of instance data.

4.1 SNOMED CT Technical Overview

This section provides an overview of the *components* and *derivatives* that form part of a *SNOMED CT release* as well as several other topics that relate to the use of *SNOMED CT* to represent instances of clinical information.

These topics are explored in more depth by other sections in this part of the guide:

- *Logical Abstract Models*;
- *Representational Forms*.

More detailed information about technical design and content is provided in other parts of the guide:

- *Release File Specifications (5)*;
- *Concept Model Guide (6)*.

4.1.1 Components

This section summarizes the essential *components* of *SNOMED CT* (*concepts*, *descriptions* and *relationships*). A *SNOMED CT enabled implementation* must be able to process and make appropriate use of these *components*, which are distributed as a set of *Release Files (5)*.

4.1.1.1 Concepts

A *SNOMED CT Concept* is a clinical idea to which a unique *SNOMED CT identifier* has been assigned.

Each *Concept* is associated with:

- A unique human-readable *Fully Specified Name* (FSN), which specifies the meaning represented by the *Concept*.
- A set of other *Descriptions*, each of which represents the same *Concept* using a different human-readable *term*. These *Descriptions* support alternative representations such as *synonyms* and translations into different *languages*.
- A set of *Relationships* to other *Concepts* which provide a logical definition of the *Concept* that can be processed by a computer.

4.1.1.1.1 Concept Identifiers

Each *SNOMED CT Concept* has a permanent unique numeric *Identifier* which is known as the *Concept Identifier*.

The sequence of digits in a *Concept Identifier* does not convey any information about the meaning or nature of the *Concept* [1]. The meaning of *Concept* is represented in human-readable forms by *Descriptions* and in a computer processable form by *Relationships* with other *Concepts*.

The advantages of meaningless *Identifiers* include:

- *Identifier* permanence without undermining interpretation:
  - In contrast, to maintain consistency, a meaningful code may need to change to reflect revised understanding of the nature of a disorder. .
- Enabling multiple aspects of meaning to be represented in the same way:
  - A meaningful code can only represent part of meaning of a complex *concept*. For example, |staphylococcal pneumonia| is an |infection|, a |respiratory disorder| and a |disorder| caused by |staphylococcus| but only one of these aspects can be represented by a code based *hierarchy*. Thus in the 'J' in the *ICD-10* code 'J152: Pneumonia due to staphylococcus' represents that fact that this is a respiratory disorder but does not represent the fact that it is an infection (codes starting with 'A') or that it is due to staphylococcus ('A490: Staphylococcal infection, unspecified').
- No artificial limitation on *concept* granularity:
  - Typical approaches to meaningful coding impose limits on both the number of levels of specificity (i.e. the length of the code) and the number of options at each level (i.e. the number of different symbols that can be used in each character position).

4.1.1.1.2 Concept granularity

The meaning represented by a *Concept* can be general (for example | procedure |), specific (for example | excisional biopsy of lymph node |) or somewhere in between (for example | biopsy of lymph node |).

- More specific *Concepts*:
  - Have finer granularity (more granular);
  - Represent clinical detail.
- More general *Concepts*:
  - Have coarser granularity (less granular);
  - Represent less clinical detail;
  - Aggregate similar *Concepts*.

Support for multiple levels of granularity allows *SNOMED CT* to be used to represent clinical data at a level of detail that is appropriate to a range of different uses.

Concepts with different levels of granularity are linked to one another by | is a | *relationships*. This enables appropriate aggregation of specific information within less detailed categories.

[1] The use of meaningless identifiers differs from the approach taken by some other coding systems and classifications. For example, the first character of an ICD-10 code indicates the general classification that it falls within.

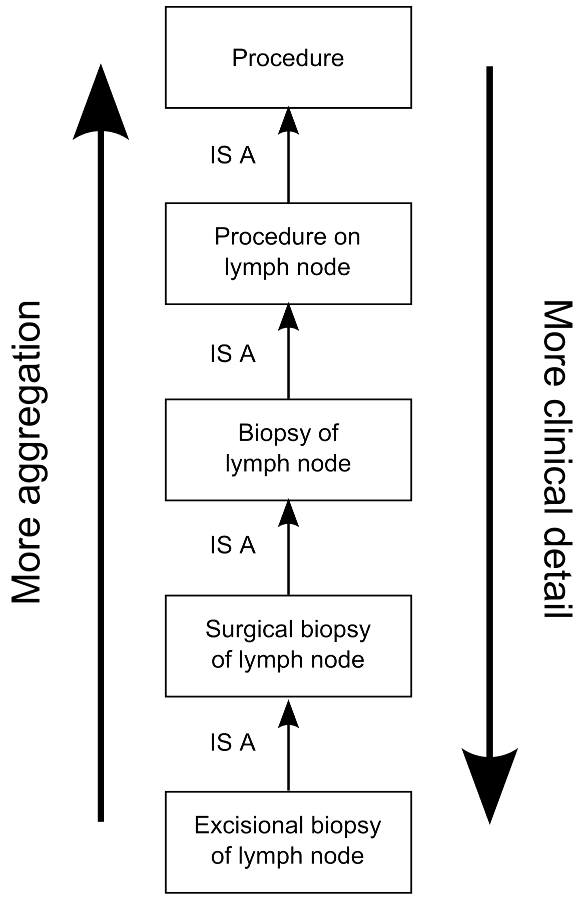


Figure 4: Multiple levels of granularity

4.1.1.2 Descriptions and Terms

Terms are character strings that consist of words, phrases and other human-readable representations that convey the meanings of *concepts*. A term in connection to a particular *concept* is called a *description*.

Each *descriptions* has a *description* type and may be marked a preferred for use in particular languages or *dialects*. There are two commonly used *description* types, *Fully Specified Name* (FSN) and *Synonym*. A *Synonym* that is marked as *preferred* for use in a particular language or dialect is preferred to as a *Preferred Term*. A *description* may be a *Preferred Term* in one *dialect* and a *synonym* in another *dialect*. This is indicated by references to the *description* from the Language *Reference Set* for that language or *dialect*.

- Each *description* associates a human-readable *term* with one *concept*.
- A *concept* has several associated *descriptions*.
- Each *description* has a unique *Description Identifier* and is distributed as a row in the *Description File*.

4.1.1.2.1 Fully Specified Name

Each *concept* has at lease one *Fully Specified Name* (FSN) intended to provide an unambiguous way to name a *concept*. The purpose of the FSN is to uniquely describe a *concept* and clarify its meaning. The FSN is not a commonly used term or natural phrase and would not be expected to appear in the human-readable representation of a clinical record.

A *concept* may have more than one FSN, but only one of these may be marked as *preferred* in a given language. A Language *Reference Set* is used to specify which FSN *descriptions* is *preferred* in each language or *dialects*. The original *fully specified name* (the first FSN created for a *concept*) is the ultimate source of reference, if FSNs in different languages have conflicting meanings. Most original FSNs are in US English and, as many translators choose not translate FSNs, the original FSN is preferred by default.

Note: The term in each FSN is unique across the entire active content of a *SNOMED CT release*.

Each FSN term ends with a “semantic tag” in parentheses. The semantic tag indicates the semantic category to which the *concept* belongs (e.g. clinical finding, disorder, procedure, organism, person, etc.). The “semantic tag” helps to disambiguate different *concepts* which may be referred to by the same commonly used word or phrase.

Example: | Hematoma (morphologic abnormality) | is the FSN of the *concept* that represents the “hematoma” that a pathologist sees at the tissue level. In contrast, | Hematoma (disorder) | is the FSN of the *concept* that represents the clinical diagnosis that a clinician makes when they decide that a person has a “hematoma”.

4.1.1.2.2 Synonym

A *synonym* represents a *term*, other than the FSN, that can be used to represent a *concept* in a particular language or *dialect*.

Each *concept* one or more *descriptions* of type synonym in each language. A *description* of type *synonym* contains a term that represents a word or phrase, other than the term in the *fully specified name* that can be used to represent a *concept*. One synonym for each *concept* is marked as *preferred* in each *dialect* and the associated term is called the *preferred term* for that *concept*.

The use of a *description* can vary between different languages, *dialects* and contexts, so a *description* may be *preferred* in some *dialects*, *acceptable* for use in other *dialects* and may not used in some *dialects*. A Language *Reference Set* is used to specify the *descriptions* that are *acceptable* or *preferred* in each language or *dialect*.

Example: *Synonyms* of the *concept* 22298006 | myocardial infarction (disorder) | in English include:

- | cardiac infarction | (*Description*.id: 37442013);
- | heart attack | (*Description*.id: 37443015);
- | infarction of heart | (*Description*.id: 37441018);
- | myocardial infarction | (*Description*.id: 37436014).

The *synonym* | myocardial infarction | (*Description*.id: 37436014) is marked as *preferred* in the US English Language *Reference Set*. Thus in US English this is the *preferred term*.

Note: Unlike *fully specified names*, synonyms are not required to be unique.

4.1.1.2.3 Preferred Term

The *preferred term* is the preferred common word or phrase used by clinicians to name that *concept* in a particular language, *dialect* or context. Each *concept* has one to more *descriptions* of type *synonym* in each language. In each language or *dialect* one of these *description* is marked as *preferred* and is the *preferred term* for that *concept*.

The use of a *description* can vary between different languages, *dialects* and contexts, so a *description* may be *preferred* in some *dialects*, *acceptable* for use in other *dialects* and may not used in some *dialects*. A Language *Reference Set* is used to specify the *descriptions* that are *acceptable* or *preferred* in each language or dialect.

Example: The *concept* 54987000 | repair of common bile duct (procedure) | has a *description* of type *synonym* | choledochoplasty |. This is marked as *preferred* in the US English Language *Reference Set*. Therefore, | choledochoplasty | is the *preferred term* for this *concept* in US English.

Note: Unlike the *fully specified name* (FSN) the *preferred terms* need not be unique. Occasionally, the *preferred term* for one *concept* may also be a *synonym* for a different *concept*. Interpretation in these cases will depend on context of use.

Example:

- I Cold sensation quality (qualifier value) I has a *preferred term* of “Cold”;
- I Common cold (disorder) I also has a *synonym* of “Cold”.

In both cases, “cold” represents a common clinical phrase used to capture the meaning of the *concept*.

Note: Selection of one term over another as "preferred" in a given language *dialect* depends entirely on whose preferences are being expressed. Different users are likely to have different preferences, and implementers are encouraged to select terms that properly represent the *concept* and meet the preferences of users. There is no expectation that the *preferred term* distributed with a given language *dialect* will meet all use cases; nor is there anything sacrosanct about the term. The US English *preferred term* has no special status relative to other terms. Rather, it is merely one term that properly represents the *concept* and can be used as a starting point.

4.1.1.3 Relationships

A *Relationship* represents an association between two *Concepts*.

Each *Relationship* is identified by a unique *Relationship* Id and is distributed as a row in the *relationship file*.

A *Relationship* contains *Identifiers* of two logically associated *Concepts* and the *Identifier* of another *Concept* that indicates the *Relationship Type* by which they are associated.

Table 5: Example: Defining arthritis as a type of joint disorder

Relationship.id	sourceId	typeId	destinationId
2227469024	3723001	116680003	399269003
In human readable *terms* …	I arthritis I	I is a I	I joint disorder I

4.1.1.3.1 Relationships and concept definitions

Each *concept* in *SNOMED CT* is logically defined through its *relationships* to other *concepts*.

Every *active SNOMED CT concept* (except the *SNOMED CT Concept Root concept*) has at least one I is a I *relationship* to a supertype *concept*.

I is a I *relationships* and defining attribute *relationships* are known as the *defining characteristics* of *SNOMED CT concepts*. They are considered defining because they are used to logically represent a *concept* by establishing its *relationships* with other *concepts*. This is accomplished by establishing I Is a I *relationships* with one or more defining *concepts* (called supertypes) and modeling the difference with those supertypes through defining attributes.

Example: I Fracture of tarsal bone (disorder) I is defined as:

- I is a I *subtype* of I Fracture of foot (disorder) I
- and has I finding site I I Bone structure of tarsus (body structure) I ;
- and has I associated morphology I I Fracture (morphologic abnormality) I .

Note: A *relationship* is assigned only when that *relationship* is always known to be true.

Example: Group A Streptococcus causes most cases of Streptococcal pharyngitis. However, a small percentage of these cases are caused by other species of Streptococcus. Therefore, it would be incorrect to define I Streptococcal sore throat (disorder) I as having I causative agent I I Streptococcus pyogenes (organism) |. Instead it is correctly defined as having the more general | causative agent | | Genus Streptococcus (organism) |.

4.1.1.3.2 IS A Relationships

| is a | *relationships* are also known as "Supertype - *Subtype relationships*" or "Parent - *Child relationships*". | is a | *relationships* are the basis of *SNOMED CT*'s hierarchies, as illustrated below.

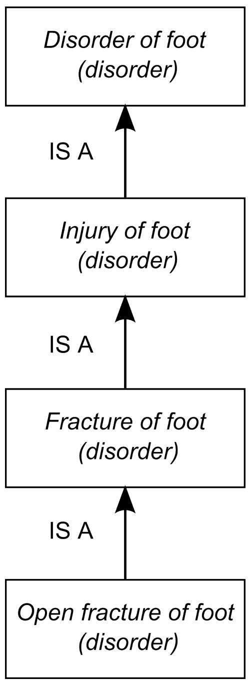


Figure 5: Example IS A hierarchy

A *concept* can have more than one | is a | *relationship* to other *concepts*. In that case, the *concept* will have parent *concepts* in more than one *sub-hierarchy* of a top-level *hierarchy*. *Subtype relationships* can be multi-hierarchical.

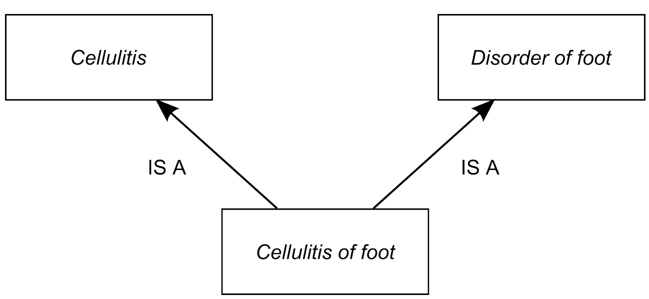


Figure 6: Example IS A Relationships

4.1.1.3.3 Attribute Relationships

An *attribute relationship* is an association between two *concepts* that specifies a *defining characteristic* of one of the *concepts* (the source of the *relationship*). Each *attribute relationship* has a name (the type of *relationship*) and a value (the destination of the *relationship*).

The combination of the *attribute relationships* and | is a | *relationships* associated with a *concept* represent the logical definition of that *concept*. Therefore, the logical *concept* definition includes one or more supertypes (represented by | is a | *relationships*), and a set of defining *attributes* that differentiate it from the other *concept* definitions.

Example:

Since pneumonia is a disorder of the lung, the logical definition of the *concept* | Pneumonia (disorder) | in *SNOMED CT* includes the following *relationship*. The *Attribute* | Finding site | is assigned the value | Lung structure (body structure) |.

- | Finding site | = | Lung structure (body structure) |

The full definitions of the *concepts* | Pneumonia (disorder) |,|Infective pneumonia (disorder)| and |Bacterial pneumonia (disorder)| are shown below. Each line represents a defining *Attribute* with a value.

- | is a | = | pneumonitis |
- , | is a | = | lung consolidation |
- , { | associated morphology | = | inflammation |
- , | associated morphology | = | consolidation |
- , | finding site | = | lung structure | }

Figure 7: Definition of |Pneumonia (disorder)|

- | is a | = | infectious disease of lung |
- , | is a | = | pneumonia |
- , | pathological process | = | infectious process |
- , { | associated morphology | = | inflammation |
- , | associated morphology | = | consolidation |
- , | finding site | = | lung structure | }

Figure 8: Definition of |Infective pneumonia (disorder)|

- | is a | = | bacterial lower respiratory infection |
- , | is a | = | infective pneumonia |
- , | causative agent | = | bacteria |
- , | pathological process | = | infectious process |
- , { | associated morphology | = | inflammation |
- , | associated morphology | = | consolidation |
- , | finding site | = | lung structure | }

Figure 9: Definition of |Bacterial pneumonia (disorder)|

Figure 10 illustrates some of these *Relationships* graphically. | is a | *Relationships* relate a *concept* to more general *concepts* of the same type. In contrast, *Attribute Relationships* (such as | Finding site | and | Causative agent |) relate a *concept* to relevant values in other branches of the *subtype hierarchy*.

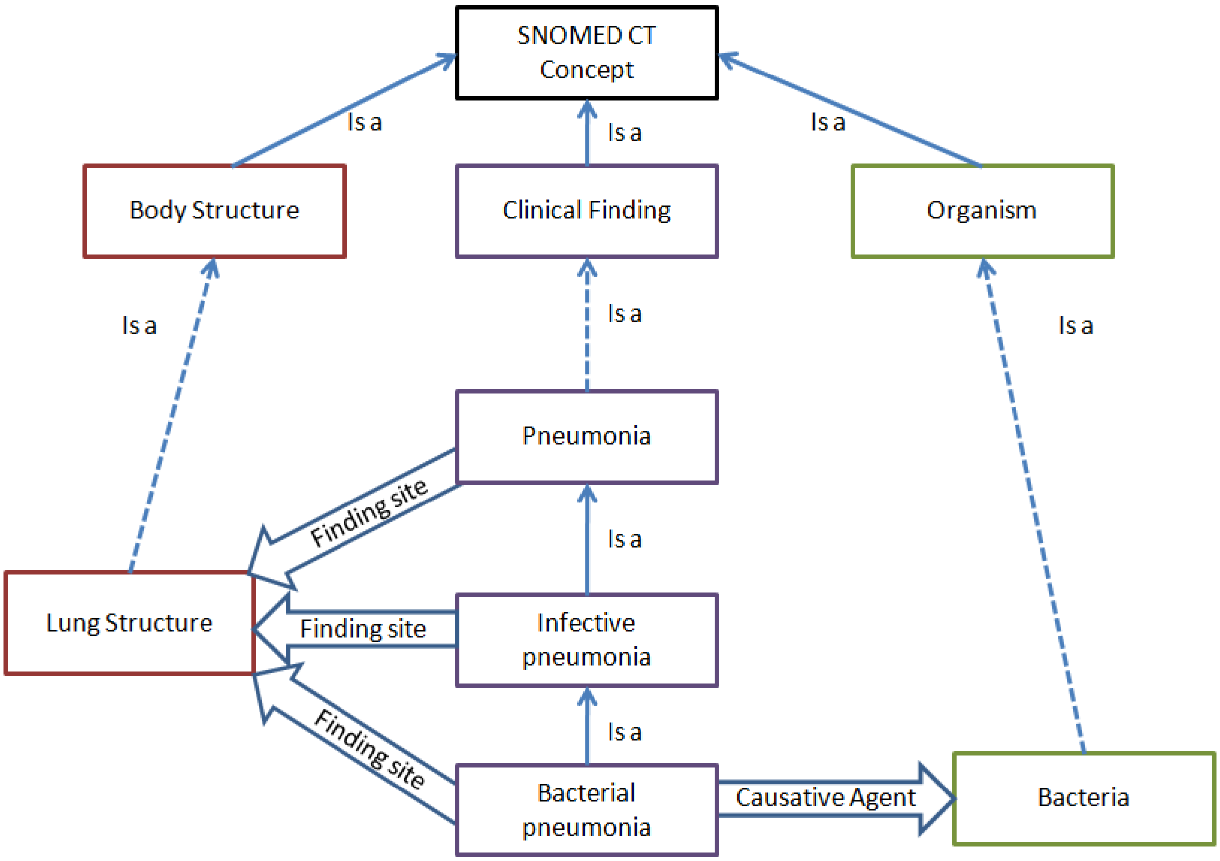


Figure 10: Illustration of Defining Relationships

4.1.1.4 Common Features of Components

This section describes common features of all *SNOMED CT Components* including identification and history management.

4.1.1.4.1 Component features - History

The content of *SNOMED CT* evolves with each release. The types of changes made include new *Concepts*, new *Descriptions*, new *Relationships* between *Concepts* and new *Reference Sets*, as well as updates and retirement of any of these *components*. Drivers of these changes include changes in understanding of health and disease processes; introduction of new drugs, investigations, therapies and procedures; and new threats to health, as well as proposals and work provided by *SNOMED CT* users.

Once released, the unique *Identifiers* of *SNOMED CT components* are persistent, and their *Identifiers* are not reused. When a *components* becomes inactive this is indicated by the value of the *active* field, which is present in all *components*. *Components* continue to be distributed even when they are no longer *active*. This allows a current release to be used to interpret data entered using an earlier release.

Since the implementation of *Release Format 2* (RF2), all changes in *components* are represented in the corresponding files, by adding a new row, with the same component ID, a new effective time and any necessary change in the component values.

The Component Inactivation *Reference Sets* are used to indicate the reason for inactivating a *component*. These reasons include errors, duplication of another component and ambiguity of meaning, and the files are used to describe reasons for inactivation of *Concepts*, *Descriptions* and *Relationships*.

Due to the origins of *SNOMED CT*, some *SNOMED CT Concepts* represent classification *concepts* that have imprecise and potentially changeable meanings. These are marked with the inactivation indicator value [900000000000486000|limited]. However, these *concepts* and were considered *active* until the January 2010 release of *SNOMED CT*. At that point, due to a change in editorial policy, this status was declared to be *inactive*. [2]

4.1.1.4.2 Component features - Identifiers

Components within *SNOMED Clinical Terms* are identified and referenced using numeric *Identifiers*. These *Identifiers* have the data type *SCTID* (*SNOMED CT Identifier*).

The *SCTID* data type is 64-bit *integer* which is allocated and represented in accordance with a set of rules. These rules enable each *Identifier* to refer unambiguously to a unique component. They also support separate partitions for allocation of *Identifiers* for particular types of component and *namespaces* that distinguish between different issuing organizations .

4.1.2 Derivatives

This section describes *derivatives* that are specified by and distributed as part of *SNOMED CT*. *Derivatives* are artifacts which are either required or useful to support some aspect of *SNOMED CT enabled implementation*. These artifacts are known as *derivatives* because they are derived from *SNOMED CT Components* and either add properties to them or specify sets of related *components*. All *SNOMED CT enabled applications* need to support some *derivatives*.

The set of *derivatives* that need to be supported by an implementation depend on user requirements for particular types of functionality. Important aspects of functionality that require support for relevant *derivatives* include:

- Tracking changes to the status of *components*;
- Filtering and prioritizing searches;
- Representing alternative *navigation hierarchies*;
- Adding annotations to *components*;
- *Mapping* to and from other coding schemes and classifications.

4.1.2.1 Reference Sets

Reference Sets are a flexible standard approach used by to support a variety of requirements for customization and enhancement of *SNOMED CT* content. These include representation of subsets, language preferences maps for or from other code systems.

Practical uses of *Reference Sets* include:

- Indicating the *descriptions* that contain acceptable and preferred *terms* for each *concept* in a particular language or dialect;
- Subsets of *components* that are included, excluded from the set of values that can be used in a particular country, organization , specialty or data entry context;
- Frequency of use of *descriptions* or *concepts* in particular country, organization , specialty or context;
- Suitability of particular *concepts* for use in a particular field in a record or message;
- Structure and ordering of hierarchies displaying *concepts* for user *navigation*.

Rows in *Reference Set* files reference a *component* that is a member of the set and may associate some additional information with the referenced *component* (e.g. whether a term is acceptable or preferred, codes that are the target of a maps).

Some types of *Reference Set* may also be represented by an intensional definition specified as a set of rule or constraints (e.g. all subtypes of a specified *concepts*.

[2] Some Concepts derived from classifications such as ICD-10 include the abbreviations NOS (not otherwise specified) or NEC (not elsewhere classified). These are only valid in respect of a particular classification and change in their meaning if additional precisely defined codes are added to that part of the classification. Furthermore, a Concept that is not otherwise specified in ICD-10 may well be more precisely represented by another SNOMED CT Concept and thus from a SNOMED CT perspective "otherwise classified."

4.1.2.2 Navigation Hierarchies

SNOMED CT subtype Relationships provide a logical semantic *hierarchy*. Often it is possible to view parts of the terminology and select particular *Concepts* by navigating through this *subtype hierarchy*. However, there are many situations in which the pure *subtype hierarchy* does not provide an ideal route for navigating to *concepts*.

Navigation links are used to provide an alternative route through parts of the terminology. A *navigation link* can link any two *Concepts* together to identify a useful route for *navigation*. Each of the *navigation links* is directional, linking a navigational parent *Concept* to a more refined navigational *child Concept*. However, unlike the *subtype relationship* the presence or absence of a *navigation link* neither adds to nor subtracts from the definition of either of the *Concepts* that it links.

Some *Concepts* may exist only to provide nodes in a *navigation hierarchy*. These *Concepts* are *subtypes* of |Navigational *Concept* | and play no part in the semantic definitions of any other *Concept*.

4.1.2.2.1 Uses of Navigational Hierarchies

4.1.2.2.1.1 Breaking down a subtype into manageable categories

Some *Concepts* have a large number of *subtype children* that cannot be logically divided into intermediate *subtypes*. At the *user interface* these result in long lists of options, which are difficult to visualize and navigate. Navigational *Concepts* with appropriate navigational links to the *supertype parent* and its *subtype children* provide an intermediate layer without disrupting the semantic definitions.

The | clinical finding | top-level *Concept* has a large number of *subtype children*. Intermediate *navigation Concepts* group some of these together in a convenient way.

Example:

Three *subtypes* related to pregnancy are grouped together under a single natural navigational *Concept*:

- Disorder of pregnancy / labor / delivery / puerperium [*navigation concept*];
- Disorder of pregnancy;
- Disorder of labor / delivery;
- Disorder of puerperium.

4.1.2.2.1.2 Bypassing levels in the subtype hierarchy

Some *Concepts* that are members of the same rational set of choices may be found at different levels in the *subtype hierarchy*. This may occur because some have intervening *subtypes* and some of these intervening *concepts* may not be required for data entry. Addition of new *concepts* in a release may change the *concepts* available at some levels in the *subtype hierarchy*. *Navigation links* can "bypass" levels in the *subtype hierarchy* to represent a rational sets of choices for use in a particular situation.

Example:

While it is semantically correct to nest | common cold | in the following *subtype hierarchy*, a user may reasonably expect to see "common cold" as an immediate navigational *child* of | upper respiratory infection |.

- | upper respiratory infection |
    - | Viral upper respiratory tract infection |
        - | common cold |

4.1.2.2.1.3 Linking related Concepts of different types

Navigational links can also be used to provide access to connected *Concepts* even when they are from different *hierarchy* branches.

Example:

A *navigation links* could associate:

- "hypertension" (the disorder) with | blood pressure | (the observation);
- | cataract | (disorder / finding) with "cataract surgery" (the procedure).

4.1.2.2.1.4 Ordering the display of subtypes

Sibling *Concepts* in a *subtype hierarchy* are not ordered. However, at the *user interface* a particular *order* may be useful to highlight commonly used *Concepts* or to mirror a conventional ordering.

Example:

Vertebrae, cranial nerves, disease stages, etc.

Navigational links are ordered and are used to impose *order*, even when the set of navigational *children* is the same as the set of *subtype children*.

4.1.2.2.1.5 Providing alternative hierarchies

The *subtype hierarchy* is logically defined and there can only be one such *hierarchy*. However, as *navigation* hierarchies have no definitional consequences, it is possible to have different hierarchies for different groups of users with differing needs.

Initial releases of *SNOMED CT* will contain a single set of *navigation links* but those engaged in technical implementation should be aware that in the future there may be separate sets of *navigation links* for use in different environments.

4.1.2.3 Maps

SNOMED CT specifications and content include resources that support *Mapping* to and from other code systems, classifications and terminologies. These resources support simple mapping, where there is a one-to-one *Relationship* between a *SNOMED CT concept* and code in a *target scheme*, and more complex maps where these are required.

More complex mapping requirements supported by the *SNOMED CT Mapping* model include:

- Maps from a single *SNOMED CT concept* to a combination of codes (rather than a single code) in the *target scheme*.
- Maps from a single *SNOMED CT concept* to choice of codes in the *target scheme*. In this case, the resolution of the choices may involve:
  - Manual selection supported by advisory notes.
  - Automated selection based on rules that test other relevant characteristics in the source data (e.g. age and sex of the subject, presence or absence of co-existing conditions, etc).
  - A combination of automated processing with manual confirmation or selection where rules are insufficient to make the necessary decisions.

In *Release Format 2 Maps* are represented using *Reference Sets*. The type of *Reference Set* used varies according to the nature and complexity of the mapping, there is a Simple Map *Reference Set* and a Complex Map *Reference Set*.

4.1.2.4 Search support

The *Developer Toolkit,* which is supplied as part of the *SNOMED CT International Release*, includes several tables that can be used to simplify and provide support for text searching.

There are two *WordKey Tables*. These tables link each word used in *SNOMED CT* to every:

- *Description* in which it is used;
- *Concept* associated with an *active description* in which the word is used.

There are also two *Dualkey Tables*. These tables link each abbreviated word pair to every:

- *Description* in which that pair of words is used;
- *Concept* in which the combined set of *active descriptions* contains that pair of words.

These tables are provided to assist implementation. However, use of these tables is optional, as developers may generate and use alternative search support resources.

An extended version of the *Developer Toolkit*, provides Java® programs to generate indexes that may be useful to organizations that develop *SNOMED CT Extensions*.

4.1.3 Extensions

SNOMED CT is designed to allow the *International Edition* to be enhanced by adding *Extensions* that meet national or local requirements. *Extensions* are managed by *IHTSDO Members* or *Affiliates* who have been issued with a *Namespace Identifier*, which distinguishes the *Identifiers* of the *Components* they maintain. An *Extension* may contain *Components* of various types (e.g. *Concepts*, *Descriptions*, *Relationships*, and *Derivatives* including *Reference Sets* used for a variety of purposes).

4.1.3.1 Rationale for Extensions

SNOMED CT is a detailed clinical terminology which covers a broad scope. However, some groups of users will need additional *Concepts*, *Descriptions* or *Reference Sets* to support national, local or organizational needs.

This section explains the structures that enable *IHTSDO Members* (*National Release Centers*) and *IHTSDO Affiliates* to add *Concepts*, *Descriptions*, *Relationships* and *Reference Sets* to *complement* the *SNOMED CT International Release*.

The *Extension* mechanism allows *SNOMED CT* to be adapted to address the terminology needs of a country or organization which are not met by the *International Release*. The mechanism provides a structure within which the components of each *Extension* are uniquely identified and attributed to a specific issuing organization. This ensures that, when instance data containing content from different *Extensions* if communicated, the provenance of each referenced *Concepts* is clear and ambiguity is avoided. Since the *International Release* and all *Extensions* share the same common structure, the same application software can be used to enter, store and process information from different extensions. Similarly, *Reference Sets* can be constructed that refer to content from the *International Release* and a variety of *Extensions*.

The common structure also means that, content developed by one organization can where relevant be easily submitted for possible inclusion in a *National Edition* or in the *International Edition*.

Using the *extension* structure can also help organizations transfer responsibility for terminology to the *IHTSDO* or to another organization , subject to the *terms* of the Affiliate License .

- Local content requirements that are likely to have wider applicability should be submitted to a *National Release Center* for consideration.
- National requirements likely to have International value should be submitted to the *IHTSDO* so they can be considered for inclusion in the *International Edition*.

4.1.3.2 Practical uses of Extensions

An *Extension* mechanism offers many advantages to developers, vendors, terminologists, national bodies and users.

Such a mechanism allows:

- **Users** to access the *SNOMED CT International Release* and one or more *Extensions* through a single *user interface*;
- **Developers** to implement *SNOMED CT Extensions* without developing specialized software;
- **Vendors** to develop and sell products to take advantage of both *International Release* content and *Extensions*;

- **Organizations** to develop and share terminology that meet their business needs, without procuring software;
- ***IHTSDO Affiliates*** to develop terminology that can be shared with other organizations and considered for addition to the *International Release* content;
- ***IHTSDO Affiliates*** to use locally-developed terminology without potential overlap with the work of other organizations .

This structure also enables specialized *Concepts* and *Descriptions* within an *Extension* to be related to *Concepts* and *Descriptions* distributed as part of *SNOMED CT*.

- An *Extension Concept* may be:
  - A national or organizational definition of a *concept*, which is more rigorous or specific than that generally applied to the *SNOMED CT Concept*;
  - An experimental procedure that is not established sufficiently to merit the inclusion in the main body of *SNOMED CT* but which may be in a local controlled study.
- *Extension Descriptions* may be colloquial *synonyms* for a *SNOMED CT Concept* or *descriptions* for an *Extension Concept*.
- *Extension Relationships* may be required to allow analysis packages or decision-support protocols to access additional information about a *SNOMED CT Concept* or to describe *relationships* between *Extension Concepts*:
  - Links between local procedures and relevant administrative actions;
  - Links between local procedures and *SNOMED CT* Procedures.
- *Extension Reference sets* may group *SNOMED CT Concepts* in ways that are specific to data entry contexts of a particular application or communication specification.

The *Concepts*, *Descriptions*, *Relationship* and *Reference Sets* that form an *Extension* must be:

- Distinguishable from the main body of *SNOMED CT*, not only in the thesaurus, but also when stored in a patient record, *query* or decision support protocol;
- Distinguishable from other *Extensions*, in the same way as they are distinguishable from the main body of *SNOMED CT*;
- Able to be distributed and processed in the same way as equivalent *components* from the main body of *SNOMED CT* without requiring specific adaptations of *SNOMED-enabled applications*.

The requirements for *Extensions* can be summarized as follows:

- Support for extra terminology *components* including *Concepts*, *Descriptions*, *Relationships* and *Reference Sets*:
  - These extra *components* behave as though they are *components* of *SNOMED CT* but they are distinguishable from *components* that are part of the *SNOMED CT International Release*.
- Globally unique identification of any terminology *component* that may be used outside the scope of a limited local environment:
  - The mechanism allows several organizations to issue mutually exclusive *Identifiers* for *components* of their *Extensions*.
  - To avoid the risk of misinterpretation, this mechanism is effective in various contexts including:
    - Within the thesaurus;
    - In patient records;
    - In queries, decision-support protocols or knowledge bases.
  - The mechanism indicates when *Concepts* have moved, or are expected to move, between an *Extension* and the *International Release*, or from one *Extension* to another.

- A shared understanding of the responsibility of an organization that creates an *Extension* and provides it for the use of other organizations . These responsibilities include:
  - Maintenance of the *Concept*, *Descriptions*, *Relationships*, and *Reference Sets*;
  - Inactivation of these *components* as appropriate (duplication, ambiguous, outdated, etc.);
  - Submission to an *IHTSDO Member*'s *National Release Centre* for consideration as an addition to a *National Edition* or to the *International Release* content.

4.1.4 Instance data

4.1.4.1 Introduction

This section describes the use of *SNOMED CT* to express clinical ideas in patient records, messages, documents, decision support protocols, queries and other artifacts .

Applications need to create, manipulate and consistently interpret standard *SNOMED CT* representations in instance data to support the entry, storage, retrieval and communication of clinical information.

4.1.4.2 Expressions

An *expression* is a structured combination of one or more *concept identifiers* used to express an instance of a clinical idea.

- ***precoordinated expression***: An *expression* containing a single *concept identifier* is *precoordinated*. The clinical idea it expresses is represented by the identified *concept*. The defining *relationships* of that *concept* precoordinate its meaning.
- ***postcoordinated expression***: *expression* that contains two or more *concept identifiers* is *postcoordinated*. The *concept identifiers* in a *postcoordinated expression* are related to one another in ways that build a more specific clinical idea. The required meaning is expressed by postcoordinating several clinical ideas each of which is represented by an identified *concept*.

  Example: A *postcoordinated expression* can indicate the specific site of a finding even when that specific combination of disorder and site is not represented by a single *SNOMED CT Concept*.

4.1.4.3 Terminology Bindings

Terminology binding is one part of the process of specifying *constraints* on the way that information is structured and represented.

Consistent representation is a prerequisite for effective retrieval and reuse of clinical record information. Requirements for reuse are many and varied, ranging from direct support for the care of the individual patient, through to aggregate analysis for research, statistics and audit. The common theme of these requirements is the need to retrieve particular items of information reliably and consistently, irrespective of the environment in which the data was entered and stored.

Since both the information model and *SNOMED CT* contribute to the processable meaning of an entry in a clinical record it is essential to manage the interdependencies between these two components.

Simple requirements can be addressed by specifying a value-set consisting of the permitted coded values that can be used in a particular field. However, effective representation of clinical records requires a rich information model and an expressive terminology.

Models such as *EN13606* and the *HL7* RIM provide the necessary structural flexibility and *SNOMED CT postcoordinated expressions* provide expressivity. An inevitable side-effect of a richer approach to information representation is an increase in the interdependencies and overlaps between the information model and the terminology. In order to specify and validate consistent representation of meaningful clinical records, *constraints* must be applied to both the information model and terminology. These *constraints* must address all the facets of the model and terminology (e.g. including the use of *postcoordination* and the effect of modeled record structures). The *constraints* on information model and terminology components must be integrated, or bound together, in ways that ensure consistency, avoid ambiguity and minimize the number of different ways in which the same meaning may be expressed.

A terminology binding is an instance of a link between a *terminology component* and an *information model artifact .* Therefore, it is necessary to consider the representation of the required *terminology components* and the way these are associated with relevant *information model artifacts .*

The *information model artifact* to which a *terminology binding* is applied may be a field of a class in a static model or a collection of *fields* of one or more related classes.

Bound components include:

- Information model artifacts :
  - Coded attributes in an *HL7* Version 3 model, an *EN13606* Archetype or in the proprietary information model of an operational application.
- Terminology components:
  - *Constraints* on *SNOMED CT expressions*.

4.1.4.4 Expression Constraints

SNOMED CT contains hundreds of thousands of *Concepts* and this rich resource is greatly expanded by use of *postcoordinated expressions*. In any given situation the range of *Concepts* or *expressions* that are useful, relevant and meaningful is much more limited. This gives rise to a requirement to represent *constraints* on the content or a particular field in a way that can be interpreted and applied by application software.

The simplest *constraint* requirements can be met by specifying the list of valid codes. This requirement is addressed by *subsets* specified using the *Reference Set* mechanism. In some cases, it is useful to express the range of possible values 'intensionally' by specifying rules rather than by listing every member of the set (e.g. to include all *concepts* that are *subtypes* of a specified *concept*).

The use of *postcoordinated expressions* adds further dimensions to the requirement for *constraints*. It may be necessary to specify whether all *postcoordinated refinements* of *concept* are permitted or whether some types of *refinement* are prohibited or required. It may also be necessary to specify whether a *postcoordinated expression* that is equivalent to a permitted value is itself permitted.

Requirements for representing *expression constraints* are closely related to the requirements for representing *query predicates* in queries.

4.1.4.5 Query Predicates

Queries to be applied to *electronic health records* that including *SNOMED CT expressions* may need to represent predicates that test *postcoordinated expressions*. The requirements for representing *postcoordinated expression query predicates* are closely related to the requirements for representing *constraints* on *expressions*. While a *constraint* specifies whether a particular *expression* is permitted in a particular situation, an *expression* predicate specifies the range of candidate *expressions* that match the *query*.

4.2 Logical Abstract Models

This section provides a logical abstract view of *SNOMED CT components* and *derivatives*; and the use of these to represent instances of clinical information. Subsequent sections provide detailed technical *descriptions* of the *SNOMED CT components*, *derivatives* and related artifacts .

4.2.1 Logical Model of SNOMED CT Components

The abstract logical model of *SNOMED CT components* is illustrated by *Figure 11*. The model is centered around the representation of *concepts* and their associated *relationships* and *descriptions*.

Alignment between *release files* and the logical model:

- *SNOMED CT Release Format 2* is closely aligned with the logical model;
- *A mapping table* is provided with the *Release Format 1* file specification to map *RF1* file structures to the abstract model.

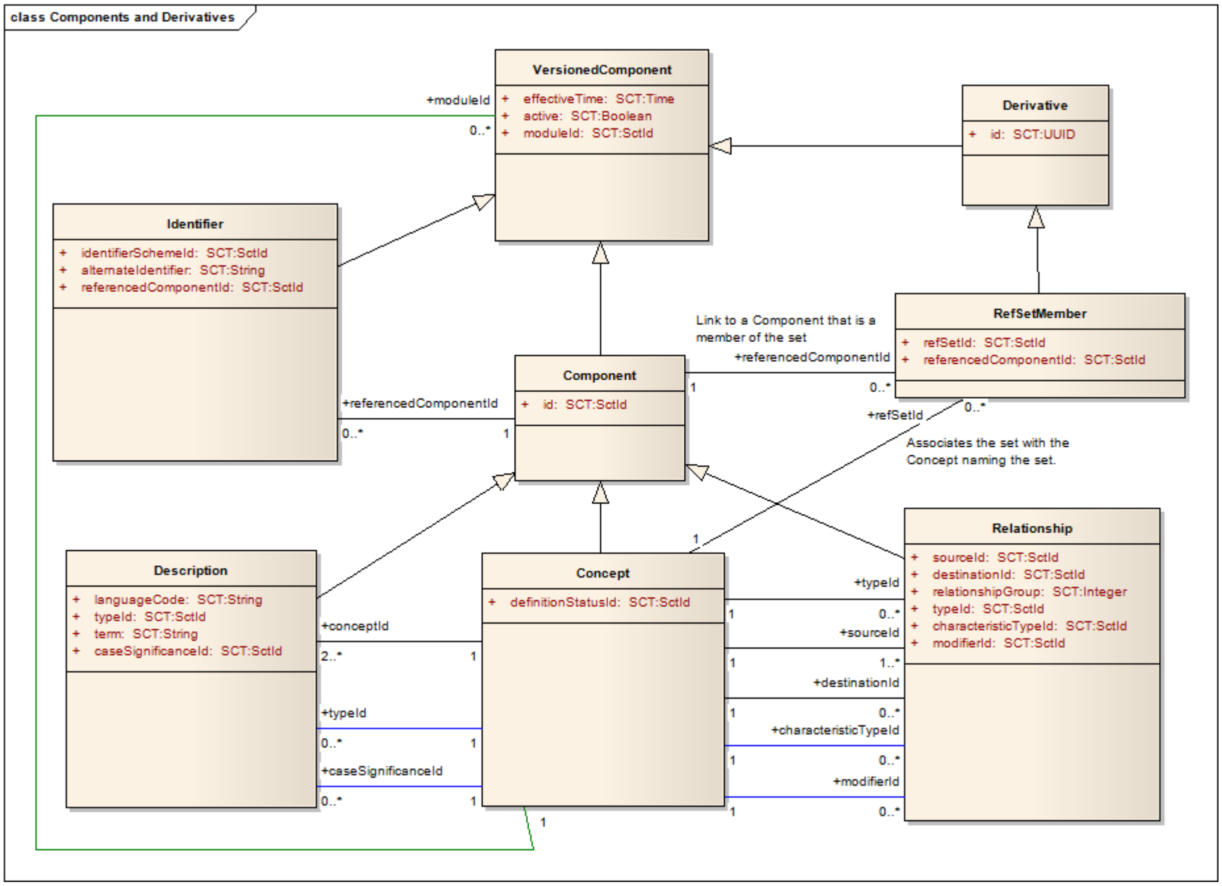


Figure 11: Abstract logical model of SNOMED CT components

4.2.1.1 Descriptions

The set of *terms* that describe a *concept*. These include *fully specified names*, *preferred terms* and *synonyms* in each supported *language*.

4.2.1.2 Relationships and concept definitions

Each *concept* is defined by a set of *relationships* to other *concepts*. The resulting definition may be sufficient to distinguish the *concept* from its parents and siblings in the *subtype hierarchy* in which case the *concept* is considered to be *fully defined*. If the definition is not sufficient to distinguish the *concept* from its parents and siblings, the *concept* is said to be *primitive*. The *concept* contains a field that is set to indicate whether is definition status is *primitive* or *fully defined*.

Figure 12 illustrates the abstract logical model of a *concept*, including the defining *Relationships* between *concepts* (represented by the associations labelled *sourceId*, *destinationId* and *typeId*) and the definition status (represented by the *definitionStatusId*).

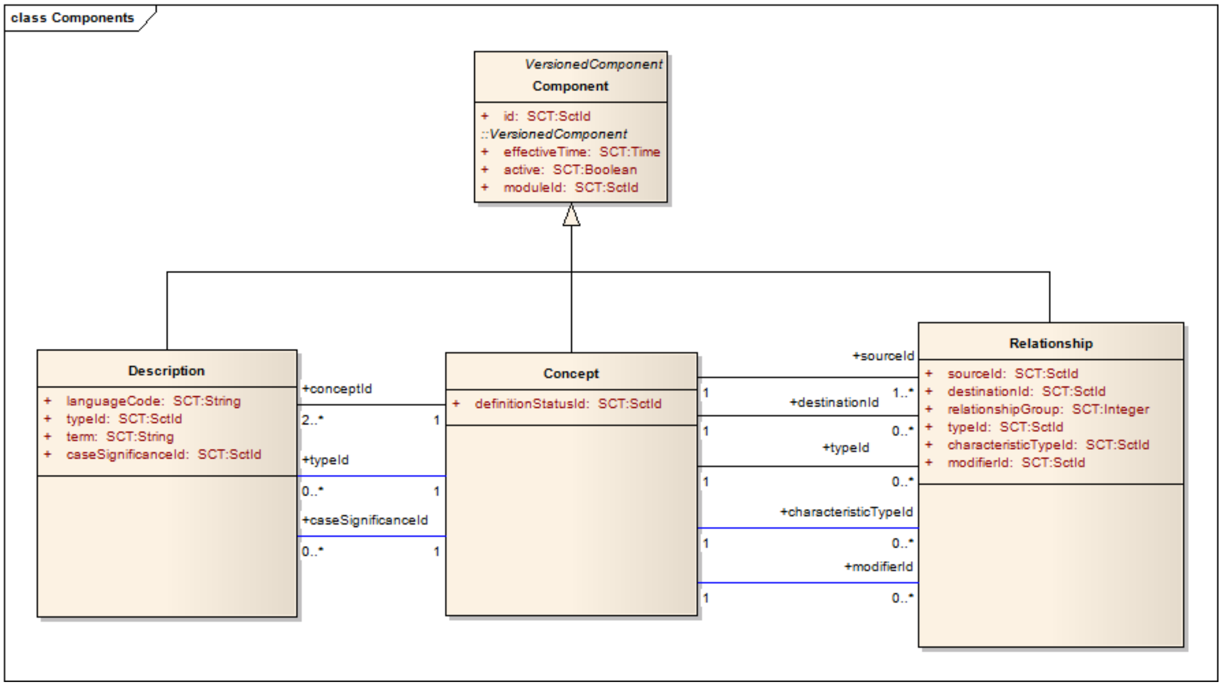


Figure 12: General Abstract Logical Model of a SNOMED CT concept definition

4.2.1.3 Alternative logical abstract model views of concept definitions

The definition of a *concept* can be logically transformed between different views without loss of meaning based on the definitions of related *concepts*.

For example:

Consider the following set of defining *relationships*:

I pain in upper limb I I is a I I pain I
I pain in upper limb I has I finding site I I upper limb structure I
I hand structure I I is a I I upper limb structure I
I Hand pain I I is a I I pain I
I Hand pain I has I finding site I I hand structure I

Based on the above five *relationships* it is possible to infer a new *relationship*:

I Hand pain I I is a I I pain in upper limb I

The definition of I Hand pain I can thus be viewed in three semantically identical forms:

1. As originally stated: :

- I Hand pain I I is a I I pain I and has I finding site I I hand structure I

or

2. With the additional inferred *relationship*:

- I Hand pain I I is a I I pain I and I is a I I pain in upper limb I and has I finding site I I hand structure I

or

3. With the inferred *relationship* but without the redundant stated *relationship* I is a I I pain I: :

- I Hand pain I I is a I I pain in upper limb I and has I finding site I I hand structure I

The *relationship* | is a | | pain | is redundant because this can be determined by traversing the | is a | *relationship* to | pain in upper limb | which in turn is defined as | is a | | pain |.

The result of manipulations like this is that several distinct views of the logical abstract model can be described based on the manner in which they are derived.

Different views of *concept* definitions vary in one or more of the following three dimensions:

- Flattened or nested;
- Stated or inferred;
- Direction and extent of logical *transformation*

These three dimensions are considered in the following subsections of this guide.

4.2.1.3.1 Flat and nested definition views

4.2.1.3.1.1 Flat definition views

In a flat view a *concept definition* consists only of defining *relationships* with target values that are themselves identified *concepts*.

To support this view *concepts* must be created (and defined) for any value that needs to be expressed in the definition of another *concept*.

Example: The | finding site | for the *concept* | pain in left hand | could only be defined by first creating a *concept* | structure of left hand | leading to a definition such as:

> | pain in left hand | has | is a | | pain |.
> | pain in left hand | has | finding site | | structure of left hand |.

The *concept* | structure of left hand | could be defined as follows:

> | structure of left hand | | is a | | hand structure |
> | structure of left hand | has | laterality | | left |.

4.2.1.3.1.2 Nested definition views

In a nested view of a *concept definition* the target value of a defining *relationship* may itself be a nested definition.

This avoids the need for creating intermediate *concepts* but results in more complex definitions.

Example:

The | finding site | for the *concept* | pain in left hand | could be defined without creating the *concept* | structure of left hand | by nesting an appropriate definition as follows:

> | pain in left hand | | is a | | pain |
> | pain in left hand | has | finding site | (| is a | | hand structure | and has | laterality | | left |).

4.2.1.3.1.3 SNOMED CT support for flat and nested definition views

Currently the *SNOMED CT* editing environment works with flat definition views and the standard relational distribution files do not support nested definition views.

Views of *concept definitions* that include nested definitions can be generated from existing *SNOMED CT* data. The proposed *SNOMED CT* XML distribution format does have the potential to support nested views.

Logically the flat form is as expressive as the nested form. The only difference is the need to create and define *concepts* to represent the nested elements in the definition.

Example:

To allow the *concept* | pain in left hand | to be *fully defined* without using a nested definition, | structure of left hand | must exist as a *concept* in *SNOMED CT*.

4.2.1.3.2 Stated and inferred definition views

4.2.1.3.2.1 Stated definition view

A stated *concept definition* is the set of *relationships* (and groups of *relationships*) that an author (*modeler*) has stated to be *defining characteristics* of a *concept*. The *stated view* is maintained in the *SNOMED CT* editing environment and is reviewed and modified during the process of editing a revised edition of *SNOMED CT*.

The *stated view* is distributed in a format similar to the *relationship file*.

4.2.1.3.2.2 Inferred definition views

Inferred *concept definitions* are derived from a stated *concept definition* taking account of the definitions of the *concepts* referred to in the stated definition.

Inferences are derived by applying a consistent set of logical rules to the definition taking account of the definitions of related *concepts*.

Several semantically identical views may be inferred and these are discussed in the following section.

The standard *SNOMED CT* distribution includes the *relationship file* which represents one of the inferred views of the definitions of all *active concepts*.

4.2.1.3.2.3 Alternative inferred definition views

Several semantically identical views may be inferred by applying different logical *transformations* to the *stated view*. Logical *transformations* may vary in the extent to which they normalize the definition and the level of redundancy in the resulting definition.

Different inferred views have properties that optimize different types of function.

The extreme points in the spectrum of possible *concept* definition views are:

- Comprehensive:
  - The set of all defining *relationships* that can be inferred to be true for a *concept* based on the stated definition of this *concept* and the stated definitions of all other directly or indirectly related *concepts*.
- Minimal:
  - The smallest set of defining *relationships* that expresses the definition of the *concept*.

Each inferred view is a combination of a specific *supertype view* (| is a | *relationships*) and an *attribute view* (other defining *relationships*).

4.2.1.3.2.3.1 Supertype aspects of inferred definition views

An inferred definition view includes one of several alternative views of the supertype | is a | *Relationships*. The considerations in this section exclude the *defining characteristics* of a *concept*.

4.2.1.3.2.3.1.1 Comprehensive view of supertype ancestors ("transitive closure")

An inferred *concept definition* view may explicitly contain *relationships* to all supertypes *ancestors* of the defined *concept*.

This comprehensive view of supertypes is known in *description logic* as a "*transitive closure*". It involves traversing (transiting) the target of each | is a | *relationship* to look for and follow further | is a | *relationships* until all paths through the *hierarchy* reach the *root concept* (closure).

This is a highly redundant *expression* of the logical abstract model of a *concept* definition. Applied to the full content of *SNOMED CT* it results in tens of millions of *relationships*.

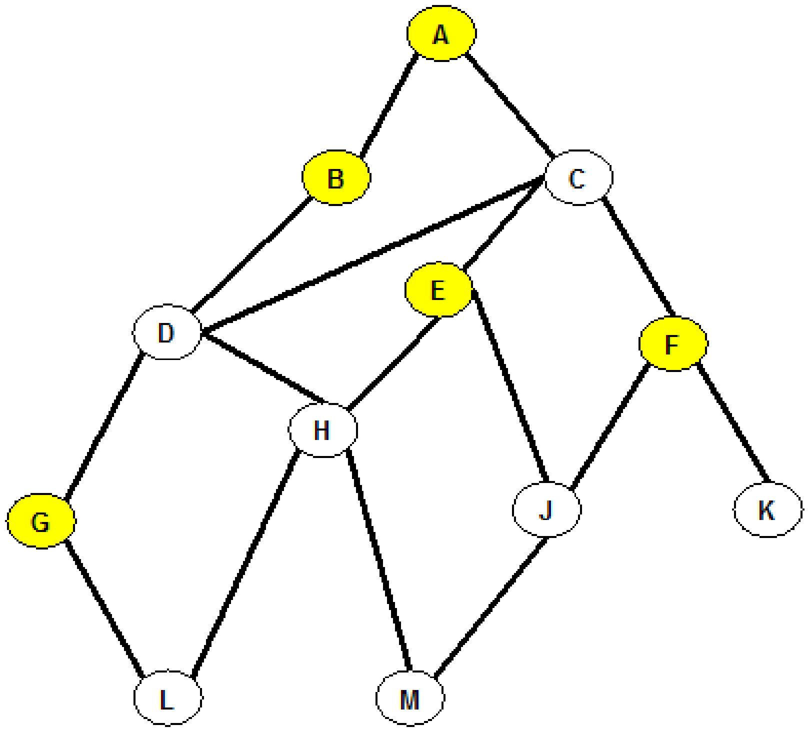


Figure 13: Example hierarchy with list of supertypes in the transitive closure

Table 6: Transitive Closure of Supertypes in the Example Hierarchy

Concept	***Transitive closure* of supertypes**
A	-
B	A
C	A
D	A, B, C
E	A, C
F	A, C
G	A, B, C, D
H	A, B, C, D, E
J	A, C, E, F
K	A, C, F
L	A, B, C, D, E, G, H

Concept	***Transitive closure* of supertypes**
M	A, B, C, D, E, F, H, J

The advantage of this type of view is that there is no need to walk the *hierarchy* tree to answer the question "is *concept* M subsumed by *concept* B". Instead this can be answered simply by checking the *transitive closure* of " *concept* M" for the presence of " *concept* B". Therefore, this view enables high-performance subsumption testing.

Note: Experience suggests that a pre-computed *transitive closure* table out-performs other options and is robust, flexible and easy to implement. Therefore, unless storage capacity is significant concern, this approach is recommended.

4.2.1.3.2.3.1.2 Proximal supertype view (standard distribution view)

An inferred view of a *concept definition* may contain *relationships* to the set of proximate *supertype parents* of that *concept*. *Relationships* with other *supertype ancestors* that can be reached by traversing multiple I is a I *relationships* are omitted.

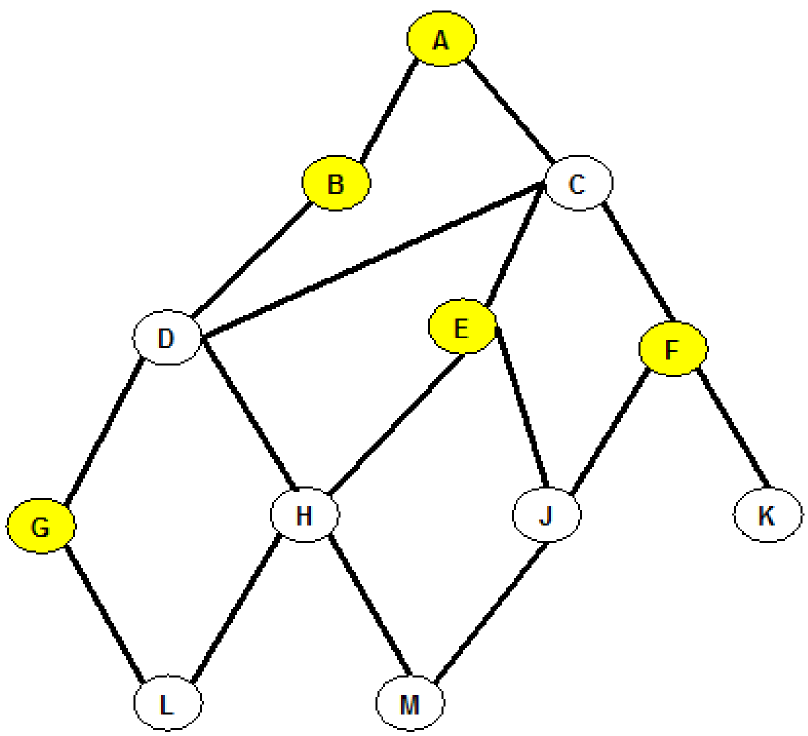


Figure 14: Example hierarchy with list of proximal supertypes

Table 7: Proximal Supertypes in the Example Hierarchy

Concept	**List of proximal supertypes**
A	-
B	A
C	A

Concept	**List of proximal supertypes**
D	B, C
E	C
F	C
G	D
H	D, E
J	E, F
K	F
L	G, H
M	H, J

4.2.1.3.2.3.1.3 Comprehensive primitive supertype view

An inferred view of a *concept definition* may contain *relationships* to all *supertype ancestors* that are "*primitive*" *concepts* (yellow shaded in examples).

The rationale for this is that all the distinguishing features of the "*fully defined*" *concepts* (white unshaded in examples) are represented by other defining *relationships* which will show up in the attribute part of the view.

This view can be used when testing whether a candidate *concept* is subsumed by a predicate *expression*. If the proximal *primitive* supertype view of the predicate *expression* includes any *concept* that is not in the comprehensive *primitive* view of the candidate *concept definition*, then the *concept* is not subsumed by the *expression*.

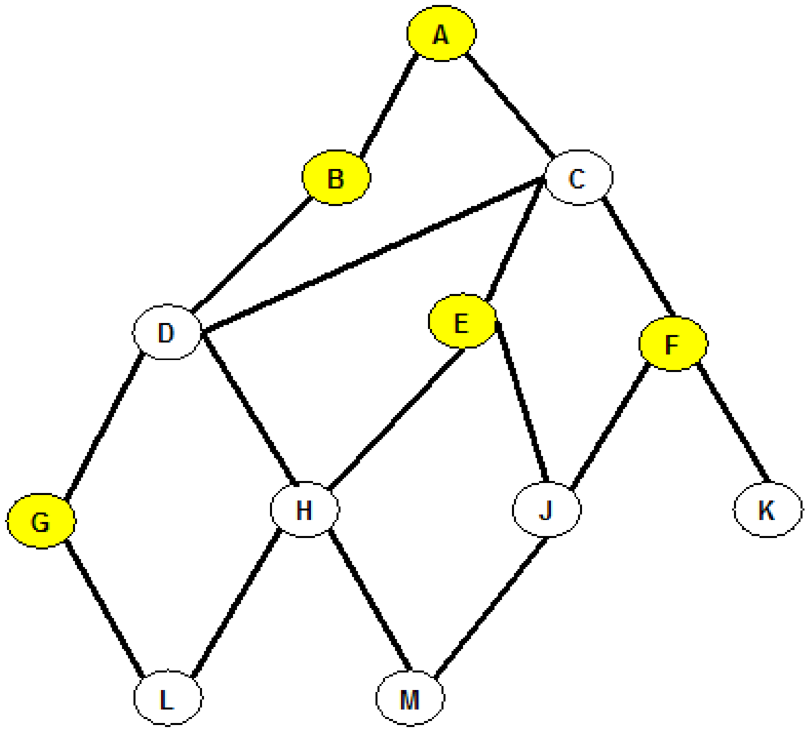


Figure 15: Example hierarchy with comprehensive list of primitive supertypes

Table 8: Primitive Supertypes in the Example Hierarchy

Concept	**Comprehensive list of *primitive* supertypes**
A	A
B	A, B
C	A
D	A, B
E	A, E
F	A, F
G	A, B, G
H	A, B, E
J	A, E, F
K	A, F
L	A, B, E, G

Concept	**Comprehensive list of *primitive* supertypes**
M	A, B, E, F

Note:

1. In this view the definitions of *primitive concepts* should implicitly or explicitly include a reference to the defined *concept* itself. This is because a *primitive concept* expresses some meaning that is not fully distinguished from its supertypes by other defining *relationships*. The reference to self need not be explicitly stored and provided that it is included implicitly at run time.
2. All *active concepts* include the *root concept* in their *transitive closure*. The reference to root need not be explicitly stored provided that it is included implicitly at run time.

4.2.1.3.2.3.1.4 Proximal primitive supertypes (short normal view)

An inferred *concept definition* may contain *relationships* to the set of proximate *primitive supertype parents* of that *concept*. *Relationships* with *fully defined supertype ancestors* are omitted as are *relationships* with *primitive ancestors* that are also supertypes of one of proximate *primitive* supertypes.

This view can be used to test if a candidate *expression* is subsumed by a predicate *concept*. If the proximal *primitive* supertype view of the *concept definition* of the predicate includes <u>any</u> *concept* that is not in the comprehensive *primitive* view of the candidate *expression*, then the *expression* is not subsumed by the *concept*.

The | is a | *relationships* in the *SNOMED CT* '*canonical table*' represent this view.

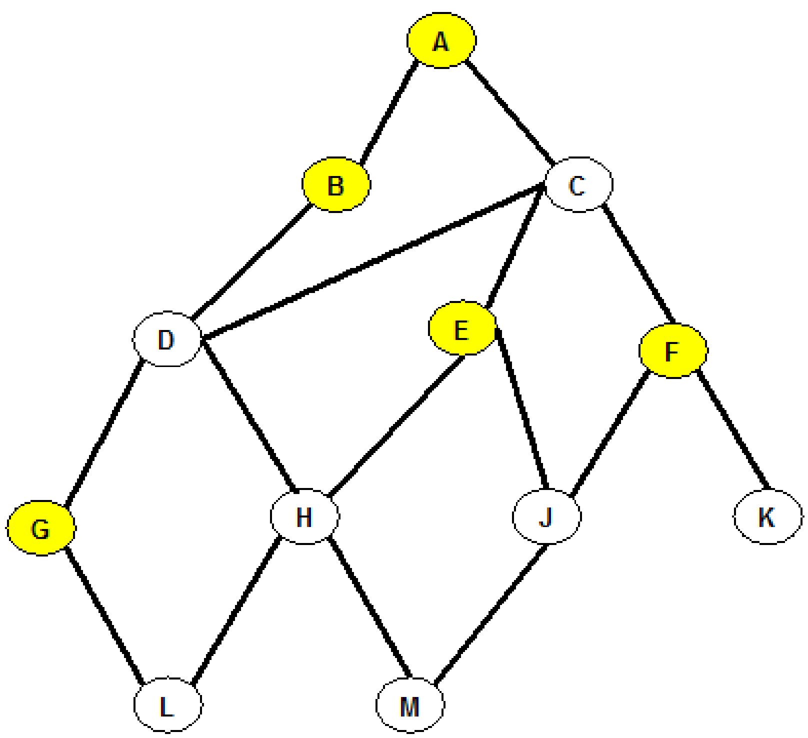


Figure 16: Example hierarchy with list of proximal primitive supertypes

Table 9: Proximal Primitive Supertypes in the Example Hierarchy

Concept	**List of proximal *primitives***
A	A
B	B
C	A
D	B
E	E
F	F
G	G
H	B, E
J	E, F
K	F
L	E, G
M	B, E, F

Note: The proximal *primitive* of a *primitive concept* is the *concept* itself.

4.2.1.3.2.3.2 Attribute aspects of concept definition views

An inferred definition view includes one of several alternative views of the *defining characteristics* of a *concept*. The considerations in this section exclude the supertype I is a I *relationships*.

In addition to the different views described in this section, alternative logical forms may be applied to the values of the *relationships*.

4.2.1.3.2.3.2.1 Comprehensive view of defining Relationships

An inferred *concept definition* may include all the defining *relationships* (and *relationships* groups) that are known to be true. This includes those stated and other inferred by inheritance from stated *supertype ancestors*.

The full form includes all possible *supertype ancestor* values of the stated attributes. This means that in many cases this will include a very large set of *relationships*.

Taken to its logical extreme this also includes *relationships* duplication of *relationships* with *relationship types* that are supertypes of those types stated (e.g. all I procedure site - indirect I *relationships* would be duplicated for the supertype attribute I procedure site I).

While this version of the definition model is an Abstract Logical view it is unlikely that explicit representation of this view will deliver benefits sufficient to merit this level of redundancy.

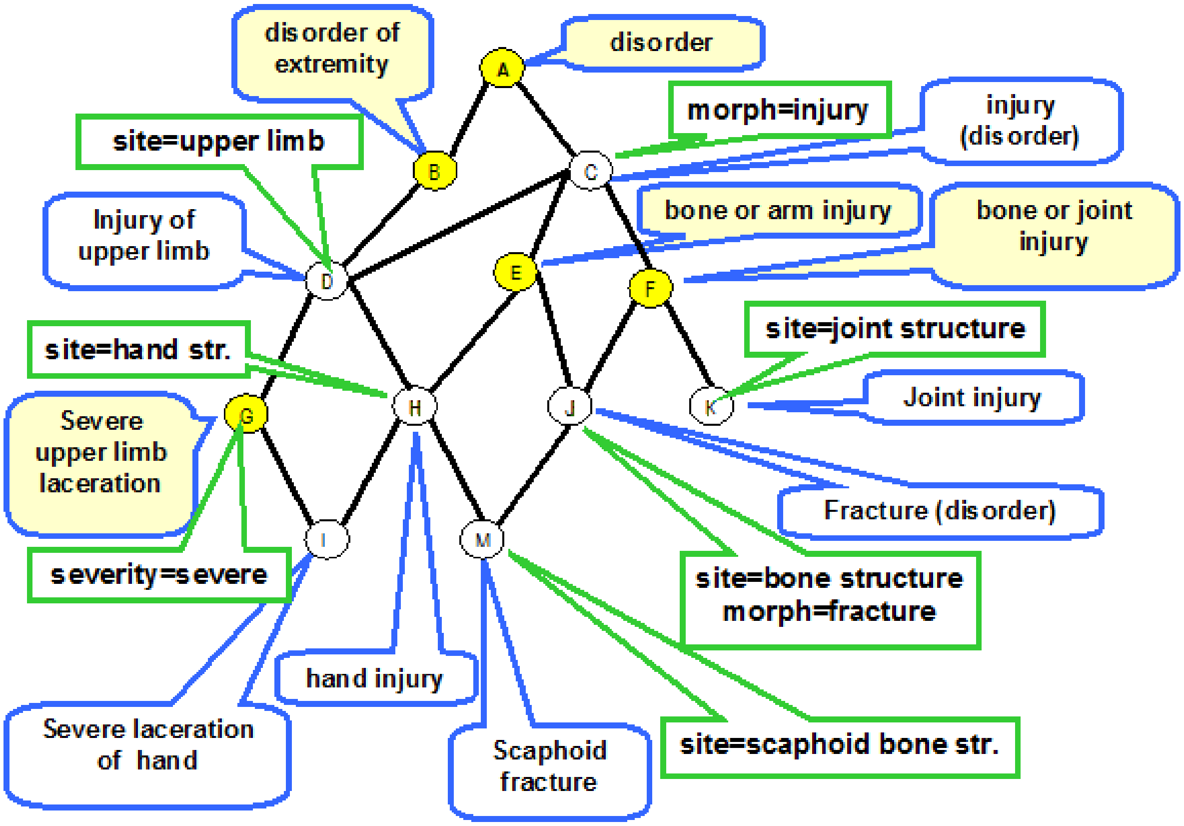


Figure 17: Illustration of sample concepts with differentiating defining characteristics

Table 10: Comprehensive attribute view of sample concepts

See *Figure 17*

C. | Injury disorder | | morphology |=| injury |	**D. | Injury of upper limb |** | site |=| upper limb structure | | morphology |=| injury |
E. | Bone or arm injury | *(primitive)* | morphology |=| injury |	**F. | Bone or joint injury |** *(primitive)* | morphology |=| injury |
G. | Severe upper limb laceration | *(primitive)* | site |-upper limb structure | morphology |=| injury | | severity |=| severe |	**H. | Hand injury |** | site |=| upper limb structure | | site |=| hand structure | | morphology |=| injury |
J. | Fracture (disorder) | | site |=bone structure | morphology |=| injury | | morphology |=| fracture |	**K. | Joint injury |** | site |=| joint structure | | morphology |=| injury |

L. | Severe laceration of hand | | site |=| upper limb structure | | site |=| hand structure | | morphology |=| injury | | severity |=| severe | <u>Note</u> Although the morphology | laceration | is not specified in the example | severe upper limb laceration | refined to the site hand fully defines this *concept*. In a complete view (including supertypes and attributes) this difference is clear.	**M. | Scaphoid fracture |** | site |=| upper limb structure | | site |=| hand structure | | site |=| bone structure | | site |=| scaphoid bone structure | | morphology |=| injury | | morphology |=| fracture |

4.2.1.3.2.3.2.2 Non-redundant defining Relationships ("distribution view")

An inferred *concept definition* may include the set of non-redundant defining *relationships* (and *relationships* groups) that are known to be true. This includes those stated and others inferred by inheritance from stated *supertype ancestors*. However, any *relationships* (or *relationships* groups) that are supertypes of other *relationships* (or *relationship groups*) are redundant and are not included in this view.

A *relationship* that is part of a *relationship group* is only regarded as redundant if the *relationship group* as a whole subsumes another *relationship group*.

This is the view expressed in the standard *SNOMED CT* distribution and this same view also forms part of the long *normal form*.

Table 11: Non-redundant attribute views of sample concepts

See *Figure 17*

C. | Injury disorder | | morphology |=| injury |	**D. | Injury of upper limb |** | site |=| upper limb structure | | morphology |=| injury |
E. | Bone or arm injury | *(primitive)* | morphology |=| injury |	**F. | Bone or joint injury |** *(primitive)* | morphology |=| injury |
G. | Severe upper limb laceration | *(primitive)* | site |=| upper limb structure | | morphology |=| injury | | severity |=| severe |	**H. | Hand injury |** | site |=| hand structure | | morphology |=| injury |
J. | Fracture (disorder) | | site |=| bone structure | | morphology |=| fracture |	**K. | Joint injury |** | site |=| joint structure | | morphology |=| injury |

L. I Severe laceration of hand I I site I=I hand structure I I morphology I=I injury I I severity I=I severe I Note Although the morphology I laceration I is not specified in the example I severe upper limb laceration I refined to the site hand fully defines this *concept*. In a complete view (including supertypes and attributes) this difference is clear.	**M. I Scaphoid fracture I** I site I=I scaphoid bone structure I I morphology I=I fracture I

4.2.1.3.2.3.2.3 Primitive differential attribute view of concept definitions

The *primitive* differential view includes only non-redundant defining *relationships* (and *relationship groups*) that are not present in the sum of the definitions of the set of *primitive* supertype *concepts*. This view provides a minimal attribute view which is semantically complete when combined with one of the *primitive* supertype views.

A *relationship* that is part of a *relationship group* is only regarded as redundant if the *relationship group* as a whole subsumes another *relationship group*.

Table 12: Primitive differential attribute views of sample concepts

See *Figure 17*

C. I Injury disorder I I morphology I=I injury I	**D. I Injury of upper limb I** I site I=I upper limb structure I I morphology I=I injury I
E. I Bone or arm injury I *(primitive)* (I morphology I=I injury I) note 2	**F. I Bone or joint injury I** *(primitive)* (I morphology I=I injury I) note 2
G. I Severe upper limb laceration I *(primitive)* (I site I-I upper limb structure I I morphology I=I injury I I severity I=severe) I note I 2	**H. I Hand injury I** I site I=I hand structure I I morphology I=I injury I
J. I Fracture (disorder) I I site I=I bone structure I I morphology I=I fracture I	**K. I Joint injury I** I site I=I joint structure I I morphology I=I injury I
L. I Severe laceration of hand I I site I=I hand structure I	**M. I Scaphoid fracture I** I site I=I scaphoid bone structure I I morphology I=I fracture I

Note:

1. This is the attribute view expressed in the *SNOMED CT canonical form* table.
2. If the *primitive* supertype view of *primitive concepts* includes the *concept* itself (i.e. as its own proximal *primitive*) then the differential attribute view is empty for all *primitive concepts*. The entries shown above for *primitive concept* apply only where the *concept* itself is excluded from the proximal *primitive* supertype view.

4.2.1.3.2.3.2.4 Supertype differential attribute view of concept definitions

The supertype differential view includes only non-redundant defining *relationships* (and *relationship groups*) that are not present in the sum of the definitions of the supertypes of the *concept*. This view provides a minimal attribute view which is semantically complete when combined with the proximal or complete supertype view.

A *relationship* that is part of a *relationship group* is only regarded as redundant if the *relationship group* as a whole subsumes another *relationship group*.

Table 13: Supertype differential attribute views of sample concepts

See *Figure 17*

C. | Injury disorder | | morphology |=| injury |	**D. | Injury of upper limb |** | site |=| upper limb structure | | morphology |=| injury |
E. | Bone or arm injury | *(primitive)*	**F. | Bone or joint injury |** *(primitive)*
G. | Severe upper limb laceration | *(primitive)* | severity |=| severe |	**H. | Hand injury |** | site |=| hand structure |
J. | Fracture (disorder) | | site |=| bone structure | | morphology |=| fracture |	**K. Joint injury** | site |=| joint structure |
L. | Severe laceration of hand | *None* Note All distinguishing characteristics are inherited from one or both of the supertypes.	**M. | Scaphoid fracture |** | site |=| scaphoid bone structure |

4.2.1.3.2.3.3 The Short Canonical Form

The short *canonical form* is an alternative view of the *Relationships* that is provided as an RF1 *release file*. It consists of the union of the following two views:

- *Proximal primitive supertypes (short normal view)*
- *Primitive differential attribute view of concept definitions*.

4.2.1.3.3 Nature of the definition

A *concept definition* has one of the following two forms:

1. *fully defined concepts* :
   - The definition is complete. It contains *relationships* that represent the full set of *necessary* and *sufficient* conditions.
2. *primitive concepts* :
   - The definition is incomplete. It contains *relationships* that represent a set of *necessary* conditions but this set of conditions is not *sufficient* to fully define the *concept*.

Note: A *necessary* condition is a characteristic that is always true of a *concept*.

Example: | morphology | = | fracture | is a necessary condition of | fracture of femur |.

Note: If all members of a *sufficient* set of conditions are true they imply that the *concept* is also true.

Example: | morphology | = | fracture | and | finding site | = | bone structure of femur | form a *sufficient* set of conditions that define the *concept* | fracture of femur |.

Note: All members of the set of sufficient conditions are also necessary conditions. However, some *necessary* conditions may not form part of the *sufficient* set of conditions.

Example:

Consider the *concept* | gastric ulcer |

- The | finding site | = | gastric mucosa | is a *necessary* condition for | gastric ulcer |:
  - This is true because all gastric ulcers necessarily involve the | gastric mucosa |
- The definition | morphology | = | ulcer | and | finding site | = | stomach structure | is a *sufficient* definition for | gastric ulcer |:
  - This is true because any ulcer in a stomach structure is a | gastric ulcer |.
- Therefore, an assertion that a person has an | ulcer | with | finding site | | stomach | is *sufficient* to imply that they have a | gastric ulcer |:
  - Since a gastric ulcer *necessarily* involves the | gastric mucosa | it should be possible to deduce that a person with an "ulcer" with finding site | stomach | has a disorder of with a site | gastric mucosa |.

4.2.1.3.3.1 Sufficient definition

A *sufficient* definition consists of a set of defining *relationships* (and *relationship groups*) which taken together imply a particular meaning.

The value of a *sufficient* definition is that it allows post coordinated *expression* that is sufficient to define a *concept* to be recognized as equivalent to (or a *subtype* of) a defined *concept*.

For example:

Gastric ulcer is defined as follows and this is a *sufficient* definition because any | ulcer | in a | stomach structure | is by definition a | gastric ulcer |.

116680003 | is a | =64572001 | disease | {116676008 | associated morphology | =56208002 | ulcer | ,363698007 | finding site | =69695003 | stomach structure |}

Based on this definition:

Any *postcoordinated expression* that specified a disease involving an | ulcer | with | finding site | | stomach | would be equivalent to or a *subtype* of | gastric ulcer |.

However, a *query* for all disorders involving | gastric mucosa | would incorrectly exclude the *concept* | gastric ulcer | as the site is not specified as some stomach structure rather then specifically identifying the gastric mucosa.

4.2.1.3.3.2 Necessary definition

A *necessary* definition consists of a set of defining *relationships* (and *relationship groups*) which express all the attributes that are necessarily true about a *concept* for a given version of the *SNOMED CT Concept Model*.

A *necessary* definition may contain *relationships* or *refinements* that are not essential for a *sufficient* definition.

The value of a *necessary* definition is that it allows more refined subsumption queries to be appropriately evaluated.

For example:

Gastric ulcer could be defined as follows:

116680003 | is a | =64572001 | disease | {116676008 | associated morphology | =56208002 | ulcer | ,363698007 | finding site |78653002 | gastric mucous membrane structure |}

This more tightly defined definition contains a *necessary* definition (| finding site | = | gastric mucous membrane structure |). This is necessarily true if the sufficient definition (| finding site | = | stomach structure |) is true, because any ulcer in a stomach structure is by definition a gastric ulcer.

4.2.1.3.3.3 Limitations of the current SNOMED CT model

The current *SNOMED CT* model and distribution format do not distinguish between *relationships* that are *necessary conditions* and those that are part of a set of *necessary and sufficient* conditions. For any *fully defined concepts* the set of defining *relationships* are regarded as *necessary and sufficient*.

As a result of this limitation some currently released *fully defined concept* definitions may include conditions that are *necessarily* true but are not required as part of the set of *sufficient* conditions.

Example: Consider the two definitions shown below:

116680003 | is a |=64572001 | disease | , 246075003 | Causative agent |=113858008 | mycobacterium tuberculosis complex | { 116676008 | associated morphology |=6266001 | granulomatous inflammation | , 363698007 | finding site |=39352004 | joint structure |}

Figure 18: | tuberculous arthritis |

116680003 | is a |=64572001 | disease | , 246075003 | causative agent |=41146007 | bacteria | { 116676008 | associated morphology |=23583003 | inflammation |, 363698007 | finding site |=39352004 | joint structure |}

Figure 19: | bacterial arthritis |

The definition of | tuberculous arthritis | differs from that of | bacterial arthritis | in two respects. In practice the first of these (| causative agent |= | mycobacterium tuberculosis complex |) is sufficient to define the *concept*. However, the nature of the inflammation that results is, necessarily, granulomatous.

Thus an *expression* that specifies | bacterial arthritis | with | causative agent |=| mycobacterium tuberculosis complex | is clinically equivalent to the *concept* | tuberculous arthritis | even though it does not explicitly refine the nature of the inflammation.

In contrast the current *SNOMED CT* model computes | bacterial arthritis | with | causative agent |=| mycobacterium tuberculosis complex | as supertype of | tuberculous arthritis |. This occurs because the *expression* | bacterial arthritis | with | causative agent |=| mycobacterium tuberculosis complex | does not specify of the nature of the inflammation.

Future enhancements: Options for distinguishing the sufficient set of defining *relationships* from those that are merely necessarily true are being investigated. A complete solution to this issue needs to support the recognition of several separate sufficient sets. However, initially a solution recognizing a single sufficient set may be introduced.

4.2.1.3.3.4 Impact on retrieval

A *necessary* definition is inevitably more complete than a *sufficient*definition. From the perspective of retrieval the completeness of a definition is a mixed blessing.

- It is an advantage for candidate *expressions* as they will be subsumed by a wider set of appropriate predicates.
- It is a disadvantage for a predicate *expression*, the necessary conditions may result in incomplete retrieval. A candidate *expression* that satisfies all the *sufficient* conditions should be included. However, it will be excluded unless it satisfies all the necessary conditions in the predicate.

This occurs where the definition of a *concept* states conditions that are *necessarily* true but which go beyond those that are *sufficient* to distinguish a *concept* from its supertypes.

Example:

The *normal form* definition of | pulmonary tuberculosis | is as follows:

116680003 | is a | = 64572001 | disease |

,246075003 | causative agent | = 113858008 | mycobacterium tuberculosis complex |

{116676008 | associated morphology | = 6266001 | granulomatous inflammation |

,363698007 | finding site | = 39607008 | lung structure | }

Used as a *query* predicate, this will exclude valid candidate *expressions* such as ...

233604007 | pneumonia | : 246075003 | causative agent | = 113861009 | mycobacterium tuberculosis |

- This *expression* is not subsumed by the full definition of | pulmonary tuberculosis | because it does not mention "granulomatous inflammation". This type of inflammation is characteristic of "mycobacterium tuberculosis" infection and so is necessarily present. Since currently *SNOMED CT* definitions do not distinguish the sufficient and necessary conditions this cannot be inferred.

A more inclusive *query* predicate that specifies a sufficient set of conditions for | pulmonary tuberculosis | can be constructed by removing the morphology condition.

116680003 | is a | = 64572001 | disease |

,246075003 | causative agent | = 113858008 | mycobacterium tuberculosis complex |

,363698007 | finding site | = 39607008 | lung structure |

- This correctly subsumes both the *precoordinated concept* | pulmonary tuberculosis | and the *postcoordinated* candidate *expression* above.

Note: To ensure complete retrieval

- When selecting a *concept* as part of a *query* predicate, view its *normal form* definition and decide whether some of the conditions should be omitted;
- Specify the minimum set of conditions sufficient for the intended purpose.

Future enhancements: In future, when the *SNOMED CT* model is revised to distinguish *sufficient* sets of defining *Relationships*, the sufficient definition can be used as the predicate for a retrieval. A candidate *expression* matches a predicate if it *necessarily* fulfills all the*sufficient* conditions specified in the *query*.

4.2.2 Logical Model of SNOMED CT expressions

Figure 20 shows the general abstract model for a *SNOMED CT expression*. This diagram also shows the references between *expressions* and components.

An *expression* is a collection of references to one or more *concepts*. The *expression* consists one or more *focus concepts* and an optional *refinement*.

The *focus concept* and the names of the refining attributes are represented by references to *SNOMED CT concepts*. The value of a refining attribute is itself an *expression* and is structured in the same way. Thus nested *expression* can be used to refine the value of a refining attribute.

An *expression* represents an instance of the meaning defined by the defining *relationships* of the *focus concepts* as modified by the *refinements*.

The meaning of each *refinement* is expressed by an *attribute name* which names a property and an *attribute-value* which expresses the value of that property.

- The *attribute name* must be a *concept* that is a *subtype* of |attribute|.
- The *refinement* value may be a *concept* or *expression* that is a appropriate to the named attribute. The values that are appropriate to an attribute are specified by the *Concept Model*. In most cases, any *subtype* of a *concept* that is permitted as a value of an attribute is also permitted.
- *Refinements* may be grouped to represent interdependencies between them in the same way as *relationship groups*.

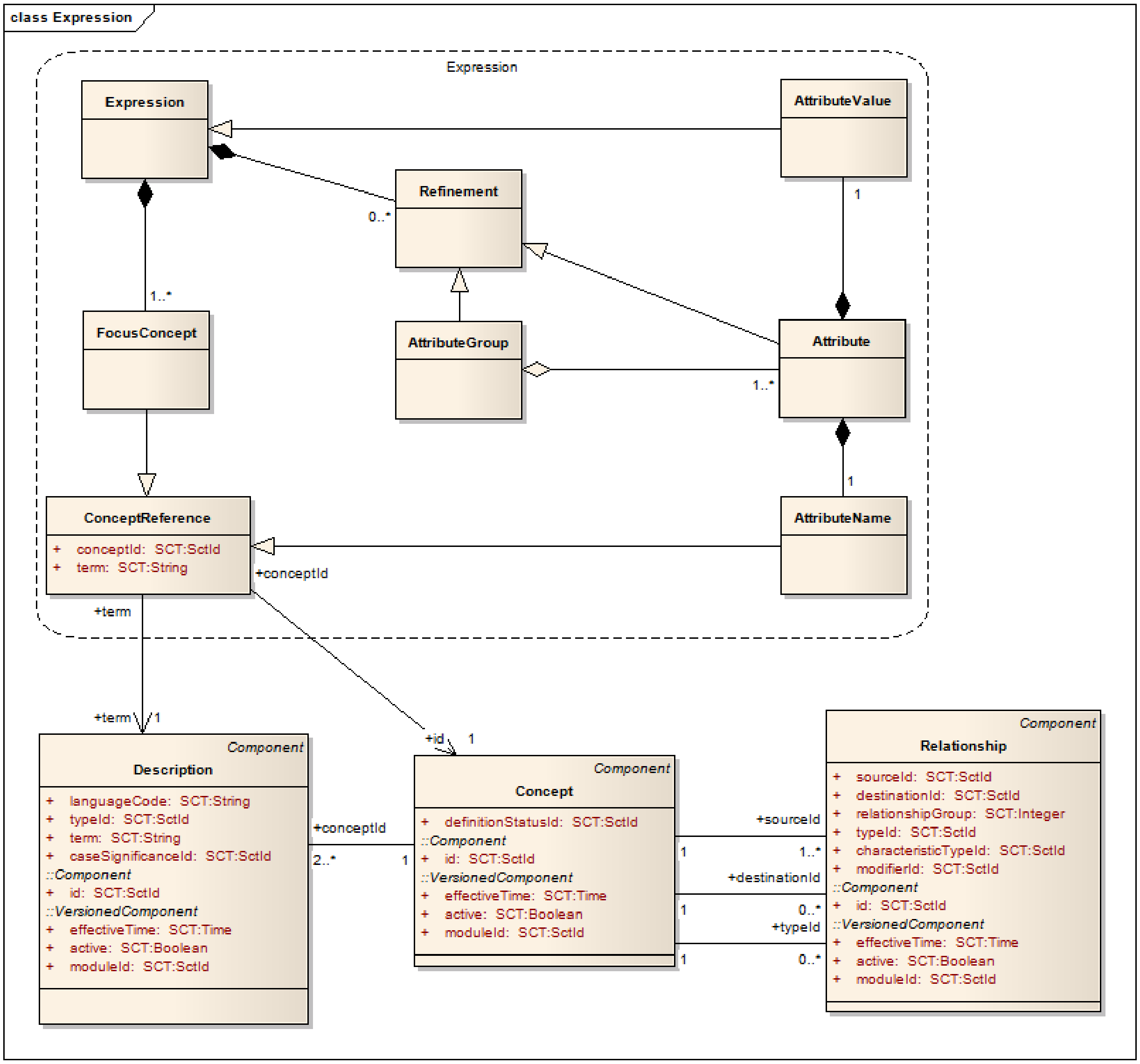


Figure 20: General Abstract Logical Model of a SNOMED CT expression

4.2.2.1 Refinement

An *expression* represents an instance of the meaning of the *focus concept* as modified by *refinements* applied to that *concept*. Various types of *refinement* are possible. Of these some are fully supported by the *SNOMED CT Concept Model* and released data while other possible methods of *refinement* step outside those boundaries.

4.2.2.1.1 Refinement of defining Relationships

4.2.2.1.1.1 Refinement individual attribute values

A defining *relationship* of the base *concept* can be refined by applying a value that is a *subtype* of the defining value.

This approach to *refinement* is fully supported by the *SNOMED CT Concept Model*.

4.2.2.1.1.2 Refinement attribute names

A defining *relationship* of the base *concept* can also be refined by applying a name that is a *subtype* of the defining *attribute name*. For example, if the defining *relationship* specifies a | procedure site | this may be refined to | procedure site - direct | or | procedure site - indirect |.

4.2.2.1.1.3 Refinement of defining Relationship groups

If a *refinement* is applied to one of the defining *relationships* within a *relationship group*, it is the group a whole that is refined.

It is also permissible for a stated (close-to-user) *expression* to refine a *relationship* without grouping the refined *relationship* or without fully enumerating the group of which it is part. In this case, resolution to an inferred structure should apply the ungrouped *relationship* value (or partially enumerated group) as a *refinement* of any group to which that *refinement* can be appropriately applied.

4.2.2.1.1.4 Nested refinement of defining Relationships

The value of a defining *relationship* may itself be refined. In this case the value of the *relationship* becomes a *postcoordinated expression* rather than a *precoordinated concept*.

This occurs most frequently in the following situations:

Laterality refinement

The laterality qualification applies to the value of the | procedure site | or | finding site | *relationship* and is logically nested under site.

(Note lateralization is discussed separately)

Refinement of *situation with explicit contexts*

The | associated finding | or | associated procedure | is a | clinical finding | or | procedure |, which may itself be refined (e.g. with severity).

4.2.2.1.2 Applying values to qualifiers

4.2.2.1.2.1 Applying values to individual qualifiers

A qualifying *relationship* of the base *concept* can be used to apply a *refinement*.
The nature of the allowable *refinement* using *qualifiers* is determined by the refinability of the qualifying *relationship*.

<u>Not refinable</u>

The *qualifier* can only be used to refine the base *concept* by applying the qualifying value specified in the distributed table.

<u>Refinable</u>

The *qualifier* can be used to refine the base *concept* by applying the qualifying value specified in the distributed table or any *subtype* of that value.

<u>Mandatory to refine</u>

The *qualifier* can be used to refine the base *concept* by applying a *subtype* of the qualifying value specified in the distributed table. It cannot be applied with the specified value itself as this is a non-specific grouping value for possible *refinements*.

This approach to *refinement* is fully supported by the *SNOMED CT Concept Model*.

4.2.2.1.2.2 Grouping qualifier refinements

In theory the value of a *qualifier* may apply only to the content of one *relationship group*.

Currently *qualifiers* are not grouped in *SNOMED CT releases* and therefore grouping of *qualifier refinements* is not supported in the current *Concept Model*. However, this is under review and the model may be extended to include grouped *qualifiers* in future. This review is required because problems arise with subsumption testing where *precoordinated* definitions include grouped *attribute-value pairs* and the *expression* uses an ungrouped, *qualifier* derived, attribute.

4.2.2.1.2.3 Nested refinement of qualifiers

The value of a *qualifier* may itself be refined and represented as an *expression* rather than a *precoordinated concept*.

This occurs most frequently with *expressions* which qualify high level "*situation with explicit context*" *concepts* (e.g. "finding with *explicit context*"). In this case the | associated finding | is applied as a *qualifier* which may itself be refined (e.g. with severity).

4.2.2.1.3 Applying laterality to a concept

A laterality value (left, right or bilateral) can be applied as a *qualifier* to lateralizable body structure *concepts*.

It is also permissible for a stated (close-to-user) *expression* to lateralize a base *concept* that has a definition including reference to a lateralizable body structure. In this case, resolution to an inferred structure should apply the laterality to all values in the base *concept definition* that are lateralizable body structures.

This approach is fully supported by the *SNOMED CT Concept Model*, provided that appropriate *transforms* are applied.

Note

If lateralization is specific to particular aspects of the *concept* then the laterality should be applied to the appropriate *relationship* as part of a nested *expression*.

4.2.2.1.4 Sanctioned and unsanctioned refinement

4.2.2.1.4.1 Introduction to refinement sanctioning

SNOMED CT relationships provide information that may be used to determine the types of *refinements* can be processed to determine *equivalence* and subsumption. However, even where a *concept* has no specific *relationship* it is possible to apply a refinement using an *attribute* that the *Concept Model* permits for *concepts* in that domain. Other *attributes* are not recommended for refinement as they will result in *expression* that cannot be normalized or reliably compared. Specific issues with unsanctioned refinements are considered in:

- *Unsanctioned use of "Concept Model attributes"*;
- *Use of "unapproved attributes"*;
- *Advantages and disadvantages of unsanctioned refinements*.

4.2.2.1.4.2 Unsanctioned use of "Concept Model attributes"

In some situations it may seem to be useful to use one of the attributes used in the *SNOMED CT Concept Model* to refine a *concept* that does not have a defining *relationship* or *qualifier* named by this attribute.

Provided that this is limited to qualifications that the *Concept Model* specifies for *concepts* of the same general type this approach can be applied. However, *Concept Model* attributes should not be applied to *concepts* of other types (for example the "approach"attribute should not be applied to a "finding"). Use of unsanctioned (but 'allowable') attributes for *refinement* may limit semantic interoperability.

Despite this limitation it may be appropriate to use a community agreed approach for a particular defined purposes. However, care should be taken to use attributes only in the manner described in the *Concept Model Guide (6)*.

4.2.2.1.4.3 Use of "unapproved attributes"

The *SNOMED CT release* also includes a large number of attributes that are classified as "unapproved attributes".

Most of these originate from earlier terminology efforts. They have as yet not been applied in the *SNOMED CT Concept Model* and there is no guarantee that they will be used in a particular manner in the future.

This approach is not supported by the *SNOMED CT Concept Model*. Therefore any use of unapproved attributes for *refinement* is likely to limit semantic interoperability.

Despite these limitations, it may be appropriate to use a community agreed *Reference Set* of unapproved attributes within a defined user community for a particular defined purpose. Any such use should be fully documented by those responsible for its adoption. In the future as the *SNOMED CT Concept Model* evolves, additional supported attributes may provide a *migration* path for information recorded using a well-documented set of rules for a limited set of use cases.

4.2.2.1.4.4 Advantages and disadvantages of unsanctioned refinements

Note: THIS SECTION CONTAINS DISCUSSION NOTES ONLY.

The presence of defining or qualifying *relationships* certainly simplifies the task of implementing facilities for *refinement*. It also provides an indication that subsumption and *equivalence* computation may be possible. However, at this stage there is no definitive view of the extent to which *SNOMED CT* should sanction and permit particular *refinements* while deprecating or prohibiting other *refinements*.

Disadvantages of prohibition of all unsanctioned *refinements*

- **Lack of ability to express some required meanings**:
  - Until an attribute is included in the *Concept Model* and appropriately populated for all relevant *concepts*, it cannot be used to refine some *concepts* that might reasonably be so refined. The consequence of this are an inability to express some meanings required by users with approved *SNOMED CT expressions*.
- One example of this is that at present the following *expression* would not be sanctioned as headache has no associated severity *qualifier*. While this looks like an error that could readily be corrected it serves to illustrate the point.

25064002 | headache |:246112005 | severity |=24484000 | severe |

Disadvantage of allowing unconstrained *refinement*

- **Nonsense *expressions* with no "sensible" meaning**:
  - e.g. 25064002 | headache |:103366001 | with color |=414497003 | infra-red |
  - These are probably not a major cause for concern because it is impossible to create a foolproof approach that guarantees that all *expressions* will be sensible:
    - The following nonsense example is "sanctioned" in the sense that the site specified is a *refinement* of | head structure | which is the defined finding site for | headache |:
      - 25064002 | headache |: 363698007 | finding site |= 87056002 | infantile diploetic mastoid cell |
  - A nonsense *expression* is meaningless and where it is subsumed is largely irrelevant. Ideally it would subsume under nonsense *expressions* but that would require a knowledge of the rationality of all possible *expressions*.
  - In the absence of a tractable way to prohibit nonsense, avoidance and management of nonsense is an issue for implementers, users and qualify reviewers.
- **Nonsense *expressions* which may express a superficial "sensible" meaning**:
  - e.g. 25064002 | headache |:103366001 | with color |=301888000 | pale color |
  - A person reading this might think this expresses the fact the person's head (or face) was pale at the time of the headache. Logically in *SNOMED CT* it would mean that the headache is pale in color which is nonsense. However, an argument could be advanced that the same rules apply as those for indirect laterality and thus this could *transform* to:
    - 25064002 | aching pain |: 363698007 | finding site |=(69536005 | head structure |:103366001 | with color |=301888000 | pale color |).
  - This is still nonsense from a *SNOMED CT* perspective or perhaps it could correctly mean is | aching pain in the pale colored head structure |. However, if the author (or authoring application) assigned such an *expression* to represent two distinct findings | headache | and "head is pale in color " this meaning would not be apparent from a logical computational perspective.

- While prohibition of nonsense is not tractable it may be feasible to state rules that express which forms of *expression* are logical and computable. Furthermore the outcome of these rules needs to be deterministic so that the result of transforming do not differ according to implementation.

- **Alternative rational *expressions* of similar meanings** :
  - Consider the following:
    1. 25064002 | headache |: 279114001 | character of pain |=410704005 | throbbing sensation quality |
    2. 162308004 | throbbing headache |
    3. *code*=162306000 | headache character | *value* =410704005 | throbbing sensation quality |
       - This assumes an information model with an observable entity *concept* naming a value in a separate information model attribute (*HL7* Observation supports this).
    4. 29695002 | throbbing pain |:363698007 | finding site |=69536005 | head structure |
    5. 25064002 | headache |:162306000 | headache character |=410704005 | throbbing sensation quality |.
  - All these *expressions* appear rational but only options 2 and 4 have the same *normal form* in the present *SNOMED CT Concept Model*.
  - Potentially option 3 could also be computed if both (a) the information model *terminology model* interface was clear and (b) the *SNOMED CT* definition of 162308004 | throbbing headache | is enhanced to add "363713009|has interpretation|=410704005|throbbing sensation quality|.
  - On the other hand option 1 is more in line with the way disorders are refined by "severity" and other qualitative *refinements*. For this to be computable equivalent the *concepts* "29695002|throbbing pain|" and 162308004 | throbbing headache | would both need revised definitions in which they were defined as having "279114001|character of pain|=410704005|throbbing sensation quality|".
  - Option 5 also looks superficially reasonable and shares the general feel of option 3. However, since 162306000 | headache character | is an "observable entity" rather than an "attribute" this representation would be contrary to one fundamental principle of *refinement* - that the name of the *refinement* should be a *subtype* of the *concept*"attribute". This means current normal *transform* rules would not result in a proper *normal form* and indeed might reasonable report an error.

- ***User interface* design issues** :
  - Given all of the above points, application designers will struggle to create sensible and consistent interfaces unless advice on sanctioning is provided.
  - Different issues will apply according to the nature of the interface. For example this may include:
    - What options to offer users to allow *refinement* of specific *concepts*;
    - How to represent the meaning that results from selecting options on a structured data entry form as a *SNOMED CT expression*;
    - How to encode meaning derived from *natural language processing*.

Interim recommendations

1. Wherever refining an existing defining or qualifying *relationship* enables representation of the required meaning this approach should be preferred.
2. Where 1 does not meet the requirements any attribute which is used in the *concept model* for *concepts* of the same type may be applied. The value applied to the attribute must be one of the allowable values as specified for that attribute in the *Concept Model Guide (6)*:
   - For example a | causative agent | attribute can be applied to a clinical finding *concept*. The value assigned to this attribute is a value assigned from | Organism |, "physical force", "physical object" or | substance |. However, | causative agent | cannot be applied to refine a procedure. Furthermore the value of the | causative agent | cannot be a procedure or disorder.

3. Where neither 1 nor 2 meet the requirement use of additional attributes or values may be considered to meet a specific requirement. However, in this case, the implementer and/or user community will need to:

- Avoid a direct conflict with other uses of the same attribute.
  - Ambiguity will arise if an existing attribute is overloaded to fulfill a different use-case:

    Example: The |laterality| attribute is used in the *concept model* to specify which of two functionally symmetrical paired structures is involved (e.g. "left wrist", "right kidney"). It should not be used for:

    - non-symmetrical structures (e.g. heart structures where the use of |left| and |right| refers to functionally different structures).
    - right or left side of a midline structure (e.g. |head| :|laterality| = |left| does not mean the "left side of the head" it means "left head" - and is thus not a useful *refinement*).
    - relative laterality (e.g. |trachea| : |laterality| = |left| does not mean "to the left of the trachea" or "trachea deviated to the left" it means "left trachea" - and is not a useful *refinement*).

- Agree the approach to be taken in advance:
  - Ad-hoc *refinement* by end-users without any guidance on an agreed approach is liable to lead to multiple ways of representing the same required meaning and a loss of interoperability.

- Document the approach taken in forms that:
  - Allow consistent use within the community;
  - Identify any issues related to computation of *equivalence* and subsumption between these local variant *expressions* and the content of *SNOMED CT*;
  - Are communicated to an appropriate *SNOMED CT* Working Group to help establish a wider consensus.

- Make provision for future *migration* of data as a common *SNOMED CT* approach is developed in future.

Note: Within the UK, *NHS* Connecting for Health has issued guidance on post coordination which specifies *constraints* on allowable *refinements* and adds some specific *extensions* to the *refinements* sanctioned by released *relationships*. These guidance documents are available to implementers in the UK.

4.2.2.1.5 Applying values to concepts

Information model attributes such as values applied to an observable, also effectively refine the meaning of the *concept* as used in the record.

Currently the *SNOMED CT Concept Model* does not address issues of *equivalence* between a particular value applied to an observable or measurement procedure and a potentially similar finding (e.g. |creatinine measurement, serum| with a specified value and a finding such as |serum creatinine raised|).

There is a loose approximation using the |interprets| and |has interpretation| *Relationships* between some |clinical finding| *concepts* and relevant |observable entity| or |laboratory procedure| *concepts*.

Example: |serum creatinine raised| has a definition that includes:

- |interprets|=|creatinine measurement, serum|
- |has interpretation|=|above reference range|

Future enhancements: The *relationships* between |observable entity|, |laboratory procedure|, |evaluation procedure| and |clinical finding| *concepts* are currently under review.

4.2.2.2 Modeling semantic context

When a clinical finding is mentioned in a patient record certain assumptions are usually made about what it means in relation to the person who is the subject of that record. Thus if the finding | wheezing | is present in a record it is assumed to mean that the subject of that record is wheezing at the time of examination. This assumed meaning might be stated in full "the subject of the record is currently wheezing" but a contracted form that omits explicit reference to the subject, timing and presence of the finding is more usual in written records.

Similarly when a procedure is mentioned in a patient record assumptions are usually made about what it means in relation to the subject of that record. Thus, in the absence of other information, the mention of the procedure | cholecystectomy | may be assumed to mean that a "the subject of the record had a cholecystectomy at a stated time ".

Although default assumptions such as those above may be made, it is also possible for mention of the same finding or procedure to have a very different meaning. For example, "past medical history of wheezing", "not wheezing", "father suffers from wheezing", | cholecystectomy planned |, | cholecystectomy not done |.

The *SNOMED CT* context model provides a way to model *concepts* that explicitly state the *clinical situation* in which they are used. This same model also allows the construction of *expressions* that explicitly state the *clinical situation* in which a *concept* is being used in a particular record.

A proprietary record structure or a *reference information model* may also express aspects of context and these can be mapped to the *SNOMED CT* context model where appropriate to create comparable *expressions*.

The context model also specifies a default context that applies to findings and procedures which are expressed in a patient record without any explicit statement of context.

The most important aspects of the context model are those which have the potential to express a meaning that differs fundamentally from the meaning associated with the default context. Changes to context that have this fundamental effect on meaning are referred to as "axis modifying". The phrase "axis modifying" indicates a change that shifts the meaning between different axes in the *subtype hierarchy*.

The context model allows "axis modification" to be expressed within the general abstract logical model applied to all *SNOMED CT concepts*. To achieve this a *concept* such as | FH: Diabetes mellitus | is modeled as a *subtype* of | family history of disorder |. It is not a *subtype* of | diabetes mellitus | but instead its association with the finding | diabetes mellitus | is modeled using a defining *relationship* | associated finding |. Similarly a | Hip replacement planned | is a *subtype* of | planned procedure | (not a *subtype* of "hip replacement"). It is related to "hip replacement" by an | associated procedure |*relationship*.

4.2.2.3 Alternative logical abstract model views of expressions

Like a *concept*, an *expression* may be logically transformed into a variety of different views taking account of the definitions of the *concepts* which it references (i.e. the *Concept Identifiers* included in the *expression*).

4.2.2.3.1 Close-to-user expression view ("stated")

The close-to-user (or "stated") view of an *expression* contains references to the *concept* (or combination of *concepts*) together with *refinements* as selected by the user or as encoded by a clinical application to represent the semantics of a single clinical statement (i.e. a discrete clinical record entry).

The close-to-user view of an *expression* is the faithful representation of the information entered. For clinical safety and accountability purposes this should be regarded as the primary stored and communicated view of clinical information encoded using *SNOMED CT*.

Note: This view includes *refinements* applied by an application based on selections made in an entry form as well as those made explicitly. It does not include any *relationships* that are added based on *classifier* rules to make the *expression* complete or to normalize it.

4.2.2.3.2 Inferred expression views

An inferred *expression* can be derived from a stated *expression* by applying rules that take account of the definition of the refined *concept* and the associated refined values.

Inferences are drawn based on a consistent set of logical rules applied to the *expression* taking account of the definitions of *concepts* referenced by the *expression*.

Alternative semantically identical *expressions* may be generated using different logical *transformations*. The purpose of logical *transformations* is to support accurate and complete information retrieval through subsumption testing.

In general *terms* the types of *transformation* and resulting inferred views for *expression* are similar to those for *concept definitions*. The following sections of the guide identify some of inferred *expression* views and some of the differences between *expressions* and *concept definitions*.

4.2.2.3.3 Simple, nested and grouped expressions

A typical close-to-user *expression* consist of a single *concept* modified by optional *refinements* as shown in *Figure 21*. This may look like a *concept definition* but it is not defining the *concept* | hand pain |, it is specifying a more specific meaning by refining the | finding site | of the *concept* | hand pain | and adding a severity *qualifier*.

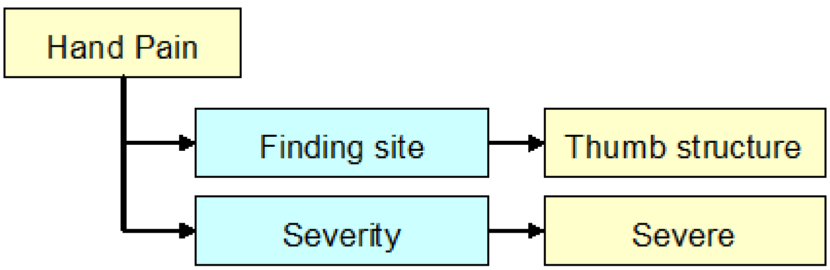


Figure 21: Refining a concept to add specificity

The target of a *refinement* may itself be refined producing a nested structure. An example of this is the application of laterality to finding site as shown in *Figure 22*.

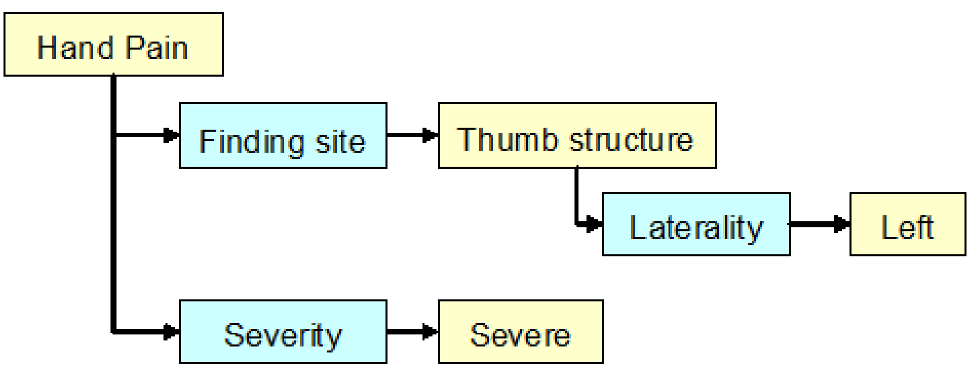


Figure 22: Nested refinement applied to a body site

In some cases, *refinements* within an *expression* may be grouped to represent association between a two different *refinements*. For example, a method and a target site or device as shown in *Figure 23*

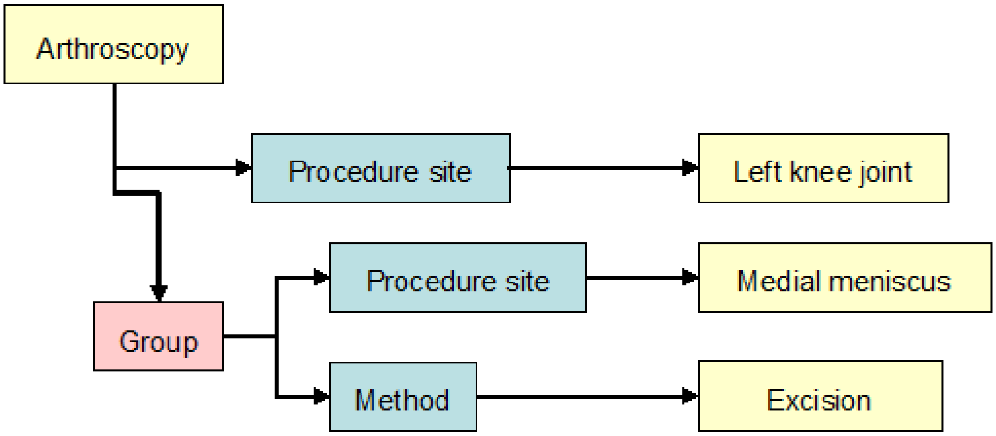


Figure 23: Grouped refinement

4.2.2.3.4 Expressions with multiple focus concepts

Some *expressions* may have multiple *concepts* followed by optional *refinements* as shown in *Figure 24*

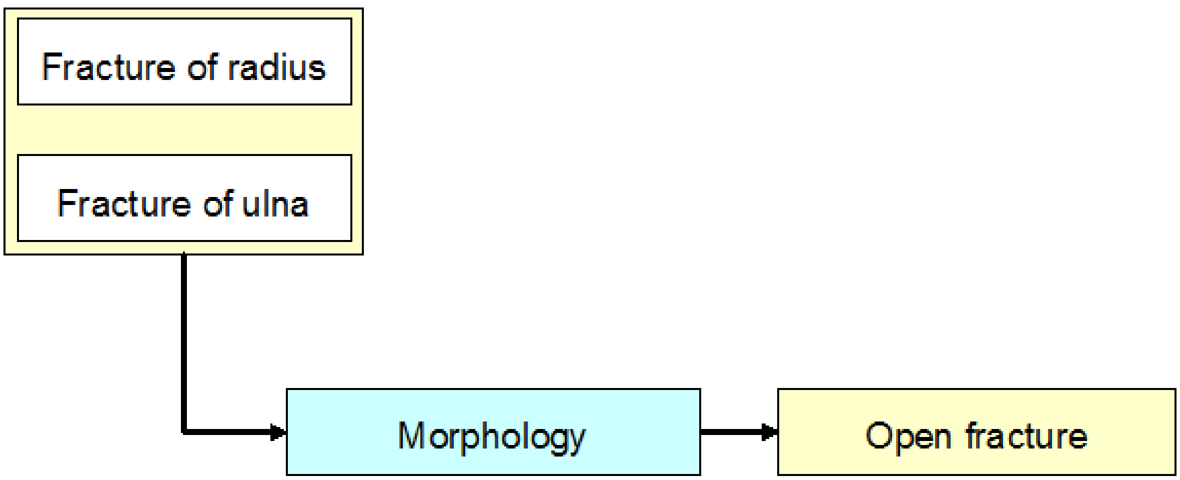


Figure 24: An expression with two focus concepts

The base of an *expression* may thus be one or more supertype *concepts* that are combined to produce a single meaning.

It is important to note that combining *concepts* at this level presumes that the result is intended to be a single combined meaning which is subsumed by the meaning of the combined *concepts*. Furthermore, the same *refinements* apply to the combined meaning of this set of *concepts*.

Some representational forms (e.g. *HL7* version 3 *Concept* Descriptor data type) do not allow combinations to be expressed in this way. However, it is possible to apply a simple logical *transformation* to create a semantically identical view that can be conveyed in a syntax that supports a single *focus concept* with *refinements* (see *Figure 25*).

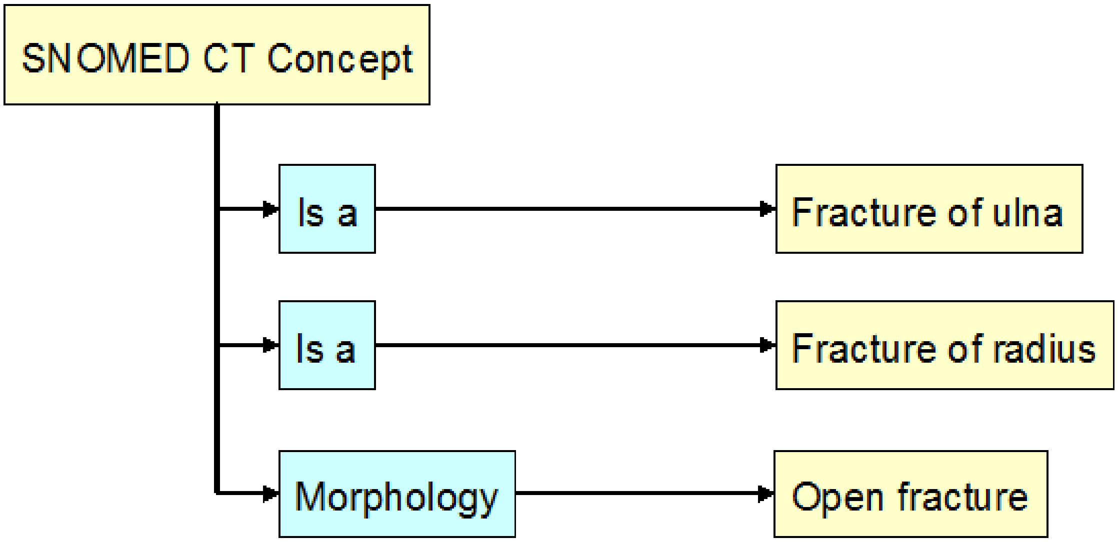


Figure 25: An alternative view of an expression with two focus concepts

4.2.2.3.5 Expressions that include context

Expressions may also explicitly represent the semantic context surrounding a finding or procedure. In these cases, the finding or procedure is nested inside the context component of the *expression*. The outer layer of the *expression*, which expresses the context, is sometimes referred to as the *context wrapper*. The nested *expression* representing the finding or procedure is sometimes referred to as the "clinical kernel".

Figure 26 illustrates how the general *concept* 281666001 I family history of disorder I can be refined to represent family history of a specific condition.

Figure 27 illustrates an alternative (computationally equivalent) representation of the same situation. In this case the family history situation is itself represented by an *expression*.

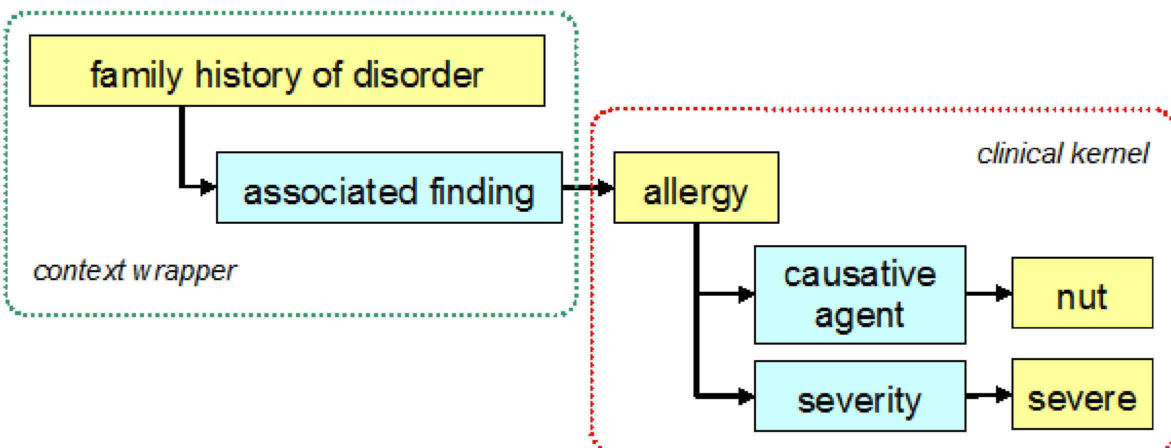


Figure 26: Family history of a specific type of severe allergy to nuts as close-to-user form expression

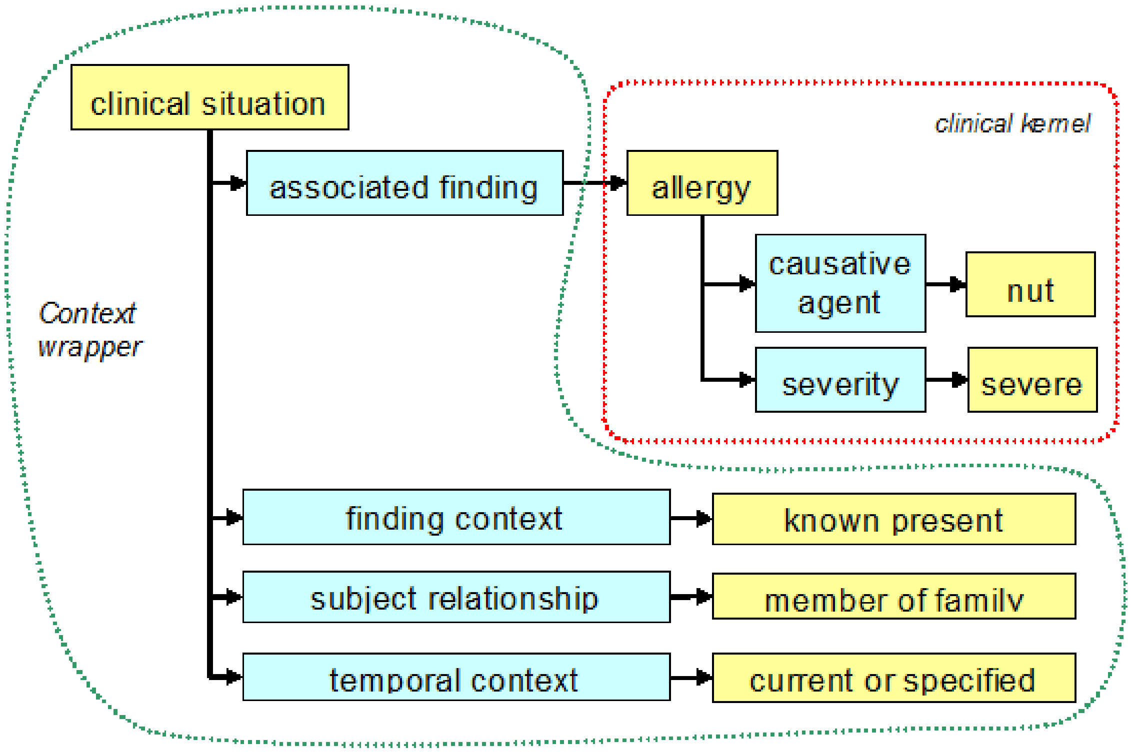


Figure 27: Family history of severe allergy to nuts represented by using a context wrapper expression

4.2.2.3.6 Normal form expression view

The theoretical range of equivalent *expressions* for a single idea includes two end-points:

- A fully *precoordinated expression* in which a single *concept identifier* is used to represent the idea;
- A maximally *postcoordinated expression* in which every facet of the idea is separately represented by an *attribute-value pair*.

In between these end points are a variable number of equivalent partially *postcoordinated expressions*.

Example: For a detailed example see *Example of equivalent postcoordinated expressions* on page 82

In order to compare *expressions*, is useful to be able to transform from these varied *expression* into a common *normal form expression*. This is possible using the combination of the *expression* and the definitions of the *concepts* to which it refers. A long as a reference *concept* is *fully defined* the defining *Relationships* for that *concept* can replace the *concept identifier* in the *expression*. This process reveals redundancies that can be removed by merging the definitions with the *expression*. An end-point is reached when all the *concepts* referenced by the *expression* are *primitive*. This is referred to as the *normal form*.

The process of normalization of *expressions* is described in detail in *Transforming expressions to normal forms* on page 575.

Note: The most important requirements for logical transformation of *expressions* is to enable information entered (in a close-to-user view) to be readily tested for *equivalence* or subsumption against another *expression* or to test inclusion within constrained range of values.

4.2.2.3.6.1 Example of equivalent postcoordinated expressions

To illustrate the range of possible equivalent *expressions* *Table 14* shows the *defining characteristics* of the hypothetical [3] *concept* "red steel pedal bike" and its *supertype ancestors*.

Table 14: Definitions of concepts used in illustration of alternative representations

Id	*Concept*	*Defining Characteristics*
1	Device (*PRIMITIVE*)	| is a | = Thing
2	Metal device	| is a | = Device Made of = Metal
3	Transport device	| is a | = Device Used as = Transport
4	Steel transport device	| is a | = Transport device | is a | = Metal device Made of = Steel Used as = Transport
5	Pedal powered transport device	| is a | = Transport device Used as = Transport Used as = Transport Power = Pedals
6	Bicycle (*PRIMITIVE*)	| is a | = Transport device Used as = Transport Moves on = 2 wheels
7	Pedal bicycle	| is a | = Pedal powered transport device | is a | = Bicycle Used as = Transport Moves on = 2 wheels Power = Pedals

[3] This hypothetical concept is chosen in preference to a real SNOMED CT concept to allow illustration of theoretical points with simple qualifiers. While all the points illustrated apply to some SNOMED CT concepts but there is no single concept that readily illustrates all these points without introducing other issues or having a long name that complicates the illustration.

Id	*Concept*	*Defining Characteristics*
8	Red pedal bicycle	I is a I =Pedal bicycle I is a I =Pedal powered transport device Used as = Transport Moves on = 2 wheels Power = Pedals Color = Red
9	Steel pedal bicycle	I is a I =Pedal bicycle I is a I =Steel transport device Used as = Transport Moves on = 2 wheels Power = Pedals Made of = Steel
10	Red steel pedal bike	I is a I = **Red pedal bicycle** I is a I = **Steel pedal bicycle** Used as = Transport Moves on = 2 wheels Power = Pedals Made of = Steel Color = Red

Figure 28 illustrates a range of *expressions* based on each of the *concepts* defined in might be used to represent the *concept* "red steel pedal bike".

Expression K is a *precoordinated expression* using the *concept* "10 I red steel bikeI ". Each of the other forms is *postcoordinated* by adding refinements that build on the *concept* definitions shown in *Table 14*.

These *expressions* would all be equivalent if the definitions were complete and accurate. In that case, it would possible to transform between them without losing information by appropriately adjusting the associated refinements to take account of the *concept* definitions. In practice the *concept* "bicycle" is marked as *primitive* which places a limit on transformation process.

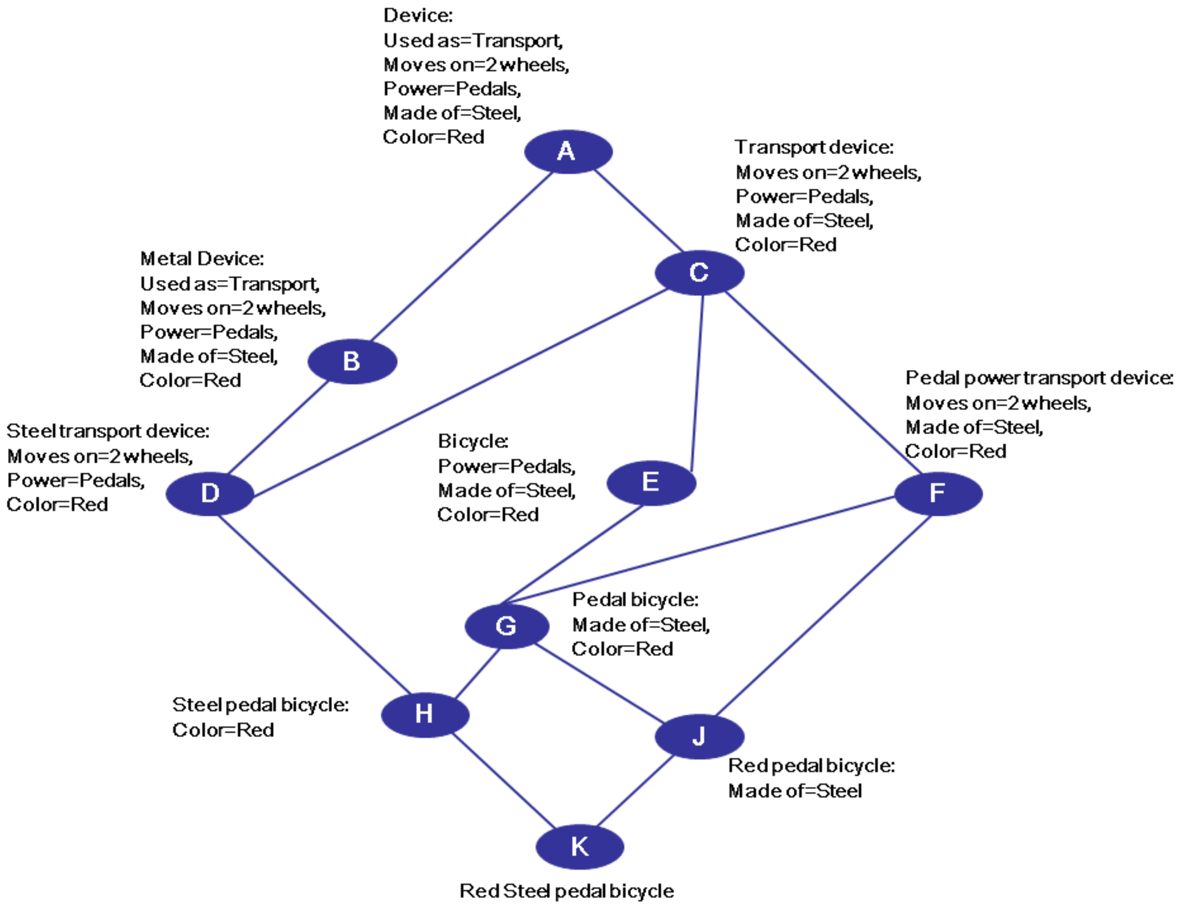


Figure 28: Alternative expressions that mean "red steel pedal bicycle"

Two rules limit the range of equivalent *expressions*:

Rule 1: It is not possible to transform a *primitive concept* into a *postcoordinated expression*.

- A *primitive concept* has a facet that is not represented by its *defining characteristics* and therefore any attempt to represent it in a *postcoordinated* form results in a loss of information.

This is illustrated by consideration of the definitions of the *concept* "bicycle" in *Table 14*. The definition stated in the table is as follows:

| is a | " = "Transport device", | Used as | = | Transport |, | Moves on | = | 2 wheels |, | Origin | = | Man made |

This definition would also apply to a horse-drawn cart or a trailer. Therefore the *concept* "bicycle" must be regarded as *primitive*. Recognizing this fact means that some of the apparently equivalent *expressions* in *Figure 28* cannot be computed as equivalent. Unless the *focus concept* is a *subtype* of "bicycle" it is not possible to compute that it is a kind of bicycle. This means that to create an equivalent *expression* it would be necessary to add | is a | = "Bicycle". This is shown in *Figure 29*.

Examining these definitions, it is apparent that the characteristics shown in gray are redundant because they are part of the definition of "bicycle."

As a consequence of this rule, *primitive concepts* create the limits on the ability to transform an *expression* to a more post-coodinated form. An *expression* can be normalized until all the *concepts* referred to by the *expression* are *primitive*. An *expression* in which all the referenced *concept* are *primitive* is referred to as the *normal form*.

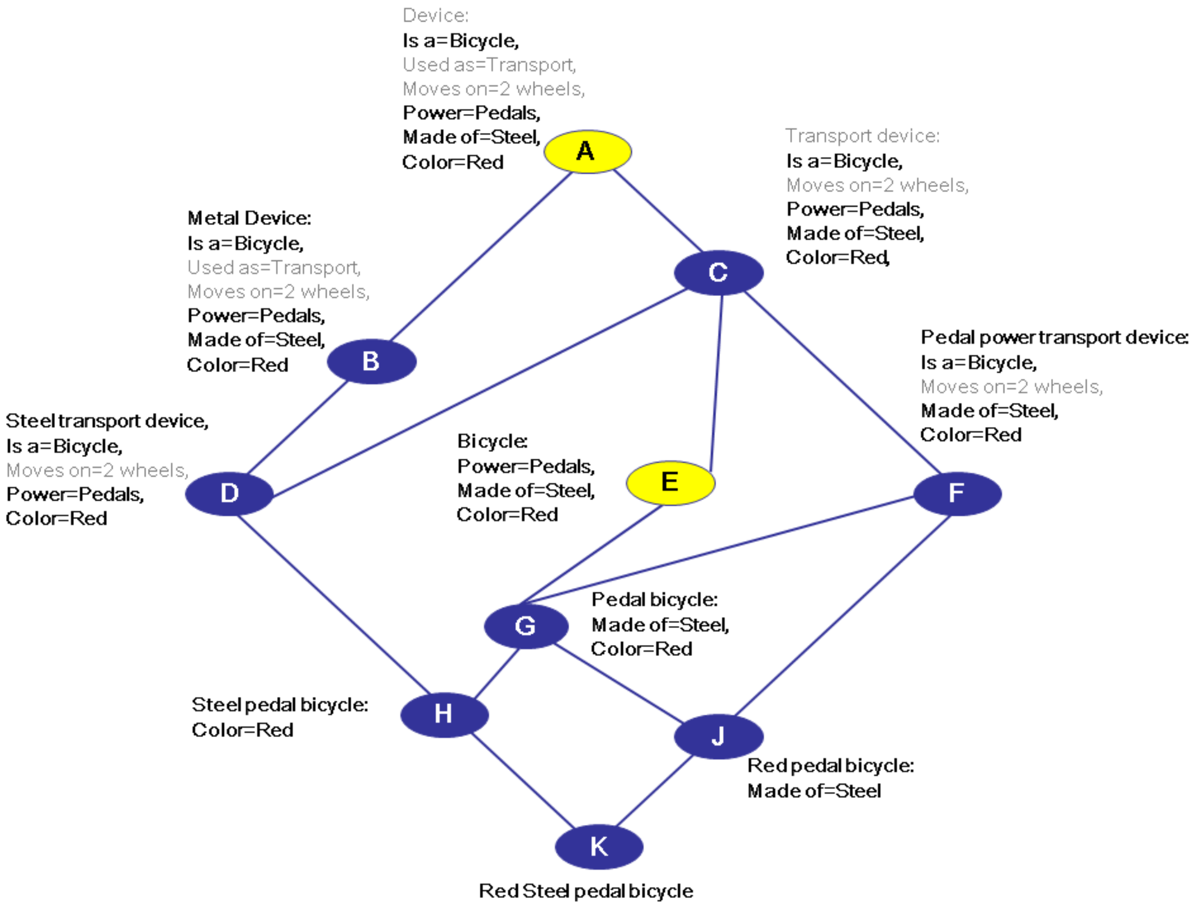


Figure 29: Expressions meaning "red steel pedal bicycle" with "bicycle"recognized as primitive

Rule 2: It is not possible to transform a *postcoordinated expression* into a fully *precoordinated concept* unless such a *concept* already exists in the released terminology.

This second rule is perhaps self-evident but it is stated because, like the first rule, it alters the available representations. If the *concept* "red steel pedal bicycle" was not available in a *precoordinated* form, there are two distinct *expressions* that are as *precoordinated* as possible (i.e. "steel pedal bicycle" + "color " = "red" and "red pedal bicycle" + | made of | = | steel |). This is illustrated in *Figure 30*. In such cases there is no obvious reason to prefer one compared to the other.

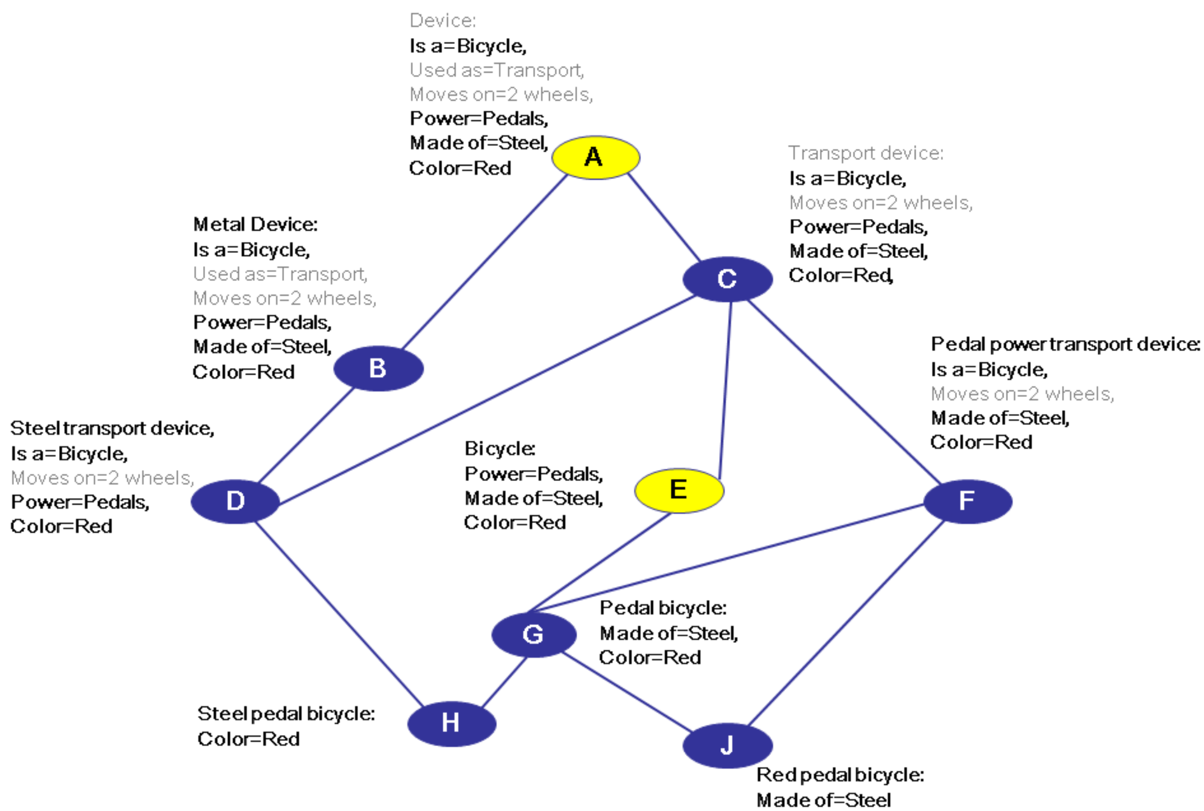


Figure 30: Possible expressions of "red steel pedal bicycle" without precoordinated option

4.3 Representational Forms

This section describes different ways in which *SNOMED CT components*, *derivatives* and *expression* can be represented. These representations include the files in which *SNOMED CT* is distributed as well as possible representations that may be used assist implementation or optimize particular functions.

4.3.1 Release Files

SNOMED CT is provided to licensees as a set of *release files*. The file naming conventions and the structure of these files is described in *Release File Specifications (5)* in a separate section of this guide. There are currently two distinct *Release Formats*:

- *Release Format 1* (*RF1*): The specification in which *SNOMED CT* has been provided since its first release in 2002 (with a few minor amendments).
- *Release Format 2* (*RF2*): Based on a draft trial specification that adds a range of significant enhancements.

4.3.2 Representing SNOMED CT identifiers

Components within *SNOMED Clinical Terms* are identified and referenced using numeric *Identifiers*. These *Identifiers* have the data type *SCTID* (*SNOMED CT Identifier*).

The *SCTID* data type is 64-bit *integer* which is allocated and represented in accordance with a set of rules. These rules enable each *Identifier* to refer unambiguously to a unique component. They also support separate partitions for allocation of *Identifiers* for particular types of component. In the case of *components* that originate in an *Extension*, the *Identifier* also supports separate *namespaces* that distinguish between different issuing organizations .

4.3.2.1 SCTID Data Type

The *SCTID* data type is a 64-bit positive *integer*.

When rendered as a string an *SCTID* must always be represented using decimal digits and when rendered as a string has a maximum permitted length of 18 digits and a minimum length of 6 digits.

Note: Leading zeros are always omitted from the string rendering of an *SCTID*. For example the value "101291009" must <u>not</u> be rendered as "0101291009".

4.3.2.2 SCTID Representation

Each *SCTID* identifies a *SNOMED CT component*. The *Identifier* itself does not contain information related to the meaning of a *concept* or *description*. This means it is not possible to infer anything about the meaning of a *concept* from the numeric value of the *Identifier* or from the sequence of digits in that form of the identifer. The meaning of a *concept* can be determined from *relationships* to other *concepts* and from associated *descriptions* that include human readable terms.

The *SCTID* does however have a structure which includes valuable information about the nature and source of the identified component and the validity of the *Identifier*. This structure supports the following features:

- *Check-digit* validation of the *Identifier*.
  - The *check-digit* is the final digit in the decimal rendering of the *Identifier*. This can be checked to minimize errors from transcription or incomplete copy-paste actions.
- Partitioning between *Identifiers* for different types of *SNOMED CT component*.
  - A two-digit *partition-identifier* distinguishes the *Identifiers* of different component types and prevents the same *Identifier* from being allocated to both a *concept* and a *description*. As a result, when an *SCTID* is read from a record or other resource, it is possible to determine whether it represents a *concept*, a *relationship* or a *description*, before searching for the identified component.
- Namespaces to separate component *Identifiers* originated by different organizations.
  - Organizations are only permitted to issue *Identifiers* which fall within a specified namespace of potential *Identifier* values. This prevents collisions between *Identifiers* issues by different organizations which would otherwise result in ambiguity and errors when sharing data.
  - There are two formats used for representing namespaces.
    - Short format in which *partition-identifiers* are reserved for an organization which is permitted to issue any valid *Identifiers* within the allocated partitions. The short format approach does not require a specific *namespace-identifier* and is only applicable to components originated and maintained by the *IHTSDO* as part of the *International Release* of *SNOMED CT*.
    - Long format in which the *partition-identifier* value indicates that a separate *namespace-identifier* is required to distinguish between components originated as part of an *Extensions* created by an appropriately authorized organization.

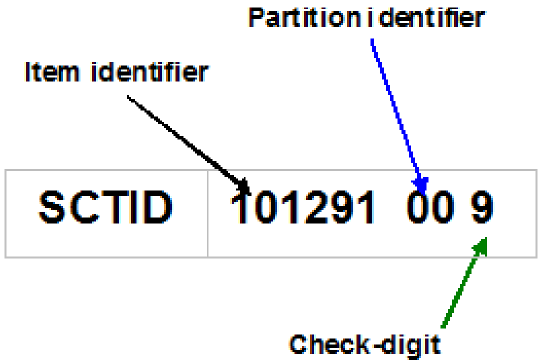


Figure 31: SCTID Short Format - Applicable to components originating from the International Release

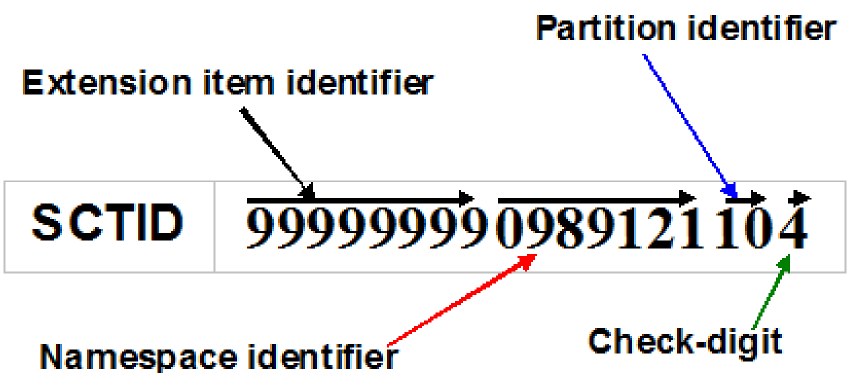


Figure 32: SCTID Long Format - Applicable to components originating from a SNOMED CT Extension

Note: The *IHTSDO* allocates *namespace-identifiers* to organizations such as *IHTSDO Members* and *Affiliates* to enable them to create content and or derivatives in an *Extension*. The *namespace-identifiers* enables unique *SCTIDs* to be issued by many organizations and allow each *SCTID* to be traced to an authorized originating organization .

4.3.2.3 SCTID Constraints

The permissible value for the *SCTIDs* are limited by the following rules:

- Only positive integer values that are greater than 10^5 and less than 10^{18} are permitted.
- The only valid *string* renderings of the *Identifier* value are *strings* of decimal digits (0-9), commencing with a non zero digit.
- The second and third digits from the right hand end of the *string* rendering of the *Identifier* must match one of the *partition-identifier values specified in this guide*.
- The rightmost digit of the *string* rendering is a *check-digit* and must match the value calculated using the specified *check-digit computation*.

Notes:

1. As a result of these rules, many 64-bit integers are not valid *SCTIDs*. The value limitations enable any valid *SCTID* to be stored in either a signed or unsigned 64-bit integer.
2. The rules also ensure that an *SCTID* can be distinguished from code from one of the antecedent code systems *Read Codes* (which are 4 or 5 characters in length) and legacy *Identifiers* from *SNOMED RT* and it predecessors (which always start with a letter).
3. *SNOMED RT* identifiers are *SCTIDs* identical to those used in *SNOMED CT* but in some cases will now refer to *inactive concepts*. In these cases, data in the 900000000000489007 | Concept inactivation indicator reference set | and 900000000000522004 | Historical association | *Reference sets* can be used to find the identifier of the closest equivalent *active concept*.

4.3.2.4 Check-digit

The final (units) digit of the *SCTID* is the *check-digit*. It is not envisaged that users will be routinely required to type *SCTID* values. However, the objective of the *check-digit* is to detect the commonest types of error that may occur due to typographical errors on those occasions where transcription or communication mechanisms may introduce error. Examples may include high-level development such as creating or modifying protocols or pre-specified queries.

An *SCTID* is checked by using the "Verhoeff check", which is a Dihedral D 5 Check. This detects a higher proportion of common typographical errors than either the IBM or Modulus 11 check. Unlike the Modulus 11 check it is effective on decimal strings longer than ten-digits. Furthermore its value can always be represented as a decimal digit without excluding any values.

See *Check-digit computation* for detailed information about the Verhoeff *check-digit* and sample program code.

4.3.2.5 Partition identifier

The penultimate two-digits of the *SCTID* (second and third from the right), are the *partition-identifier*.

The *partition-identifier* indicates the nature of the component identified. This allows the *Identifier* of a *Description* to be distinguished from the *Identifier* of a *Concept*.

The *partition-identifier* also indicates whether the *SCTID* contains a *namespace-identifier* (*long format*) or follows the *short format* applicable to *Identifiers* of *components* that originated in the *International Release*.

Identifiers of *components* that originated in the *International Release* of *SNOMED CT* have one of the following *partition-identifier* values:

Table 15: partition identifier Values for Short Format SCTIDs

PartitionId	*Description*
00	A *Concept*
01	A *Description*
02	A *Relationship*

Identifiers of *components* that originated in an *Extension* have one of the following *partition identifier* values:

Table 16: partition identifier Values for Long Format SCTIDs

PartitionId	*Description*
10	A *Concept*
11	A *Description*
12	A *Relationship*

All other *partition-identifier* values are reserved for future use.

4.3.2.6 Namespace-Identifier

If the *partition-identifier* indicates a long format *SCTID*, the seven-digits immediately to the left of the partition-digit are a *namespace-identifier*. The *namespace-identifier* is an integer value, left padded with '0's as necessary to ensure there are always seven digits in the value. The *namespace-identifier* does not hold meaning.

Each organization that is authorized to generate *SCTIDs* is allocated a *namespace-identifier* by the *IHTSDO*. Each allocated namespace is represented in the *Namespace Concept* metadata sub-hierarchy, released as part of the *International release* (see details in *The Namespace hierarchy*).

4.3.2.7 Item-identifier digits

The string of digits to the left of the *partition-identifier* (in a *short format SCTID*) or to the left of the *namespace-identifier* (in a *long format SCTID*) is referred to as the *item-identifier*. These values are available to uniquely identify an individual entity within the specified partition or namespace. The same *item-identifier* can be allocated in each partition of each namespace as the *SCTID* is rendered unique by the *partition-identifier* and the *namespace-identifier*.

For components in the *International Release* of *SNOMED CT*, *item-identifiers* will usually be issued in the arbitrary order in which components are added to *SNOMED Clinical Terms*. However, due to management of the editing process the sequence of issued *item-identifiers* may be discontinuous.

Caution: In all cases, the value of *item-identifier* on it is meaningless. The only way to determine the meaning of an *SCTID* is by looking up the complete value in an appropriate distribution file.

4.3.2.8 Example SNOMED CT identifiers

The following examples conform to the *SNOMED CT identifier* specification and illustrate a range of possible *Identifiers* within different partitions and namespaces.

SctId	*Partition identifier*	*Check digit*	Notes
100005	00 =*Concept*, using short format	5	The Item *Identifier* digits '100' are the lowest permitted value. Therefore this is the lowest SctId that can be allocated to a *Concept*.
100014	01=*Description*, using short format	4	This is the lowest SctId that can be allocated to a *Description*.

SctId	*Partition identifier*	*Check digit*	Notes
100022	02=*Relationship*, using short format	2	This is the lowest SctId that can be allocated to a *Relationship*.
1290023401004	00=*Concept*, using short format	9	A valid SctId for a *Concept*.
1290023401015	01=*Description*, using short format	5	A valid SctId for a *Description*.
9940000001029	02=*Relationship*, using short format	9	A valid SctId for a *Relationship*.
10000001105	10=*Concept*, using long format	5	A valid long format SctId for a *Concept* in the 0000001 namespace.
10989121108	10= *Concept*, using long format	8	A valid long format SctId for a *Concept* in the 0989121 namespace.
1290989121103	10= *Concept*, using long format	3	A valid long format SctId for a *Concept* in the 0989121 namespace.
1290000001117	11= *Description*, using long format	7	A valid long format SctId for a *Description* in the 0000001 namespace.
9940000001126	12= *Relationship*, using long format	6	A valid long format SctId for a *Relationship* in the 0000001 namespace.
999999990989121104	10= *Concept*, using long format	4	The maximum valid SctId for a *Concept* in the 0989121 namespace.

The Namespace hierarchy

SNOMED CT core release files include metadata *Concepts* that represent each of the allocated *namespace-identifiers*. The *Concepts* representing the namespaces are arranged in a single parent hierarchy, as follows:

- 370136006 | Namespace concept |
  - 373872000 | Core Namespace |
    - | Extension Namespace A {} |
    - | Extension Namespace B {} |
      - | Extension Namespace D {} |

- I Extension Namespace E {} I
- I Extension Namespace C {} I

Figure 33: Hierarchy for: Namespace concept (namespace concept)

In the above hierarchy,I Extension Namespace A {} I, I Extension Namespace B {} I and I Extension Namespace C {} I are all child namespaces of the 373872000 I Core Namespace I (representing the *International edition* which does not have a *namespace-identifier*, and uses short format *SCTIDs* to identify *components*). Also, I Extension Namespace B {} I is the parent namespace of I Extension Namespace D {} I and I Extension Namespace E {} I .

Each *Namespace concept* may only have one parent *Namespace concept* in the 370136006 I Namespace concept I sub-hierarchy.

The namespace hierarchy is used to constrain which content can be promoted from one *Extension* to another without amending the *SCTID*. Content may be moved (without amendment of *SCTID*) from an *Extension* released by the owner of a child namespace to an *Extension* released by the owner of a parent (or ancestor) namespace, as described by the I370136006 I Namespace concept I sub-hierarchy.

Examples:

1. A *concept* with an *SCTID* that includes I Extension Namespace D {} I may be moved to the *Extension* maintained by the owner of I Extension Namespace B {} I without changing its *SCTID*, because this is a parent of the originating namespace.
2. A *concept* with an *SCTID* that includes I Extension Namespace D {} I must not be moved to the *Extension* maintained by the owner of I Extension Namespace C {} I because this is not parent (or ancestor) of the originating namespace. Therefore, to make this move the original *concept* must be inactivated and replaced by a new component with a new *SCTID* in target namespace.
3. Any *concept* may be moved from any *Extension* to the *International Release* (subject only to formal acceptance that is a valid addition for international use).

Namespace concepts have the following characteristics:

- They are *subtypes* (either children or *descendants*) of 370136006 I Namespace concept I.
- The *Fully Specified Name* of each *Concept* has the form " *Extension* Namespace {nnnnnnn} (*namespace concept*)" – where nnnnnnn is the seven digit Namespace-*Identifier*.
- A *Synonym* associated with each *Concept* has the form " *Extension* Namespace nnnnnnn"
- Where appropriate further *Synonyms* may be included to identify the nature of the responsible organization.

When requesting a *namespace-identifier* from *IHTSDO*, there will be a facility to optionally specify a parent *Namespace-identifier* for the new namespace.

To specify a parent namespace for an existing *namespace-identifier*, please contact *info@ihtsdo.org* with details of your existing *namespace-identifier* and its proposed parent *namespace-identifier*.

Caution: Once a *namespace-identifier* has be allocated a parent *namespace-identifier* in this hierarchy, further changes to this hierarchical *Relationship* are not permitted. This restriction is imposed to avoid changes that would undermine traceability of moves between namespaces.

4.3.3 Representing Extensions

4.3.3.1 Extension Tables - Structure

Extensions use the same table structure as the *Concepts*, *Descriptions*, *Relationships*, and *Reference Sets* tables defined in those respective sections of this manual. These tables have the same structure or schema as the *core tables* but are in separate files.

When packaged, *extension* file names should follow the conventions defined by the *IHTSDO*. For more information, refer to the document *SNOMED CT File Naming Convention*.

4.3.3.2 Specification for Namespace within the SCTID

The *identifiers* assigned to all *components* that originated as part of an *extension* include a *namespace-identifier* (see *Representing SNOMED CT Identifiers*). This means that the sets of *Identifiers* available to each organization authorized to issue components are distinct, which ensures that the same *Identifier* cannot be issued by two different organizations.

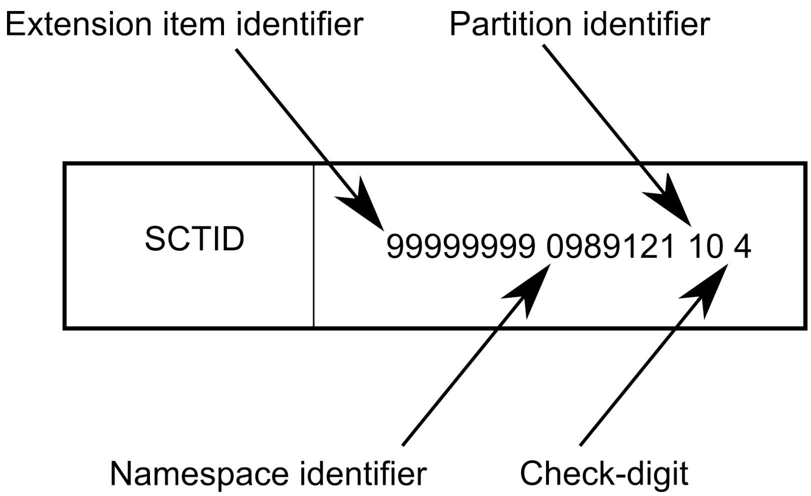


Figure 34: SCTID for a concept originated as part of an Extension

All *Extension components* (rows) originated by an organization must use the *Namespace Identifier* assigned to that organization . *Namespace Identifiers* are issued by the *IHTSDO* so that the *Namespaces* remain unique between organizations . Allocated *Namespaces* are recorded as *Concepts* in the | SNOMED CT Model Component |*hierarchy* ("special *concepts*" hierarchy in *RF1*) when they are issued to an organization .

Namespaces serve three *Roles*:

- Preventing collision or reuse of *SNOMED CT identifiers*;
- Indicating the origin of a component. In RF2 responsibility for maintenance is tracked only by the ModuleId, this is the field that should be used to avoid potential risk of two organizations make conflicting changes to the same *component*.
- Indicating the source for information about a *concept* - relevant for *Extension Concepts* that are not directly available in a particular system.

All *Extension components* (rows) should use the appropriate *partition-identifiers* for *Extensions*. This ensures that *components* of the *SNOMED CT International Release* can be distinguished from *components* that are part of an *Extension*.

Note: *Components* that originate as part of the *International Release* do not have a *namespace* field and are distinguished instead by *partition-identifier* values that are specific to *International Release*.

Note: Each organization can assign the item-*Identifier* portion of the *SCTID* in any way within its *Namespace*. If there is a need to allocate part of the development process to a subdivision within an organization , they may be allocated a set or range of *item-identifiers* that have not yet be used or allocated within that *Namespace*. The authorized organization must ensure that it tracks and manages all such allocations in a way that avoids any risk of reuse of the same *SCTID*.

4.3.3.3 Namespace allocation

*Namespace-identifier*s are allocated by the *IHTSDO* to licensed organizations . The *IHTSDO* is under no obligation to allocate a *namespace* to any organization and makes these allocations at its discretion.

Allocation of a *namespace* does not imply any endorsement of the reputation of an organization nor to the quality or fitness for purpose of any *Extensions* created by that organization . Users and/or vendors incorporating *Extensions* into their application do so at their own risk and should satisfy themselves with the reputation of the responsible organization and the quality the *Extensions* so incorporated.

4.3.3.3.1 Namespace Allocation Policy/Regulation

Title: *Namespace* Allocation Policy/Regulation

Effective Date: August 4, 2009 **Owner** : Management Board

Date Last Reviewed: October 14, 2009 **Date Last Revised** : October 14, 2009

4.3.3.3.1.1 Regulation Statement

IHTSDO will allocate *SNOMED CT Namespace Identifiers* upon written request from a Member or an Affiliate in accordance with the procedures outlined below. The *IHTSDO* will also maintain and publish a register of *Namespace Identifiers* issued.

Section 9 of the Articles of Association provides the starting point for the *Namespace* Allocation Policy. It states that:

- 9.1 Only the Association may issue *Namespace Identifiers*.
- 9.2 The Association shall, upon written request from a Member or an Affiliate in accordance with such procedures as the Association may prescribe by Regulations, issue one or more *Namespace Identifiers* to the Member or Affiliate. The Association shall not unreasonably refuse to issue a *Namespace Identifier* to a Member or an Affiliate.
- 9.3 The Association shall be responsible for ensuring that each *Namespace Identifier* is only issued to a single Member or Affiliate.

In addition, section 7.1.7 states that "An Affiliate may not create any Standards-Based Third Party *Extension* or any Standards-Based *Derivative* from the Member's National *Extensions* unless that Affiliate has been issued with a *Namespace Identifier*."

4.3.3.3.1.2 Definitions

Affiliate: An Affiliate of *IHTSDO* in accordance with *IHTSDO*'s *Articles of Association*, i.e. a person or organization to which the *International Release* of *SNOMED CT* (whether on its own or as part of a Member's *National Release* of *SNOMED CT*) is distributed or otherwise made available under the *Articles of Association.*

Namespace Identifier: A code or that part of a code that identifies the organization responsible for creating and maintaining a standards-based *extension* or a standards-based *derivative.* It is an element of *SNOMED CT concept Identifiers*.

4.3.3.3.1.3 Context

Namespace Identifiers are 7-digit numbers that *IHTSDO* issues to those who create *extensions* to *SNOMED CT*, such as national *extensions*. *Namespace identifiers* ensure that it is clear who developed and maintains particular customized terminology. They also ensure that terminology in *SNOMED CT extensions* has unique *Identifiers* but a common structure, which facilitates application development and the creation of *Reference Sets*. There is a defined process for management of *Namespace Identifiers* when terminology is moved between *extensions* or from an *extension* into the *International Release*.

It should be noted that this policy covers the technical mechanism to allocate *Namespace Identifiers* in order to be able to identify the source of content, prevent collisions in terminology that would affect interoperability, and achieve similar goals. It does not cover whether or not particular types of content, including *extensions* and *derivatives*, can be used in a given context. This may be the subject of national policies, guidelines, or other documents. Requesters of a *Namespace Identifier* are encouraged to review *SNOMED CT International*

Release documentation and to consult with *IHTSDO Members* in countries in which deployment of any content developed in the *Namespace* is planned for additional guidance, policy, and/or process documents which may be relevant.

4.3.3.3.1.4 Procedures

4.3.3.3.1.4.1 Informing the Community of Practice :

IHTSDO will inform the Community of Practice about the process for requesting a *Namespace Identifier*.

- A copy of this regulation will be posted to the *IHTSDO* website.
- Instructions for requesting a *Namespace Identifier*, including the form for making such a request, will be posted to an appropriate location on the *IHTSDO* website (e.g. the Frequently Asked Question or "How do I?" pages).
- Confirm which Members would like to be notified when one of their Affiliates requests a *Namespace*, i.e. an Affiliate listing an address in the jurisdiction in question on their *Namespace Identifier* Application Form and/or who identified that they received their Affiliate License through that jurisdiction.

4.3.3.3.1.4.2 Requesting & Granting a Namespace:

The Association shall, upon written request from a Member or an Affiliate in accordance with such procedures as the Association may prescribe by Regulations, issue one or more *Namespace Identifiers* to the Member or Affiliate. The Association shall not unreasonably refuse to issue a *Namespace Identifier* to a Member or an Affiliate.

- To request a *Namespace Identifier*, individuals/ organizations should complete and submit a copy of the *Namespace Identifier* Application Form by email to support@ihtsdo.org.
- *IHTSDO* should verify that the requester is either:
  - An *IHTSDO Member*
  - An individual or organization who holds a valid AffiliateLicense
  - An individual or organization who does not fall into the above categories but whose application is approved in writing by the *IHTSDO*
- If the conditions above apply, *IHTSDO* should issue a unique *Namespace Identifier* to the requester if:
  - The request is from an *IHTSDO Member*;
  - The request is from the holder of a valid Affiliate License , is for a single *Namespace Identifier*, and the requester has not already been issued a *Namespace Identifier*; OR;
  - *IHTSDO*'s CEO approves the request in writing.
- The issuance of the *Namespace Identifier* should be confirmed in writing with the requester, along with a link to this policy and a reminder that they will be contacted annually to reconfirm their contact information and potentially provide additional information to be published in the register. Requesters should also be provided with Member contact information and the recommendation that they should contact relevant Members to obtain any additional national guidance, policy, and process documents which may be relevant.
- The relevant Member should be informed if they have requested to be notified when an Affiliate from their jurisdiction requests a *Namespace*, i.e. an Affiliate listing an address in the jurisdiction in question on their *Namespace Identifier* Application Form and/or who identified that they received their Affiliate License through that jurisdiction.

4.3.3.3.1.4.3 Format of Namespace Identifiers:

- *Namespace Identifiers* are 7 digit numeric codes;
- *Namespace Identifiers* are issued in sequence, unique, and not re-used.

4.3.3.3.1.4.4 Maintaining and Publishing a Namespace Register:

IHTSDO will maintain and publish an up-to-date *Namespace* Register.

- When *Namespace Identifiers* are allocated, a record of the number of that *Identifier*, the date of issuance, the body from which the Affiliate License was obtained (if applicable), and the name and contact information of the individual/organization to which it was issued will be added to the *Namespace* Register.
- On an annual basis, *IHTSDO* will contact all those to whom *Namespace Identifiers* have been issued by email to confirm contact information. A reminder will be sent after 30 days if a response has not been received. The year in which a confirmation of current contact information was last received will also appear in the *Namespace* Register.
- *IHTSDO* reserves the right to make a *Namespace Identifier inactive* for current and future use (i.e. it cannot be used for newly-created *concepts* from that point onward) if the individual or organization to which it was issued cannot be contacted after three attempts. This *status* will be noted in the *Namespace* Register and the individual or organization to which the *Namespace Identifier* was issued will be notified accordingly.
- *IHTSDO* also reserves the right to make a *Namespace Identifier inactive* if (1) it is requested to do so by the organization to which the *Namespace Identifier* was issued, (2) the organization to which the *Namespace Identifier* was issued is involved in a merger or acquisition with another organization to which a *Namespace Identifier* has been issued, or (3) it receives a written complaint about the use of that *Namespace* that, upon investigation, it determines to be well-founded, according to the protocol for material breaches and termination of Affiliate Licenses identified in clause 5.2 of the Affiliate License .
- The *Namespace* Register will be published with each version of the *SNOMED CT International Release*. In the future, *IHTSDO* reserves the right to also publish the NamespaceRegister on the *IHTSDO* website.

4.3.3.3.1.5 References

- *IHTSDO Articles of Association* ;
- *SNOMED CT Technical Implementation Guide* ;
- *IHTSDO Namespace Identifier Application Form* .

4.3.3.3.1.6 Document Control

This policy was approved by the *IHTSDO* Management Board on August 4, 2009 and is subject to regular review according to *IHTSDO*'s policy review processes. Key stakeholders include the Technical Committee, the Implementation and Innovation Committee, the Member Forum, and the Affiliate Forum.

4.3.3.4 Component Guidelines

Descriptions that are part of an *Extension* can refer to either a *Concept* that is part of that *Extension*, a *Concept* that is part of another *Extension*, or an *International Release Concept*.

Relationships that are part of an *Extension* can relate two *Concepts* in the *Extension* or two *Concepts* in different *Extensions*. The *relationship* can also relate the *extension Concept* to an *International Release Concept*-- that is, *sourceId* is in the *extension* and *destinationId* is in the *International Release*.

4.3.3.5 Content of Extensions

The *components* in an *Extension* have the same structure as *International Release components* of *SNOMED CT*.

SNOMED CT International Release is not dependent on availability of any *Extension*. However, all *SNOMED CT Extensions* are dependent on the *SNOMED CT International Release*. Some *Extensions* may also be dependent on other *Extensions*. Dependencies between *Extensions* must be declared and must not be circular.

The following rules apply to dependencies between *components* and *derivatives* in *Extensions*.

Note: In these rules,

- *Containing-Extension*, refers to the *Extension* that contains the named *component* or *derivative*.
- *Dependee-Extension*, refers to another *Extension* on which the *Containing-Extension* is dependent.

1. Every *Concept* in an *Extension*:

- Must be a *subtype descendant* of an *International Release Concept*:
    - This descent may be indirect, passing through *Concepts* in either the *Containing-Extension* or a *Dependee-Extension*.

2. Every *Description* in an *Extension*:

- Must apply to a *Concept*, in the one of the following *Namespaces*: the *Containing-Extension*, the *International Release* or a *Dependee-Extension*.

3. Every defining *Relationship* in an *Extension*:

- Must define a *sourceId* which refers to a *Concept* in the *Containing-Extension*.
    - In exceptional circumstances, an *Extension* may add additional defining attributes to a *Concept* in the *International Release* or a *Dependee-Extension*.
- Must have *typeId* which refers to a *Concept*, in the one of the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.
- Must have a *destinationId* which refers to a *Concept*, in the one of the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.

4. A *Reference Set* in an *Extension*:

- May include references to *components* and *derivatives* in any of the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.

5. The enumerated values used in *RF2 components* that form part of an *Extension* must be all be represented by metadata *Concepts* that are present in one of the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.

The following additional rules are only relevant to *Extensions* represented using *Release Format 1*:

- *Reference Sets* in an *Extension*:
    - May include as its members *Components* from the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.
    - May refer to other *Reference Sets* in the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.
    - May provide maps for *Concepts* from the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.

4.3.3.6 Transfer of Responsibility between Organizations

When the need arises to transfer *components* (*Concepts*, *Descriptions*, *Relationships*) from the *International Release* content to an *Extension*, from an *Extension* to the *International Release*, or from one *Extension* to another, conversation and coordination between the sending and receiving organizations is needed. In some cases, entire tables may be transferred - not just individual components.

It should be noted that the transfer of *components* among *Extensions*, or between an *Extension* and the *International Release*, is subject to the terms of the *IHTSDO Affiliate* License and, within an *IHTSDO Member* country, may also be subject to the *terms* of that Member's *SNOMED CT* national license .

Examples of transfers include:

- From the *SNOMED CT International Release* to an organization responsible for an *Extension*:
    - This occurs if a decision is made that some *Concepts* in the *International Release* are specific to a *Realm* or domain or interest for which another organization has been allocated responsibility:
        - For example, this applies to UK specific drugs and UK specific administrative *Concepts* which are maintained by the *UK NHS*.

- From an organization responsible for an *Extension* to the *International Release*:
  - This occurs if an organization recognizes that some of its *Extension* content belongs in the *International Release* as it has general applicability;
  - It also occurs if an organization hands over responsibility for its entire *Extension* to the *IHTSDO*.
- From one organization responsible for an *Extension* to another organization :
  - This occurs if one organization recognizes that some of its *Extension* content belongs in a domain managed by another organization ;
  - It also occurs if an organization hands over responsibility for its entire *Extension* to another organization .

There are three types of transfer of responsibility:

- Transfer of an entire *Extension* (i.e. all *components* ever issued with an *SCTID* in a given *Namespace*. from one organization to another organization):
  - This is a straight forward process. All that happens is that another organization assumes responsibility for the original *Namespace-identifier*. There is no need for detailed tracking of individual *components*.
- Transfer of one or more *components* from an *Extension* to the *International Release* or to a "parent"*Extension*:
  - As a result of *revised to guidance on SNOMED CT Identifier Updates*, since 2011, some transfers can be made without changing the *Identifier* of the *component* provided that the RF2 format *moduleId* field is used to denote that the *component* is now being issued as part of a different module.
- Transfer of one or more *components* between other *Extensions* or from the *International Release* to any *Extension*:
  - In this case, the *Namespace* is not transferred and thus, to fulfill the roles of the *Namespace-Identifier*, the *component* must be assigned new *SCTIDs* in the *Namespace* of the newly responsible organization :
    - The previous instances of these *components* are withdrawn from *current* use with the *Status* value *Moved Elsewhere*;
    - Appropriate *Relationships* point to replacement *components* in the new *Namespace*.

The transfer of responsibility depends on the release schedules of the organizations involved. Often the original source organization will be aware of an intended move before the target organization has accepted responsibility and released the *component*. To facilitate this, an interim *Status* value *Pending Move* is applied to *components* that are being moved to another *Namespace* but are intended for *active* use until their replacements are found in the target *Namespace*.

To provide continuity for a *Concept* if responsibility is transferred, *Concept* versioning and supporting files are coordinated as follows:

Table 17: Processing Transfers between Extensions

	Sending Organization	**Receiving Organization** [4]
Start State	ConceptA • Active = 1	

[4] **Assume assigned namespace = 9999999**

	Sending Organization	**Receiving Organization** [4]
Agreement to transfer responsibility	ConceptA • Active = 1 900000000000489007 | Concept inactivation indicator reference set | • ConceptA - value: 900000000000492006 | Pending move | 900000000000524003 | MOVED TO association reference set | • ConceptA-> *Namespace* 9999999 **Note:** *Namespace* 9999999 is recorded in the *SNOMED CT Concepts* File. Therefore, the Sending Organization can track the organization to which the *concept* has moved, even if the new *Concept Identifier* is not yet assigned.	
Responsibility Transferred	ConceptA • Active = 0 900000000000489007 | Concept inactivation indicator reference set | • ConceptA - value: 900000000000487009 | Moved elsewhere | 900000000000524003 | MOVED TO association reference set | • ConceptA-> *Namespace* 9999999	ConceptB (created as replacement for ConceptA) • Active = 1 900000000000525002 | MOVED FROM association reference set | • ConceptB -> ConceptA **Note:** The Receiving Organization can record the *Concept Identifier* previously used for the *concept*.

4.3.3.7 Released Extensions

The following *extensions* are included in the *International Release* of *SNOMED CT* from the *IHTSDO*. As with any *extension*, their content may not be suitable for use everywhere, and users should consult with their *National Release* Center for information regarding the use of *extension* content within an *IHTSDO Member* country.

[4] **Assume assigned namespace = 9999999**

Table 18: Released Extensions

Extension	Distribution	*Extension* Contents
U.S. Drug *Extension*	*International Release*	Actual manufactured drugs approved for distribution in the United States at the "actual medicinal product" (AMP) level. The AMP is a syntactic *normal form* consisting of: • Name (Proprietary); • Strength; • Dosage Form. All AMPs relate to "virtual medicinal product" (VMP) *concepts* in the *SNOMED CT Core*. All AMPs include the "has *active* ingredient" *relationship* where the *active* ingredient is a substance in the *SNOMED CT Core*.

4.3.4 Representational Forms for Expressions

4.3.4.1 SNOMED CT compositional grammar

The *SNOMED* Composition Grammar is a lightweight syntax for representation of *SNOMED CT expressions*. It is has been proven to be both human readable and machine parsable.

4.3.4.1.1 Background

4.3.4.1.1.1 Prior versions and status of revision

The *SNOMED* Composition Grammar was initial specified as part of the document "*SNOMED Clinical Terms Abstract Logical Models and Representational Forms, External Draft for Comment Version*". This was used extensively and was proven to be both human readable and machine parsable.

The current specification which has now been adopted as an *IHTSDO* Standard, follows the prior version in most details. It includes the following enhancements:

1. The syntax of the grammar specification is now Augmented Backus-Naur Form (ABNF) [5] which provides a formal standards-based reference for the grammar's structure.
2. Unnecessary whitespace designators, <ws>, were removed from several places in the grammar.
3. The maximum length *constraint* for *SNOMED Clinical Terms Identifiers* (*SCTIDs*) is added to this grammar. *SCTIDs* consist of sequences of digits, from a minimum of 6 to a maximum of 18 digits in length.
4. The hex code for carriage return (CR) was incorrectly given as '0C' in the previous version. It is corrected to '0D' in this version.
5. Detailed character encoding information for *UTF-8* is added.
6. The definition of *term* has been amended to allow correct parsing by the APG parser generator.

4.3.4.1.1.2 Compositional Grammar and the HL7 Code data type

The *SNOMED CT compositional grammar* allows *SNOMED CT expressions* to be represented as a text string that can be carried in *HL7* version 3 messages, in the *Code* data type. In particular, the grammar is intended to replace the *qualifier* mechanism that formerly was in the *HL7 Concept Descriptor* data type (CD data type), and which was removed in the *HL7* version 3 data types *Release* 2.

In May 2008, the *HL7* Version 3 Standard "Data Types - Abstract Specification, *Release* 2" was released for Normative Ballot 2.

[5] ABNF as defined by Internet Standard 68, RFC 5234

This revised standard defined what can be carried in the *Code* data type as "the plain code symbol defined by the code system, or an *expression* in a syntax defined by the code system which describes the *concept*."

The following details are quoted from the *HL7* Version 3 Standard: Data Types - Abstract Specification, *Release* 2, Normative Ballot 2 - May 2008 (*HL7* V3 DT R2), section 4.5.1 "Code (code): ST.SIMPLE":[6]

Table 19: Code definition from HL7 Data Types Release 2

Code (code) :*ST.SIMPLE* Definition: The plain code symbol defined by the code system, ***or an expression in a syntax defined by the code system which describes the concept.*** (emphasis added) If provided, the code SHALL be an exact match to a plain code symbol or *expression* defined by the code System. If the code system defines a code or *expression* that includes whitespace, the code SHALL include the whitespace. **An *expression* can only be used where the code System either defines an *expression* syntax, or there is a generally accepted syntax for the code System**. (emphasis added) The syntax described herein is intended to satisfy the need for a "syntax defined by the code system" as stated above, when the "code System" is *SNOMED CT*.

[6] http://www.hl7.org/v3ballot/html/infrastructure/datatypes_r2/datatypes_r2.htm

4.3.4.1.2 Compositional grammar: Normative specification

Table 20: ABNF definition of the SNOMED CT compositional grammar

```
expression = concept *("+" concept) [":" ws refinements ]
concept = ws conceptId ws ["|" ws term ws "|" ws]
conceptId = sctId
term = 1*nonwsnonpipe *( 1*SP 1*nonwsnonpipe )
refinements = ( attributeSet *attributeGroup ) / 1*attributeGroup
attributeGroup = "{" attributeSet "}" ws
attributeSet = attribute *("," attribute)
attribute = attributeName "=" attributeValue
attributeName = ws attributeNameId ws ["|" ws term ws "|" ws]
attributeValue = concept / (ws "(" expression ")" ws)
attributeNameId = sctId
sctId = digitNonZero 5*17( digit )
ws = *( SP / HTAB / CR / LF ) ; white space
SP = %x20
HTAB = %x09
CR = %x0D
LF = %x0A
digit = %x30-39
digitNonZero = %x31-39 ; digits 1 through 9, but excluding 0
nonwsnonpipe = %x21-7B / %x7D-7E / UTF8-2 / UTF8-3 / UTF8-4
UTF8-2 = %xC2-DF UTF8-tail
UTF8-3 = %xE0 %xA0-BF UTF8-tail / %xE1-EC 2( UTF8-tail ) /
%xED %x80-9F UTF8-tail / %xEE-EF 2( UTF8-tail )
UTF8-4 = %xF0 %x90-BF 2( UTF8-tail ) / %xF1-F3 3( UTF8-tail ) /
%xF4 %x80-8F 2( UTF8-tail )
UTF8-tail = %x80-BF
```

4.3.4.1.3 Informative comments

Table 21: BNF representation of Compositional Grammar (detail)

```
Expression = concept *( "+" concept) [ ":" ws refinements ]
```

<table>
<tr><td></td><td>An *expression* supports combinations of one or more *concepts* optionally refined by a set of *refinements*. The meaning of the *expression* is a *subtype* of all the *concepts* constrained by the set of *refinements*.

Note that where there is a requirement for multiple separately qualified *concepts* to be present these are expressed in *attribute groups* within a *refinement* of a general *concept* such as "*situation with explicit context*".</td></tr>
<tr><td colspan="2">*concept* = ws conceptId ws ["**|**" ws *term* ws "**|** " ws]</td></tr>
<tr><td></td><td>A *concept* is represented by a *conceptId* optionally followed by a *term* enclosed by a pair of "|" characters.

Whitespace before or after the *conceptId* is ignored as is any whitespace between the initial "|" characters and the first non-whitespace character in the *term* or between the last non-whitespace character and before second "|" character.</td></tr>
<tr><td colspan="2">conceptId = sctId</td></tr>
<tr><td></td><td>The *conceptId* must be a valid *SNOMED CT identifier* for a *concept*. The initial digit may not be zero. The smallest number of digits is six, and the maximum is 18.</td></tr>
<tr><td colspan="2">*term* = 1*nonwsnonpipe *(1*SP 1*nonwsnonpipe)</td></tr>
<tr><td></td><td>The *term* must be the *term* from a *SNOMED CT description* that is associated with the *concept* identified by the preceding *concept identifier*. For example, the *term* could be the preferred *description*, or the preferred *description* associated with a particular translation. The *term* may include valid *UTF-8* characters except for the pipe "|" character [7]. The *term* begins with the first non-whitespace character following the starting "|" character and ends with the last non-whitespace character preceding the next "|" character.</td></tr>
<tr><td colspan="2">*refinements* = (attributeSet *attributeGroup) / 1*attributeGroup</td></tr>
<tr><td></td><td>A *refinement* contains all the grouped and ungrouped attributes that refine the meaning of the containing *expression*. The ungrouped attributes, if any, are all listed first, followed by all the grouped attributes.</td></tr>
<tr><td colspan="2">attributeGroup= " **{** " attributeSet " **}** " ws</td></tr>
<tr><td></td><td>An *attribute group* contains a collection of attributes that operate together as part of the *refinement* of the containing *expression*.</td></tr>
<tr><td colspan="2">attributeSet= attribute *("**,**" attribute)</td></tr>
<tr><td></td><td>An attribute set contains one or more *attribute name* -value pairs expressing *refinements*.They are separated by commas.</td></tr>
</table>

[7] The specification for term should be comparable with the specification for the Concepts.FullySpecifiedName and Descriptions. Term fields in the release table structure (as described in SNOMED Clinical TermsTechnical ReferenceGuide, July 2008, IHTSDO). The non-pipe constraint adds greater stringency to the Compositional Grammar specification.

<table>
<tr><td colspan="2">attribute= attributeName "= " attributeValue</td></tr>
<tr><td></td><td>An attribute name -value pair expressing a single refinement of the containing expression.</td></tr>
<tr><td colspan="2">attributeName= ws attributeNameId ws [" |" ws term ws "| " ws]</td></tr>
<tr><td></td><td>The name (or relationship type) of an attribute to which a value is applied to refine the meaning of a containing expression. The attribute name is represented by an appropriate conceptId optionally followed by a term enclosed by a pair of "|" characters.

Whitespace before or after the conceptId is ignored as is any whitespace between the initial "|" characters and the first non-whitespace character in the term or between the last non-whitespace character and before second "|" character.</td></tr>
<tr><td colspan="2">attributeValue= concept / (ws "(" expression ") " ws)</td></tr>
<tr><td></td><td>A concept or expression representing the value of a named attribute which refines the meaning of a containing expression. If an expression is used this must be enclosed in brackets.</td></tr>
<tr><td colspan="2">attributeNameId = sctId</td></tr>
<tr><td></td><td>The attribute name id must be the conceptId for a concept that is a subtype descendant of the SNOMED CT concept"attribute".</td></tr>
<tr><td colspan="2">sctId = digitNonZero 5*17(digit)</td></tr>
<tr><td></td><td>A n sctId is used for an attribute id or a concept id. The initial digit may not be zero. The smallest number of digits is six, and the maximum is 18.</td></tr>
<tr><td colspan="2">ws= *(SP | HTAB | CR | LF)</td></tr>
<tr><td></td><td>Whitespace characters (space, tab, linefeed and carriage return) are ignored everywhere in the expression except:

1. Whitespace within a conceptId or attributeNameId is an error.
Note: Whitespace before or after the last digit of a valid Identifier is ignored.

2. Whitespace within a term is treated as a significant character of the term.
Note: Whitespace before the first or after the last non-whitespace character of a term is ignored</td></tr>
<tr><td colspan="2">nonwsnonpipe= %x21-7B / %x7D-7E / UTF8-2 / UTF8-3 / UTF8-4</td></tr>
<tr><td></td><td>Non whitespace includes printable ASCII characters (these are also valid UTF8 characters encoded as one octet) and also includes all UTF8 characters encoded as 2- 3- or 4-octet sequences. It excludes space (which is %x20) and the pipe character "|" (which is %x7C), and excludes CR, LF, HTAB and other ASCII control codes. SeeRFC 3629 (UTF-8, a transformation format of ISO 10646 authored by the Network Working Group).</td></tr>
</table>

digitNonZero= %x31-39	
	The first character of a *concept identifier* is constrained to a digit other than zero.
digit= %x30-39	
	Any digit 0 through 9

4.3.4.1.4 Examples of Grammar

The following examples build on each other and in complexity. They are primarily aimed at demonstrating the syntax of the *expression* grammar, although its meaning is also discussed in a number of places:

An *expression* may consist of a single *concept*, followed by a *description* associated with that *concept*. Which particular *description* to use is not mandated, but as a general rule, it may be preferable to use the *preferred term* in any particular *dialect* to achieve some level of consistency. However, such guidance is not strictly in the scope of this guide, and may be given elsewhere.

297186008 | motorcycle accident |

The syntax does not require a *description* to be associated with a particular *concept*, so the following is also a valid *expression*:

297186008

Two or more *concepts* may be combined to form a new *concept* by joining them with the "+" symbol. The resultant *expression* is the *child* of each of the *concepts* in the *expression*. The resultant *expression* below IS AN accident caused by a blizzard and also | is a | motorcycle accident.

217724009 | accident caused by blizzard | +297186008 | motorcycle accident |

Although not stipulated by the syntax, note that two *concepts* joined in this way must be from the same top level *hierarchy*. The syntax does not mandate which *concepts* in the *expression* should have associated *descriptions* and which should not so it is valid, but not advisable, to mix and match. For example, the following syntax is valid:

217724009 +297186008 | motorcycle accident |

The syntax allows spaces, tabs and carriage returns in most places. For example, the following examples have identical meaning to the one above:

217724009 + 297186008 | motorcycle accident |

217724009

+ 297186008

| motorcycle accident |

Using the "+" symbol is symmetrical and equivalent to starting with one of the *concepts* and adding an | is a | *refinement*, with a *value set* to the other *concept*. For example, the following two *expressions* are equivalent to each other and to the preceding *expression*:

217724009 | accident caused by blizzard |:

116680003 | is a | =297186008 | motorcycle accident |

297186008 | motorcycle accident |:

116680003 | is a | =217724009 | accident caused by blizzard |

One or more *refinements* may be added to a *concept* to qualify it. This is done by putting the *concept* to be qualified before a colon and the qualifying *expression* after. The qualifying *expression* is of the form "attribute = value". The example below describes an operation to remove an ovary using a laser.

83152002 | oophorectomy |:

260686004 | method |=257820006| laser excision - action |

Refinements may also be applied to a conjoined *concept*. For example, the following two *expressions* (building on a preceding example) are equivalent:

313056006 | epiphysis of ulna |:

272741003 | laterality | =7771000 | left |

119189000 | ulna part | + 312845000 | epiphysis of upper limb |:

272741003 | laterality | =7771000 | left |

Note that there are no brackets round "119189000 | ulna part | + 312845000|epiphysis of upper limb"in the above example.

Where more than one qualifying *expression* is required, these can be separated using a comma. The example below describes the removal of the right ovary using laser excision.

83152002 | oophorectomy |:

260686004 | method |=257820006| laser excision - action |,

363704007 | procedure site | =20837000 | structure of right ovary |

A further example, below, describes the removal of the left fallopian tube using diathermy excision:

120053002 | Salpingectomy |:

260686004 | method | =261519002 | diathermy excision - action |,

363704007 | procedure site | =113293009 | structure of left fallopian tube |

Where a *SNOMED CT concept comprises* a number of other *concepts* or sub - *expressions*, it may be necessary to group qualifications applied to that *concept* in order to avoid ambiguity as to how they apply. An example of a *SNOMED CT concept* that *comprises* a number of other sub - *expressions* is:

116028008 | salpingo-oophorectomy |

This procedure *comprises* two sub-procedures: the excision of part of all of the ovarian structure; and the excision of part or all of the fallopian tube structure. We should note at this point that there is a subtle difference between a *subsumptive relationship* and a *comprising relationship*:

A motorcycle accident caused by low visibility ***is a*** motorcycle accident AND

is an accident caused by a blizzard.

A salpingo-oophorectomy ***comprises*** a fallopian tube excision and an oophorectomy.

This is demonstrated by the *SNOMED CT normal form* for salpingo-oophorectomy, as shown below:

71388002 | procedure |:

{260686004 | method | =129304002 | excision - action |,

405813007 | procedure site - Direct | =15497006 | ovarian structure |}

{260686004 | method | =129304002 | excision - action |,

405813007 | procedure site - Direct | =31435000 | fallopian tube structure |}

Where it is necessary within a *postcoordinated expression* to unambiguously qualify individual components of a *concept* comprised of a number of other *expressions* (as in the above example), grouping may be used. The following example describes a salpingo-oophorectomy, with laser excision of right ovary and diathermy excision of left fallopian tube. Note that without the grouping, it would not be possible to tell on what structure the laser excision was used and on what structure the diathermy excision was used.

116028008 | salpingo-oophorectomy |:

{260686004 | method |=257820006| laser excision - action |,

363704007 | procedure site | =20837000 | structure of right ovary |}

{260686004 | method | =261519002 | diathermy excision - action |,

363704007 | procedure site | =113293009 | structure of left fallopian tube |}

A number of grouped *qualifiers* may be thus used to refine a *concept*. Note there is no comma between adjacent groups (as there are between adjacent *expressions*). Also note, the syntax does not limit the number of *qualifiers* in a group or the number of groups within an *expression*.

It is also possible to nest *expressions*, one inside the other. Any legal *expression* may be wrapped in a pair of brackets, and included in another *expression* in the same way as a *concept* would be. For example, the following *expression* describes a fracture of the femur caused by a motorcycle accident in a blizzard:

71620000 | fracture of femur |:

42752001 | due to | = (217724009 | accident caused by blizzard | +297186008 | motorcycle accident |)

In the example above, note the use of "( )" brackets, to identify a nested *expression*, as opposed to "{ }" brackets, used elsewhere, to identify groups.

The following examples show how complex *expressions* may be build up from simple ones, a layer at a time. This first *expression* describes a left hip:

24136001 | hip joint structure |:

272741003 | laterality | =7771000 | left |

This next uses the "left hip" *expression* to describe a procedure to replace it:

397956004 | prosthetic arthroplasty of the hip |:

363704007 | procedure site | = (24136001 | hip joint structure | :272741003 | laterality | =7771000 | left |)

Applying a further grouped *refinement* to the above describes a procedure to replace a left hip by inserting a prosthesis. Note that this example mixes an ungrouped qualification and a grouped qualification. Where this is done, all ungrouped qualifications should appear before the groups. Note also that there is no comma between the last qualification and the first group.

397956004 | prosthetic arthroplasty of the hip |: 363704007 | procedure site | = (24136001 | hip joint structure | :272741003 | laterality | =7771000 | left |) {363699004 | direct device | =304120007 | total hip replacement prosthesis |,

260686004 | method | =257867005 | insertion - action |}

Finally, the above *expression* may be included within a contextual wrapper, to describe a procedure that has been performed on a patient to replace a left hip by inserting a prosthesis.

243796009 | situation with explicit context |: {363589002 | associated procedure | = (397956004 | prosthetic arthroplasty of the hip |: 363704007 | procedure site | = (24136001 | hip joint structure | :272741003 | laterality | =7771000 | left |) {363699004 | direct device | =304120007 | total hip replacement prosthesis |,

260686004 | method | =257867005 | insertion - action |}), 408730004 | procedure context | =385658003 | done |, 408731000 | temporal context | =410512000 | current or specified |, 408732007 | subject relationship context |=410604004 | subject of record | }

4.3.4.2 Expression in definition forms

An *expression* can be transformed to definition form and the representations applicable to this alternative form can then be applied. However, this approach is limited because several of the forms used to represent *concept* definitions do not support nesting.

4.3.4.3 Human-readable renderings

An *expression* may be rendered according to particular rules to generate human-readable representations.

Specific "simple" rules have been specified by *NHS* Connecting for Health in the UK. Alternative suggestions for more natural rendering have also been made to extend this initial outline proposal.

Advice on this topic may be added to future revisions of this guide.

4.3.5 Stated Relationships Guide

This part of the Guide provides information about the *Stated relationship file* and the Web Ontology Language (OWL) transformation.

4.3.5.1 Stated Relationships File

The *Stated Relationship File* contains the *stated form* of *SNOMED CT*. The *stated form* of a *Concept* is the *Description Logic* definition that is directly edited by authors or editors. It consists of the stated | is a | *relationships* plus the defining *relationships* that exist prior to running a *classifier* on the logic definitions. Therefore, the *stated form* of a *Concept* is represented by a collection of *relationships*: one or more | Is a | *relationships* and zero or more defining *relationships*.

The *Stated Relationship File* is in the same table format as the *Relationship File*, but the value of the *characteristicTypeId* field is | Stated relationship (core metadata concept) |.

The *stated form* enables implementers to test a *classifier* for consistency, by comparing the results of classification with the distributed *Relationship File*, which is the inferred form.

: Implementers should **not** use the *Stated Relationships File* unless they understand the implications of using this and provide software which makes *Description Logic* inferences from the *stated form*. The standard distribution form (the *Relationships File*) provides a *inferred view* which includes inferences derived from the *stated form*.

4.3.5.2 Description Logic (OWL or KRSS) Transform

The *Description Logic* Transform Script, written in the Perl language, performs a transform of the Stated *Relationships* into Web Ontology Language (OWL) format or KRSS format. There are two options for the syntax of the OWL output: RDF/XML, or OWL Functional Syntax. The RDF/XML is more verbose and results in a file approximately double the size of the Functional Syntax file.

4.3.5.2.1 Object Properties

SNOMED CT's attributes, the middle element of the *concept*- *Relationship*-*concept* triple, correspond to OWL Object Properties. The hierarchy under 410662002 | concept model attribute | contains all the attributes that have been approved for use as object properties. In addition, the *subtype Relationships* (i.e. | Is a | *Relationships*) between attributes in the | concept model attribute | hierarchy, as expressed in the *stated relationship file*, are used by the script to automatically generate the corresponding sub-property axioms in OWL. For example, | Procedure site - Direct | appears as a *subtype* of | PROCEDURE SITE | in the *stated relationship file*, and so the script automatically makes the OWL object property 'PROCEDURE SITE DIRECT' a sub-property of OWL object property 'PROCEDURE SITE'.

4.3.5.2.2 Relationship Grouping

When transforming *Relationships* to OWL or KRSS, all rows that have a RelationshipType that are allowed to be grouped, even if a particular row is ungrouped (i.e. even if the row has a RelationshipGroup value meaning ‘ungrouped’), must be nested under an existential restriction that represents the (potential) grouping. This existential restriction is labeled with the OWL object property called ‘Role group (attribute)’. It is just another attribute, in the sense that it has a *SNOMED CT identifier* and is named in the distributed *concept file* (609096000 | Role group |). In KRSS syntax, the stated definition of myConceptId1 with a stated definition that has a row with the triplet consisting of myConceptId1, myRelationshipType, myConceptId2 would translate into:

```
(defprimconcept myConceptId1
    (and parentConceptId
        (some RoleGroupId
            (some myRelationshipType myConceptId2))))
```

Attributes that are never grouped:

All RelationshipTypes are allowed to be grouped except | IS A |, and the following four:

- 123005000 | PART OF |
- 272741003 | LATERALITY |
- 411116001 | HAS DOSE FORM |
- 127489000 | HAS ACTIVE INGREDIENT |

4.3.5.2.3 Right Identities

There has historically been limited use of right identities, also known as property chains, in *SNOMED CT*. The one property chain that is in the current release is | DIRECT SUBSTANCE | o | HAS ACTIVE INGREDIENT | -> | DIRECT SUBSTANCE |. The OWL transform properly represents this property chain in the OWL 2 EL Profile. It is not yet represented in the *relationship file*, or anywhere else in standard *SNOMED CT distribution files*. This is a recognized deficiency which has not yet been addressed partly because there is only one such declaration, and no inferences in standard release are affected by this single right identity declaration.

4.3.5.2.4 Running the Perl transform script

Run the script according to the pattern:

```
perl <scriptfilename> <arg0> <arg1>
```

where

- <scriptfilename> is the name of the file containing the transform script
    - In the July 2014 release this Perl script file is named: tls2_StatedRelationshipsToOwlKRSS_INT_20140731.pl
- <arg0> can be KRSS, OWL, or OWLF
    - KRSS causes output to be formatted according to KRSS2 which is parsable by the OWL API 3.4.2, or by CEL or other classifiers
    - OWL causes output to be formatted according to OWL XML/RDF
    - OWLF causes output to be formatted according to the OWL functional syntax
- <arg1> is the directory containing the RF2 Snapshot subdirectories. If the current directory is RF2/Snapshot, then just use dot (".") to designate the current directory, as in the following example:

    ```
    C:\>perl tls2_StatedRelationshipsToOwlKRSS_INT_20140731.pl OWLF .
    ```

    The default output file name is snomedct_[arg0].owl, so if arg0 is OWLF, the output file will be snomedct_owlf.owl, but this can be changed in the Perl script itself (line 178) or the file can be renamed after generating it.

Alternatively you can separately supply arguments for all the file names (with their directories if necessary) :

```
perl <scriptfilename> <arg0> <arg1> <arg2> <arg3> <arg4> <arg5> <arg6>
```

- <arg0> can be KRSS, OWL, or OWLF
  - KRSS causes output to be formatted according to KRSS2 which is parsable by the OWL API 3.4.2, or by CEL or other classifiers
  - OWL causes output to be formatted according to OWL XML/RDF
  - OWLF causes output to be formatted according to the OWL functional syntax
- <arg1> is the directory path and name of the file containing the RF2 format *SNOMED CT Concept file* snapshot
  - In the July 2014 release this file is located in directory RF2Release/Snapshot/Terminology/ and is named: sct2_Concept_Snapshot_INT_20140731.txt
- <arg2> is the directory path and name of the file containing the RF2 format *SNOMED CT Description file* snapshot
  - In the July 2014 release this file is located in directory RF2Release/Snapshot/Terminology/ and is named: sct2_Description_Snapshot-en_INT_20140731.txt
- <arg3> is the directory path and name of the file containing the RF2 format *SNOMED CT Stated relationship file*
  - In the July 2014 release this file is located in directory RF2Release/Snapshot/Terminology/ and is named: sct2_StatedRelationship_Snapshot_INT_20140731.txt
- <arg4> is the directory path and name of the file containing the RF2 format Text Definitions Table snapshot
  - In the July 2014 release this file is located in directory RF2Release/Snapshot/Terminology/ and is named: sct2_TextDefinition_Snapshot-en_INT_20140731.txt
- <arg5> is the directory path and name of the file containing the RF2 Language Refset snapshot
  - In the July 2014 release this file is located in directory RF2Release/Snapshot/Refset/Language/ and is named: der2_cRefset_LanguageSnapshot-en_INT_20140731.txt
- <arg6> is the directory path and name of the output file. Any valid file name can be used.
  - for example: res_StatedOWLF_INT_20140731.owl

An example execution command on a Windows machine, from a command prompt at the directory RF2Release/Snapshot, to produce the *stated view* of *SNOMED CT* according to OWL Functional syntax, would then look like the following:

```
C:\> perl tls2_StatedRelationshipsToOwlKRSS_INT_20130731.pl OWLF
        Terminology/sct2_Concept_Snapshot_INT_20140731.txt
        Terminology/sct2_Description_Snapshot-en_INT_20140731.txt
        Terminology/sct2_StatedRelationship_Snapshot_INT_20140731.txt
        Terminology/sct2_TextDefinition_Snapshot-en_INT_20140731.txt
        Refset/Language/der2_cRefset_LanguageSnapshot-en_INT_20140731.txt
        res_StatedOWLF_INT_20140731.owl
```

4.3.5.2.5 Importing into an editor

Once the output file has been successfully created (e.g. res_StatedOWLF_INT_20140731.owl), an ontology editor that uses the OWL API should be able to import the file, assuming that the editor can handle very large files and that it is configured to use large amounts of memory, and your system has adequate memory (see FAQ below). The current version of the transform script has been tested with Protege running the OWL API version 3.4.2 and the OWL 2 Profile is OWL 2 EL. The table below presents the metrics that result from the July 2014 release.

Table 22: Metrics to Validate Import of SNOMED OWL, July 2014 International Release (20140731)

Protege Ontology Metrics	Value
Class count	299239
Object property count	62
DL expressivity	ALER
SubClassOf axioms count	229330
Equivalent class axioms count	69908
Sub object property axioms count	11
SubPropertyChainOf axioms count	1
Annotation Assertion axioms count	756457

4.3.5.2.6 SNOMED CT OWL Distribution FAQ

4.3.5.2.6.1 Access

1. Where do I obtain a copy of the OWL version of *SNOMED CT*?
   - You can currently **generate** an OWL version of *SNOMED CT* using the Perl script and '*stated view*' file in the standard distribution of *SNOMED CT*.
   - The Perl script and *Stated Relationships File* are distributed in the main release in different directories. The script is located in a folder called 'Resources/StatedRelationshipsToOwlKRSS/ ' and the RF2 snapshot files for *concepts*, *Descriptions* and stated *Relationships* are located in a folder called 'RF2Release/Snapshot/Terminology'. Prior to the January 2012 release, the transform was based on an RF1 format stated *Relationships* file - see documentation of prior releases for historical data and transform scripts.
2. What do you mean I need to 'generate' the OWL version of *SNOMED CT*?
   - The OWL version of *SNOMED CT* is currently not distributed with the core release. However you can generate a local OWL version of *SNOMED CT* by executing the Perl script mentioned above. The instructions for using the Perl script are included in the *Stated Relationships Guide* (part of the Technical Implementation Guide), and also as comments in the header of the file containing the Perl script, which can be viewed in your favorite text editor (e.g. Notepad, Wordpad, etc).
3. What do I need to generate the OWL version of *SNOMED CT*?
   - In order to generate the OWL version of *SNOMED CT*, you will need to have 'Perl' installed on your machine.
   - In addition to the *Stated Relationships file* and Perl script mentioned, you will also require the RF2 (*Release Format 2*) version of the *concept file*, *description file*, and language reference set. These files are named 'sct2_Concept_Snapshot_INT_yyyymmdd.txt' , 'sct2_Description_Snapshot-en_INT_yyyymmdd.txt', and 'der2_cRefset_LanguageSnapshot-en_INT_yyyymmdd.txt' in the January 2012 *International Release* and subsequently. The first two are found in the 'RF2Release/Snapshot/Terminology' folder, and the third is found in the 'RF2Release/Snapshot/Refset/Language' folder of the *International Release*.
4. I get errors when I try to generate the OWL version using the Perl script. What can I do?
   - Please check the following, before you report errors in the build process:
   - Ensure you have Perl properly installed on your machine.
   - Ensure that you are using versions of the Perl script, Stated *Relationships*, *Concepts*, and *Descriptions* all from the same release date and same *Release Format* (i.e. RF2). You will definitely get errors if you try to use a script designed for RF2 on RF1 format files, and vice versa. There is no guarantee of

backwards compatibility of the script - i.e. a version released for use with July 2013 RF2 files may not work with prior release RF2 files.

- Errors may be reported on the *IHTSDO Collaborative Space*, under the Implementation SIG site (in the General Discussions Forum).

4.3.5.2.6.2 Licensing

1. What are the license restrictions on the OWL version of *SNOMED CT*?

- There is a single world-wide license for *SNOMED CT* for all purposes, called the “affiliate license”. The same license applies to the OWL version of *SNOMED CT*. You can find it by following the highlighted link labelled "*SNOMED CT* Affiliate License Agreement" on the right hand side of the page at *www.ihtsdo.org/join-us/use-snomed-ct-licenses*

4.3.5.2.6.3 Importing and Visualization

1. How do I load and visualize *SNOMED CT* in OWL format?

- Though the KRSS or OWL files generated by the Perl script can be viewed in a text editor, in order to sensibly visualize the OWL release you require a tool like Protégé 4 (http://protege.stanford.edu/). Please note that version 4 (or later) of Protégé is required to load and visualize *SNOMED CT*.

2. Protégé crashes (or becomes unresponsive) when I try to visualize the class hierarchy on my machine!

- Protégé is known to take some time to generate the class hierarchy for display. It might be worthwhile increasing the memory allocation of Protégé before your start loading *SNOMED CT*. Please refer to the relevant Protégé documentation for exact details on increasing maximum memory available to Protégé.

3. Help, the hierarchies are rendered as *concept* IDs in Protégé! How can I change this into *fully specified names*?

- You need change the rendering options in Protégé to render using ‘labels’. In order to do that in Protégé 4, select ‘Render using annotation values’ from ‘Preferences/Renderer/Entity rendering’.

4.3.5.2.6.4 Classification

1. What DL reasoners are currently supported for classifying OWL version of *SNOMED CT*?

- There are Protégé 4.x plugins for several DL reasoners that can classify *SNOMED CT* provided the machine specifications listed below are met. These include Snorocket, ELK, and Fact++.

2. How long does it take to classify *SNOMED CT* in Protégé 4.x?

- That depends on the classifier and how fast your machine is. Both Snorocket and ELK are very fast, and complete in well under 30 seconds (actual clock time) on an adequately configured machine.

3. How do I use the DL Query Tab in Protégé 4 to create *postcoordinated expressions*?

a. We recommend looking at the Protege OWL Tutorial (http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf) for more information on using Protege 4.x to construct *expressions*. In the Protege world, *postcoordinated expressions* are referred to as DL *expressions*.
b. In order to create *postcoordinated expressions* in the DL query tab, you are required to use the Manchester syntax for the *expressions*. In order to understand the Manchester syntax, you will need to read and work the examples in the Protege OWL tutorial.

4. What can I do once I have classified *SNOMED CT* in Protégé 4.x?

- That depends on what you intended to do with a classified version of *SNOMED CT*. Within Protege 4.x, you can do subsumption testing over arbitrary DL *expressions* using the 'DL query tab' among other things. This feature might be used to implement subsumption testing over *postcoordinated expressions*.

4.3.5.2.6.5 Machine specification

1. What are the minimum specifications of machines for viewing loading and viewing *SNOMED CT* in OWL?
   - As a general rule, for reasonable performance, one would require a 64-bit machine, such as an Intel Core 2 Duo, with clock speed of 2GHz or more and 4GB of RAM to load the OWL version of *SNOMED CT* in Protégé.
   - The actual memory requirements might actually be smaller depending on your machine. Users have successfully loaded *SNOMED CT* on a 32-bit Mac OS X machine with 2GB RAM, and on a 32-bit Linux (Ubuntu) machine with 3GB RAM. However, display and editing performance is usually considered unacceptably slow when using these minimal configurations.
   - Loading and visualizing the OWL version of *SNOMED CT* using alternate methods might have different machine specifications.
2. What are the minimum specifications for classifying *SNOMED CT*?
   - It is believed that one would require a 64-bit machine with an Intel Core 2 Duo processor (or better) with 4GB of RAM to classify *SNOMED CT* using the classifiers bundled with Protégé 4. Users have successfully classified *SNOMED CT* on a 32-bit Mac OS X machine with 2GB RAM, and on a 32-bit Linux (Ubuntu) machine with 3GB of RAM.

4.3.5.2.6.6 Software

1. Can I bundle the OWL version of *SNOMED CT* in my open source software?
   - *SNOMED CT* is licensed under the affiliate license described above. *SNOMED CT* or any derivatives of *SNOMED CT* cannot be redistributed under any other license (including any form of open source license).
2. Am I allowed to make extensions or modification to the OWL release of *SNOMED CT* and include it in my software?
   - *SNOMED CT* is licensed under the affiliate license described above. *SNOMED CT* or any derivatives of *SNOMED CT* cannot be redistributed under any other license (including any form of open source license).
3. What API can I use to programmatically access the OWL version of *SNOMED CT*?
   - Though there are many candidate *APIs* available, most DL reasoners bundled with Protégé 4.x use the Manchester OWL API (*owlapi.sourceforge.net*). There are examples online on how to load an ontology. It might also be possible to use the Jena API (*jena.sourceforge.net*) to load the RDF/XML version of the file.

4.3.6 Other Representational Forms

This section summarizes some of the other forms in which *SNOMED CT components* and *expressions* may be represented. This includes some references to a selection of proprietary and standard representation which have been used or suggested for particular uses. Mention in this section is intended to be illustrative and does not represent endorsement. Additional suggestions that may be helpful to some implementers could be added in future.

4.3.6.1 Complete Concept Representations

Representation of the *concept* as a whole includes the definition of the *concept* but also includes additional properties of *concepts* and associated components such as *descriptions* and *Reference Sets*.

As a rule representations of complete *SNOMED CT concepts* will be specific to *SNOMED CT*. Some of these representations will be specified by *SNOMED* and others will be application specific designs building on the *SNOMED CT* specifications. If generic forms of representation are used then guidelines on how particular properties from *SNOMED* are represented are necessary.

4.3.6.1.1 SNOMED CT distribution files

The *Release File Specifications (5)* provide a form of representation for complete *concepts* (and other *components*).

The *release files* are designed efficient for large scale batch distribution and facilitate easy import into relational databases. They may need to be indexed and optimized to provide a practical implementable representation.

4.3.6.1.2 IHTSDO workbench internal format

The set of database table used by the *IHTSDO Workbench* to maintain *SNOMED CT* include a full representation of all types of *SNOMED CT Components*. The representation is closely aligned with *SNOMED CT Release Format 2*. However, additional data is stored to manage workflow and conflict resolution during the development process.

4.3.6.1.3 SNOMED CT Distribution XML

The XML distribution schema specified by *SNOMED* provides a form of representation for complete *concepts*(including associated components).

The XML distribution files can be used as an alternative to the *SNOMED CT distribution files*. However, they are particularly efficient for communication of individual *concepts* or sets of *concept* (e.g. for update change-sets).

4.3.6.1.4 Application internal

SNOMED CT enabled applications will usually have their own internal optimized representation of the *SNOMED* distribution information. This may simply be a relational database with a specified set of indices or it may be a significantly different form.

Examples of proprietary representation include the forms used internally by CliniClue (ClueData), Health *Language*, Apelon TDE and other implementations.

4.3.6.1.5 Various human-readable renderings

Concept information may be rendered in various ways to allow human visualization and understanding. These forms may include plain text, mark-up and graphical trees diagramming *relationships*. All of these renderings can be regarded as representations of complete *concepts* or their definitions.

4.3.6.2 Concept Definition Representations

See also: *Complete concept representations*

4.3.6.2.1 KRSS

KRSS is a general form for representing logical *descriptions*.

Transforms have been developed internally for producing KRSS representations of *SNOMED CT* definitions (see *Stated Relationships Guide*).

4.3.6.2.2 OWL

The Web Ontology Language (OWL) is a web-technology based approach to representation of logical *concept definitions*.

Transforms have been developed internally for producing OWL representations of *SNOMED CT* definitions (see *Stated Relationships Guide*).

4.3.6.2.3 Representing Definitions as Expressions

A *Concept* definition can also be represented as an *expression* (see *Representational Forms for Expressions*). One or more of the *supertype parent concepts* are represented as *focus concepts* and other defining *relationships* are represented as refining *attributes*.

4.3.6.2.4 Various human-readable renderings

*Concept definitions*may be rendered in various ways to allow human visualization and understanding. These forms may include plain text, mark-up and graphical trees diagramming *relationships*. All of these renderings can be regarded as representations of *concept definitions*.

4.3.7 Additional Reference Materials

4.3.7.1 Introduction

This section contains additional technical information that does not referenced by other part of this guide.

4.3.7.2 Unicode UTF-8 encoding

4.3.7.2.1 Introduction

UTF-8 is an efficient encoding of *Unicode* character - *strings* that recognizes the fact that the majority of text-based communications are in ASCII. It therefore optimizes the encoding of these characters.

Unicode is preferred to ASCII because it permits the inclusion of accents, scientific symbols and characters used in *languages* other than English. The *UTF-8* format is a standard encoding that provides the most efficient means of encoding 16-bit *Unicode* characters in cases where the majority of characters are in the ASCII range. Both *UTF-8* and the alternative *UTF-16* encoding issupported by all widely used operating systems and major applications (and has been for more than 15 years).

SNOMED CT uses the *UTF-8* representation of characters in *terms* and other text fields.

4.3.7.2.2 Character encoding

ASCII characters are encoded as a single byte.

- Greek, Hebrew, Arabic and most accented European characters are encoded as two bytes;
- All other characters are encoded as three bytes;
- The individual characters are encoded according to the following rules.

4.3.7.2.2.1 Single byte encoding

Characters in the *range* 'u+0000' to 'u+007f' are encoded as a single byte.

Table 23: UTF-8 Single Byte Encoding

byte 0	
0	bits 0-6

4.3.7.2.2.2 Two byte encoding

Characters in the *range* 'u+0080' to 'u+07ff' are encoded as two bytes.

Table 24: Two byte encoding

byte 0				byte 1		
1	1	0	bits 6-10	1	0	bits 0-5

4.3.7.2.2.3 Three byte encoding

Characters in the *range* 'u+0800' to 'u+ffff' are encoded as three bytes:

Table 25: UTF-8 Three Byte Encoding

byte 0					byte 1			byte 2		
1	1	1	0	bits 12-15	1	0	bits 6-11	1	0	bits 0-5

4.3.7.2.3 Notes on encoding rules

The first bits of each byte indicate the role of the byte. A zero bit terminates this role information. Thus possible byte values are:

Table 26: UTF-8 Encoding Rules

Bits	Byte value	*Role*
0???? ?? ?	000-127	Single byte encoding of a character
10??? ?? ?	128-191	Continuation of a multi-byte encoding
110?? ?? ?	192-223	First byte of a two byte character encoding
1110? ?? ?	224-239	First byte of a three byte character encoding
1111? ?? ?	240-255	Invalid in *UTF-8*

4.3.7.2.4 Example encoding

Table 27: UTF-8 Encoding Example

Character	S	C	T	®				
Unicode	0053	0043	0054	00AE		2462		
Bytes	01010011	01000011	01010100	11000010	10101110	11101111	10111111	10111111

4.3.7.3 Check-digit Computation

The *SCTID* (See *Component features - Identifiers* on page 46) includes a *check-digit*, which is generated using Verhoeff's dihedral check. This section explains the algorithm used and includes sample source code for generating and checking the *check-digit* in Java Script and Microsoft Visual Basic.

4.3.7.3.1 Verhoeff's Dihedral Group D5 Check

The mathematical *description* of this technique may appear complex but in practice it can be reduced to a pair of two-dimensional arrays, a single dimensional inverse array and a simple computational procedure. These three arrays are shown in the following tables.

- The first array contains the result of “Dihedral D5” multiplication;
- The second array consists of 8 rows of which two are defined while the rest are derived by applying the following formula: *F(i, j) = F(i - 1, F(1, j))*;
- The third array consists of a single row containing the inverse of the Dihedral D5 array it identifies the location of all the zero values in the first array.

Table 28: Results of Dihedral D5 multiplication

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	0	6	7	8	9	5
2	2	3	4	0	1	7	8	9	5	6
3	3	4	0	1	2	8	9	5	6	7
4	4	0	1	2	3	9	5	6	7	8
5	5	9	8	7	6	0	4	3	2	1
6	6	5	9	8	7	1	0	4	3	2
7	7	6	5	9	8	2	1	0	4	3
8	8	7	6	5	9	3	2	1	0	4
9	9	8	7	6	5	4	3	2	1	0

Table 29: The full array for Function F

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	1	2	3	4	5	6	7	8	9
1	1	5	7	6	2	8	3	0	9	4
2	5	8	0	3	7	9	6	1	4	2
3	8	9	1	6	0	4	3	5	2	7
4	9	4	5	3	1	2	6	8	7	0

	0	1	2	3	4	5	6	7	8	9
5	4	2	8	6	5	7	3	9	0	1
6	2	7	9	3	8	0	6	4	1	5
7	7	0	4	6	9	1	3	2	5	8

Table 30: The Inverse D5 array

0	1	2	3	4	5	6	7	8	9
0	4	3	2	1	5	6	7	8	9

The *Identifier* is checked by starting at the rightmost digit of the *Identifier* (the *check-digit* itself) and proceeding to the left processing each digit as follows:

- *Check* = ArrayDihedralD5 (*Check*, ArrayFunctionF((*Position* Modulus 8), *Digit*))

  Check = the running value of the check-sum (starts at zero and modified by each step).

  Position = the position of the digit (counted from the right starting at zero).

  Digit = the value of the digit.

  .

The final value of *Check* should be zero. Otherwise the check has failed.

When calculating the *check-digit* the same process is applied with a minor variation:

- *Position* is the position that the digit will have when the *check-digit* has been appended.
- The final value of *Check* is applied to the Inverse D5 array to find the correct *check-digit*.

  Check-digit = ***ArrayInverseD5*** (*Check*).

  .

4.3.7.3.2 Sample Java Script for computing Verhoeff's Dihedral Check

The script is presented here as part of an HTML page.

Note:

The code below can be used by copying all the lines in the above section into an HTML file and opening this with a web *browser*. From the HTML version of this guide the following link provides access to this file as an *web page*.

```
<!DOCTYPE html SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
   <head>
      <title>SNOMED CT Identifier Check</title>
      <style>
      body{font-family:Arial, Helvetica, sans-serif}
}
      </style>
      <meta content="text/html; charset=iso-8859-1" http-equiv="Content-Type"><meta>
         <script type="text/javascript" language="JavaScript">
```

```
        var FnF = new Array();
            FnF[0] = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9];
            FnF[1] = [1, 5, 7, 6, 2, 8, 3, 0, 9, 4];
            for ( var i = 2; i < 8; i++ )
            {
             FnF[i] = [,,,,,,,,,];
             for ( var j = 0; j < 10; j++ )
             FnF[i][j] = FnF[i - 1][FnF[1][j]];
            }
        var Dihedral = new Array(
           [0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
           [1, 2, 3, 4, 0, 6, 7, 8, 9, 5],
           [2, 3, 4, 0, 1, 7, 8, 9, 5, 6],
           [3, 4, 0, 1, 2, 8, 9, 5, 6, 7],
           [4, 0, 1, 2, 3, 9, 5, 6, 7, 8],
           [5, 9, 8, 7, 6, 0, 4, 3, 2, 1],
           [6, 5, 9, 8, 7, 1, 0, 4, 3, 2],
           [7, 6, 5, 9, 8, 2, 1, 0, 4, 3],
           [8, 7, 6, 5, 9, 3, 2, 1, 0, 4],
           [9, 8, 7, 6, 5, 4, 3, 2, 1, 0] );

        var InverseD5 = new Array(0, 4, 3, 2, 1, 5, 6, 7, 8, 9 );

      function VerhoeffCheck()

            {
            var check = 0;
            var IdValue = document.form.numcd.value;
            document.getElementById("out").innerText = "";
            document.getElementById("out").setAttribute("style", "color :red;");
document.getElementById("component").innerText ="Invalid partition";
document.getElementById("component").setAttribute("style", "color :green;");
document.getElementById("extnamespace").innerText ="No namespace";
document.getElementById("extnamespace").setAttribute("style", "color :red;");

            for ( var i=IdValue.length-1; i >=0; i-- )
            check = Dihedral[check][FnF[(IdValue.length-i-1) % 8][IdValue.charAt(i)]];
            if ( check != 0 ) { document.getElementById("out").innerText = "Check-digit ERROR"; }
            else if  (IdValue.length < 6) {document.getElementById("out").innerText = "SCTID too short";}
            else if  (IdValue.length > 18) {document.getElementById("out").innerText = "SCTID too long";}
else {document.getElementById("out").innerText = "Check-digit OK";
document.getElementById("out").setAttribute("style", "color :green;");
switch (IdValue.substr(IdValue.length-3,2))
{
case "00":
 document.getElementById("component").innerText ="Concept";
 document.getElementById("extnamespace").innerText ="International";
 break;
case "01":
 document.getElementById("component").innerText ="Description";
 document.getElementById("extnamespace").innerText ="International";
 break;
case "02":
 document.getElementById("component").innerText ="Relationship";
 document.getElementById("extnamespace").innerText ="International";
 break;
case "03":
 document.getElementById("component").innerText ="Subset (RF1)";
 document.getElementById("extnamespace").innerText ="International";
 break;
case "04":
 document.getElementById("component").innerText ="Cross Map Set (RF1)";
 document.getElementById("extnamespace").innerText ="International";
 break;
```

```
case "05":
 document.getElementById("component").innerText ="Cross Map Target (RF1)";
 document.getElementById("extnamespace").innerText ="International";
 break;
case "10":
 document.getElementById("component").innerText ="Concept";
 document.getElementById("extnamespace").innerText =IdValue.substr(IdValue.length-10,7);
 break;
case "11":
 document.getElementById("component").innerText ="Description";
 document.getElementById("extnamespace").innerText =IdValue.substr(IdValue.length-10,7);
 break;
case "12":
 document.getElementById("component").innerText ="Relationship";
 document.getElementById("extnamespace").innerText =IdValue.substr(IdValue.length-10,7);
 break;
case "13":
 document.getElementById("component").innerText ="Subset (RF1)";
 document.getElementById("extnamespace").innerText =IdValue.substr(IdValue.length-10,7);
 break;
case "14":
 document.getElementById("component").innerText ="Cross Map Set (RF1)";
 document.getElementById("extnamespace").innerText =IdValue.substr(IdValue.length-10,7);
 break;
case "15":
 document.getElementById("component").innerText ="Cross Map Target (RF1)";
 document.getElementById("extnamespace").innerText =IdValue.substr(IdValue.length-10,7);
 break;
default:
 document.getElementById("component").setAttribute("style", "color :red;");
}
if (document.getElementById("extnamespace").innerText=='International')
{document.getElementById("extnamespace").setAttribute("style", "color :green;");}
 else if (IdValue.length>10) {document.getElementById("extnamespace").setAttribute("style", "color :green;");}

else  {document.getElementById("extnamespace").innerText="Invalid Namespace";
}
        }
        }
    function VerhoeffCompute( )

        {
        var IdValue = document.form.num.value; var check = 0;
        document.form.numcd.value= "";
        for ( var i = IdValue.length-1; i >=0; i-- )
        check = Dihedral[check][FnF[(IdValue.length-i) % 8][IdValue.charAt(i)]];
        document.form.numcd.value = document.form.num.value + InverseD5[check];
        VerhoeffCheck();
        document.getElementById("out").innerText = "Computed check-digit";
}
        </script>
</head>
<body>
    <h1>SNOMED CT Identifier Check</h1>
    <form action="" name="form">
        <table border="1" width="441">
        <tr>
            <td width="212" height="25">
                Partial Identifier <br/>(without check-digit) 
            </td>
            <td width="115" height="25">
                <input name="num" size="18"/>
                    </td>
                    <td width="92" height="25">
```

```
                <input onclick= "VerhoeffCompute()" type="button" value="Compute"/>
            </td>
        </tr>
        <tr>
            <td width="212" height="35">
                SNOMED CT Identifier
            </td>
            <td width="115" height="35">
                <input name="numcd" size="18"/>
                    </td>
                    <td width="92" height="35">
                        <input onclick= "VerhoeffCheck()" type="button" value="Check"/>
                    </td>
        </tr>
        <tr>
            <td width="212" height="23">
                Result of check 
            </td>
            <td width="115" height="23" colspan="2" id="out">
            </td>         </tr>
        <tr>
            <td width="212" height="23">
                Component type
            </td>
            <td width="115" height="23" colspan="2" id="component">
            </td>         </tr>
        <tr>
            <td width="212" height="23">
                Namespace
            </td>
            <td width="115" height="23" colspan="2" id="extnamespace">
            </td>         </tr>

        </table>

        <p style="margin-left: 0; margin-right: 0">
            This Verhoeff checking part of this code was based on a webpage at:
        </p>
        <ul>
            <li>
                    <a href="http://www.augustana.ab.ca/~mohrj/algorithms/checkdigit.html">
                        http://www.augustana.ab.ca/~mohrj/algorithms/checkdigit.html
                    </a>
            </li>
        </ul>
    </form>
  </body>
</html>
```

4.3.7.3.3 Sample Visual Basic for computing Verhoeff's Dihedral Check

```
Option Explicit

Private Dihedral(9) As Variant
Private FnF(7) As Variant
Private InverseD5 As Variant

Public Function VerhoeffCheck(ByVal IdValue As String) As Boolean
'Check the supplied value and return true or false
   Dim tCheck As Integer, i As Integer

   VerhoeffArrayInit
```

```
    For i = Len(IdValue) To 1 Step -1
          tCheck = Dihedral(tCheck)(FnF((Len(IdValue) - i) Mod 8)(Val(Mid(IdValue, i, 1))))
    Next
    VerhoeffCheck = tCheck = 0
End Function

Public Function VerhoeffCompute(ByVal IdValue As String) As String
'Compute the check digit and return the identifier complete with check-digit
    Dim tCheck As Integer, i As Integer

    VerhoeffArrayInit
    For i = Len(IdValue) To 1 Step -1
        tCheck = Dihedral(tCheck)(FnF((Len(IdValue) - i + 1) Mod 8)(Val(Mid(IdValue, i, 1))))
    Next
    VerhoeffCompute = IdValue & InverseD5(tCheck)
End Function

Private Sub VerhoeffArrayInit()
'Create the arrays required

    Dim i As Integer, j As Integer

    'if already created exit here

     If VarType(InverseD5) >= vbArray Then Exit Sub

    'create the DihedralD5 array
    Dihedral(0) = Array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
    Dihedral(1) = Array(1, 2, 3, 4, 0, 6, 7, 8, 9, 5)
    Dihedral(2) = Array(2, 3, 4, 0, 1, 7, 8, 9, 5, 6)
    Dihedral(3) = Array(3, 4, 0, 1, 2, 8, 9, 5, 6, 7)
    Dihedral(4) = Array(4, 0, 1, 2, 3, 9, 5, 6, 7, 8)
    Dihedral(5) = Array(5, 9, 8, 7, 6, 0, 4, 3, 2, 1)
    Dihedral(6) = Array(6, 5, 9, 8, 7, 1, 0, 4, 3, 2)
    Dihedral(7) = Array(7, 6, 5, 9, 8, 2, 1, 0, 4, 3)
    Dihedral(8) = Array(8, 7, 6, 5, 9, 3, 2, 1, 0, 4)
    Dihedral(9) = Array(9, 8, 7, 6, 5, 4, 3, 2, 1, 0)

    'create the FunctionF array

    FnF(0) = Array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
    FnF(1) = Array(1, 5, 7, 6, 2, 8, 3, 0, 9, 4)

    'compute the rest of the FunctionF array

     For i = 2 To 7
        FnF(i) = Array(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
        For j = 0 To 9
             FnF(i)(j) = FnF(i - 1)(FnF(1)(j))
        Next
    Next

    'Create the InverseD5 array
    InverseD5 = Array("0", "4", "3", "2", "1", "5", "6", "7", "8", "9")

End Sub
```

4.3.7.3.4 Reasons for using a check-digit

Although a user should rarely type the *SCTID*, experience suggests that from time to time this will happen. A user may also copy and paste an *SCTID*. There is a significant risk of errors in these processes and inclusion of a *check-digit* is intended to reduce the risk of such errors passing undetected. The choice of *check-digit* algorithm has been made to maximize the detection of common typographical errors. These have been analyzed by in a paper by J. Verhoeff ("Error Detecting Decimal Codes", *Mathematical Center Tract 29*, The Mathematical Center , Amsterdam, 1969) and subsequently cited in Wagner and Putter, ("Error Detecting Decimal Digits",*CACM*, Vol 32, No. 1, January 1989). These papers give a detailed categorization of the sorts of errors humans make in dealing with decimal numbers, based on a study of 12000 errors:

- single errors: a becomes b (60% to 95% of all errors).
- omitting or adding a digit (10% to 20%).
- adjacent transpositions: ab becomes ba (10% to 20%).
- twin errors: aa becomes bb (0.5% to 1.5%).
- jump transpositions: acb becomes bca (0.5% to 1.5%).
- jump twin errors: aca becomes bcb (below 1%).
- phonetic errors: a0 becomes 1a -similar pronunciation e.g. thirty or thirteen (0.5% to 1.5%).

In the explanations above, a is not equal to b, but c can be any decimal digit.

4.3.7.3.4.1 A brief comparison of check-digit effectiveness

4.3.7.3.4.1.1 The IBM Check

The check-sums used for credit cards (the IBM check) picks up the most common errors but miss some adjacent transpositions and many jump transpositions. Assuming the pattern of errors described above, on average it will miss between 4% and 5% of expected errors.

4.3.7.3.4.1.2 The ISBN Check (Modulus 11)

The ISBN modulus 11 (used for *UK NHS* number) picks up more errors than the IBM checksum. Leaving 2% to 3% of errors undetected. However, it generates a check-sum value of 0 to 10 and thus cannot be represented as a single *check-digit* in about 9% of cases. The ISBN convention is to use "X" to represent the *check-digit* value 10 but this is incompatible with an *integer* representation. The *UK NHS* number uses this check-sum but regards and number generating a check-sum of 10 as an invalid *Identifier*. This approach could be applied to the *SCTID* but this would render 9% of possible values unusable in each partition and *namespace*. This would prevent a simple sequence of values from being allocated as the "item *Identifier*" within each *namespace*. More significantly the unusable item *Identifiers* would differ in each *namespace* or partition and this would prevent simple transpositions of item *Identifiers* between partitions and *namespaces*. Partitions could be a useful way of distinguishing developmental and released components and revising the partition and recalculating the *check-digit* would then be an elegant way to activate these components for a distribution version. It seems unwise to prevent future development and maintenance by using a check-sum that will prevent this.

4.3.7.3.4.1.3 Verhoeff's Check

Verhoeff's check catches all single errors, all adjacent transpositions, over 95% of twin errors, over 94% of jump transpositions and jump twin errors, and most phonetic errors. Therefore, like modulus 11, the Verhoeff check reduces the undetected error rate to 2% or 3%. Unlike modulus 11, it does this using a single decimal *check-digit* and without limiting the range of valid numbers.

The majority of the undetected errors with both modulus 11 and Verhoeff result from additions or omissions of digits. Any *check-digit* methods is likely to miss 10% of such errors and since these comprise 10% to 20%. The Verhoeff scheme also misses four jump twin errors involving digits with a difference of 5 (i.e. 050 vs. 505, 161 vs. 616, 272 vs. 727, and 494 vs. 949).

4.3.7.4 Search Support Tables

4.3.7.4.1 Overview

Effective implementation of *SNOMED CT* depends on the ease and speed with which users can locate the *terms* and *Concepts* that they wish to use. An essential contribution to meeting this requirement is the ability to perform rapid and flexible text searches.

A set of word search tables (indexes) are included in the Developer Toolkit. These tables are designed to facilitate development of effective search facilities while reducing duplication of effort. However, neither these tables, nor indices derived from them, are sufficient to meet the full range of search requirements. Meeting the needs of different users for appropriate methods for locating particular *Concepts* is an area in which competitive development is expected and welcomed. Developers may choose to use some or all of the word search tables distributed with *SNOMED CT* or may develop their own solutions independent of these tables.

The intention of the word search tables is to identify candidate matches among the *Descriptions* (or *Concepts*) of *SNOMED CT*. An application or coding engine will apply further filtering to these candidate matches to identify the matches to be selected or displayed. A balance must be made between specificity and completeness of a search. The *keyword* algorithm is intended to maximize the likelihood that the required *Concept* will be included in the candidate matches rather than to achieve precision.

Applications may filter candidate matches using techniques that are many and varied. Some may take account of non-textual characteristics (e.g. *Reference Sets*, *subtype Relationships* or *Relationships*) while others use more complex textual techniques (e.g. word *order* dependence, case dependence, complete phrase matching, regular-*expression* matching, Soundex). These extended text search techniques are beyond the scope of the *keyword* generation algorithm.

The algorithm for *keyword* generation is only applicable for English and other western European *languages*. It is not intended to apply to Russian, Greek, Slavic or to any non-European *languages*.

Please refer to the *Technical Implementation Guide* for additional search implementation guidance.

4.3.7.4.2 Search index - structure diagram

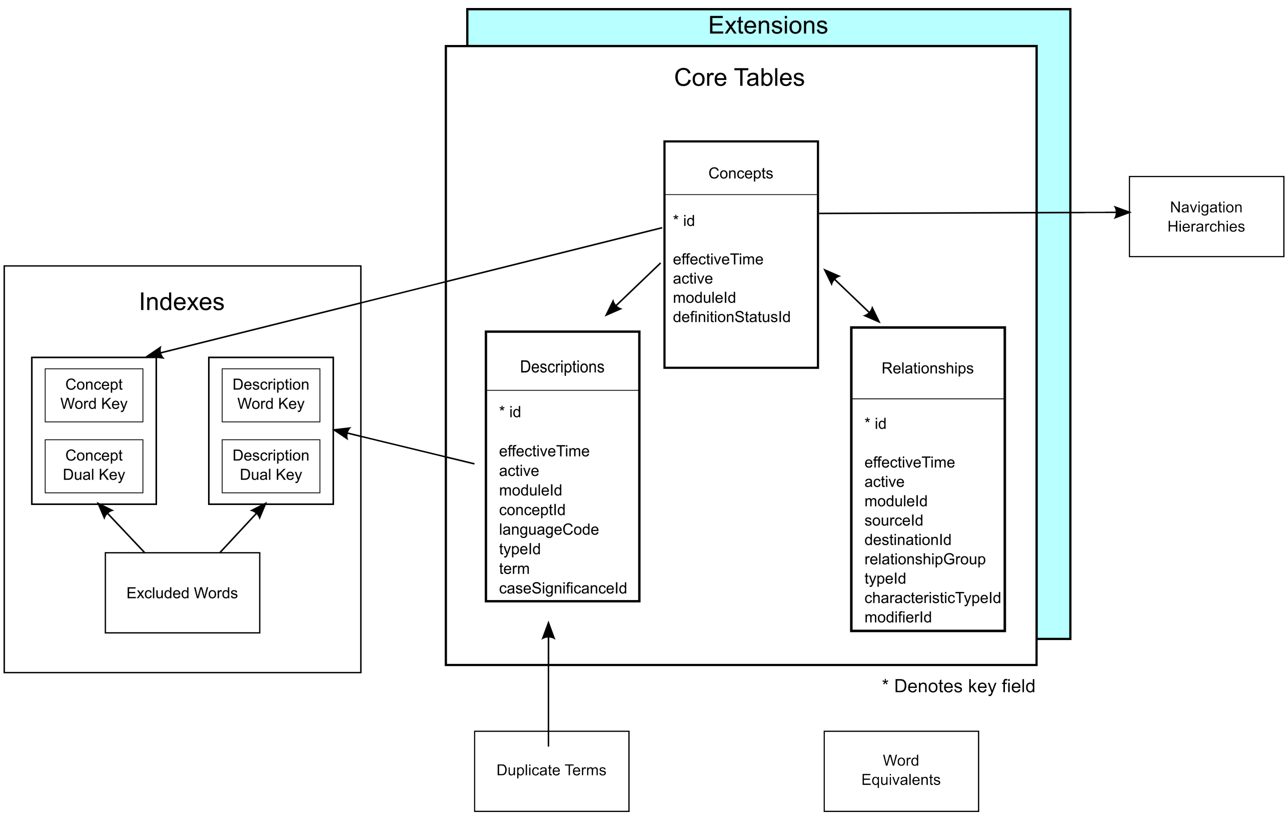


Figure 35: Search Index Overview (RF2)

4.3.7.4.3 Word Search Tables - Summary

The following five tables are included in the *Developer Toolkit* of *SNOMED CT*. These tables are derived from the *SNOMED CT Descriptions* Table. The LanguageCode of the *Description file* is used to choose only *descriptions* for a *language*.

Table 31: Summary of Word Search Tables

Table	***Description***
Excluded Words Table	Each row in this table is a word excluded from the list of possible *keywords* and *dualkeys*. Words are excluded if they are frequently used and are so limited in semantic specificity that they impair rather than enhance searches.
DescWordKey Table	Each row in this table is a word followed by a reference to a *Description* in which this word appears.
ConcWordKey Table	Each row in this table is a word followed by a reference to a *Concept*. A *Concept* is referenced if the word appears anywhere in the combination of the *Fully Specified Name* with the current valid *Preferred Term* and *Synonyms*.
DescDualKey Table	Each row in this table is a six-character *string* representing the first three letters of a pair of words followed by a reference to a *Description* in which these two words appear.
ConcDualKey Table	Each row in this table is a six-character *string* representing the first three letters of a pair of words followed by a reference to a *Concept*. A *Concept* is referenced if both words appear anywhere in the combination of the *Fully Specified Name* with the current valid *Preferred Term* and *Synonyms*.

All *keywords* are regarded as case independent and are presented in the word search tables in upper case. Case dependent searching can be applied by appropriately filtering the candidate matches.

4.3.7.4.4 Word Equivalents

The *Word Equivalent* Table is included in the Developer Toolkit of *SNOMED CT*. It supports enhanced searches that take into account semantically similar words such as KIDNEY and RENAL. It also provides commonly used abbreviations. This table can be used by implementers to offer additional search capability in applications without greatly increasing the volume of *synonyms*. It is not intended as a comprehensive dictionary of words. Many searches can be completed without using this table; like the other word search tables, it is completely optional and can be used as an example of a capability that may be customized and extended by *SNOMED CT* implementers.

4.3.7.4.4.1 Word Equivalents Tables - Summary

Table 32: Word Equivalents Table

Key Fields	
WordBlockNumber	A 32-bit *integer* shared by a set of equivalent words or phrases. The *WordBlockNumber* links together several rows that have an identical or similar meaning.

Key Fields	
WordText	A word, phrase, acronym or abbreviation that is equivalent to the *WordText* of other rows that share the same WordBlockId.

Data Fields	
WordType	An *integer* indicating the type of *equivalence*
WordRole	An *integer* indicating the usual role of this word. This should be considered if attempting to find a *postcoordinated* combination of *Concepts* that matches a phrase.

Chapter 5

5 Release File Specifications

This part of the guide specifies the formats in which *SNOMED CT* is provided to licensees (*IHTSDO affiliates*).

Note: For *SNOMED CT* licensing information and details about the availability of *release files* contact either the *IHTSDO* or the *IHTSDO Member* in your country. Contact details are available on the *IHTSDO* web site: *www.ihtsdo.org*.

5.1 Release File Formats

Currently, during a transitional period, there are two distinct *Release Formats* for *SNOMED CT*:

- *Release Format 2* (*RF2*): The new standard distribution format for *SNOMED CT*. This was developed in response to extensive feedback on its predecessor and will replace *RF1* during 2012.
- *Release Format 1* (*RF1*): The specification in which *SNOMED CT* has been provided since its first release in 2002. This format will be phased out, but support of applications that require *RF1* format files will be available using a conversion application developed and supplied by the *IHTSDO*.

The key enhancements in *RF2* are:

- More robust and consistent version representation;
- *Reference sets*, provides a more easily extensible and maintainable replacement for *RF1* representations of *subsets* and *maps*;
- Use of an added *hierarchy* to represent metadata about the structure of *SNOMED CT* itself.

Both *Release Formats* represent:

- The components of *SNOMED CT*:
  - *Concepts*
  - *Descriptions*
  - *Relationships*
- Additional *derivatives* that provide standard representations of :
  - *Subsets* of *concepts* or *descriptions*;
  - *Value sets* consisting of *concepts* and/or *expressions*;
  - *Language and dialect* preferences;
  - *Alternative hierarchies*;
  - *Maps* to other codes and classifications.

Both *Release Formats* are provided in:

- Tab-delimited text files;
- Represent character content in accordance with the *Unicode UTF-8* specification;
- Use *SNOMED CT Identifiers* as the permanent *Identifier* of released core components;

- Support *extensions* to the *International Release* using *namespaces* allocated to licensees to denote the provenance of added components and to ensure *Identifier* uniqueness.

5.2 SNOMED CT Editions, Extensions, Releases and Modules

SNOMED CT is delivered as sets of files containing terminology components and derivatives. The format, content and names of the files delivered conform to SNOMED CT specification and guidelines published by the IHTSDO.

- Components represent the content on the terminology.
  - The standard SNOMED CT representation for content is as three interrelated files. The Concept file contains unique identifiers for clinical ideas, the Description file links human readable terms with identified concepts and the Relationship file represents associations between identified concepts.
- Derivatives facilitate the effective use of the terminology.
  - The standard SNOMED CT representation for derivatives is a consistent but flexible file format, known as the reference set format. Reference sets can be used for a wide range of purposes including subsets, language preferences, ordered lists, hierarchies, annotations and mapping to or from other terminologies, classifications and code systems.

IHTSDO maintains and delivers shared content and derivatives that provide the foundation of the SNOMED CT. This is known as the SNOMED CT International Edition.

IHTSDO also authorizes Members and Affiliates to maintain and deliver additional components and derivatives known as SNOMED CT Extensions.

- Any organization maintaining an Extension needs to have a namespace identifier allocated to them by the IHTSDO. Every component created must be allocated a new permanent SNOMED CT identifier which is in the originating organization's namespace allocated. The namespace identifier of the originating organization forms part of every component identifier allocated.

IHTSDO Members may maintain and deliver additional terminology components and derivatives that adapt the terminology to meet specific national requirements (National Extension).

Examples:

A National Extension may include:

1. translation into the national language or adaption to a national dialect;
2. additional content to support national policy objectives, a national drug dictionary or other specific requirements;
3. derivatives that configure use of SNOMED CT content by specifying subsets of content to be used for particular purposes;
4. derivatives that map other code systems used in that country to or from SNOMED CT.

IHTSDO Affiliates may also maintain and deliver additional terminology components and derivatives that adapt the terminology to meet the needs of a particular organization, customer or software solution (Affiliate Extension).

Examples: An Affiliate Extension may include:

1. additional content enable a health provider organization or clinical specialty group to address its priority use cases;
2. derivatives that configure use of SNOMED CT in ways that reflect the needs of a health provider organization or specialty;

3. derivatives that configure the way SNOMED CT is used or presented to different customers using a particular software applications;
4. derivatives that map local or proprietary code systems to or from SNOMED CT.

The SNOMED CT International Edition can be used without any Extensions. However, a SNOMED CT Extension cannot be used on its own because all Extensions are dependent on the International Edition and some Extensions are also dependent on other Extensions. Therefore, for each Extension there is a corresponding Edition that includes the Extension, the International Edition and any other Extensions on which it depends.

Examples:

1. Healthitia, an IHTSDO Member, produces the Healthitia National Extension. Affiliates in Healthitia use the Healthitia National Edition (the International Edition and the Healthitia National Extension).
2. Anyville Hospital in Healthitia is an IHTSDO Affiliate. It produces its a local Anyville Extension for use in the hospital and this Extension also depends upon the Healthitia National Extension. Therefore, the Anyville Edition consists of the International Edition, the Healthitia National Extension and the Anyville Extension;
3. AcmeGest an international provider of antenatal care systems is an IHTSDO Affiliate and uses the AcmeGest Extension to support some of its specialist functionality. This Extension only depends on the International Edition, therefore the AcmeGest Edition consist of the International Edition and the AcmeGest Extension.
4. The obstetric department of Anyville Hospital in Healthia uses the AcmeGest system., so it requires the AcmeGest Extension as well as the Anyville Edition. The AcmeGest Extension has a dependency on the International Edition but this should only be resolved once.

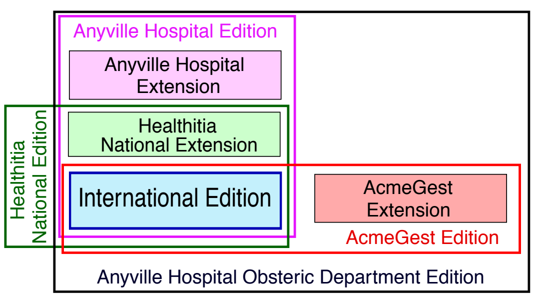


Figure 36: Illustration of the relationships between Extensions and Editions

For use in particular countries or institutions there may be advantages in distribution a complete Edition. However, as the last of example above illustrates, different Editions may share a common dependency on the International Edition. All organizations that maintain Extensions should make their Extension available as a separate set of files, even if they also provide a pre-merged Edition. This allow validation of the constituent parts of the Edition and also supports merges to produce bespoke Editions that combine several Extensions.

The IHTSDO provides regular updates of the SNOMED CT International Edition. Similarly, Members and Affiliates that maintain Extensions will from time to time release updated versions of their Extensions. Therefore,

The SNOMED CT release file specifications define three different release file types:

- Full Release: containing the complete history of every component
- Snapshot Release: containing the current state of every component
- Delta Release (containing only the additions and changes since the previous release

The International Edition is provided in all three release types. However, as the Snapshot and the Delta can be generated from the Full, the specification only requires that organizations that maintain Extensions provide a Full Release of their Extension.

5.3 Release Format 2 - Introduction

This section describes the *release file* structure of SNOMED CT®. This file structure is referred to as *Release Format 2* (*RF2*) to distinguish it from the previous *Release Format* (*RF1*) in which *SNOMED CT* was distributed between its first release in 2002 and 2012.

The *Release Format 2* specification is divided into two parts. The *RF2 Core Component Guide* is concerned with the representation of the *Concepts*, *Descriptions* and *Relationships* that contain the primary content of *SNOMED CT*. The *RF2 - Reference Sets Guide* specifies the common extensible pattern that is used to add additional information related to the core components. It also describes the ways in which this pattern is used to represent essential functionality (such as language specificity, historical status changes and associations) and optional additional functionality (including subsets, *mapping* and alternative *navigation hierarchies*).

In 2012 *RF2* became the primary *Release Format* for *SNOMED CT*. Implementers requiring data in the *RF1* format during a transitional period are initially being supported by use of conversion application developed by the *IHTSDO* which generates *RF1* from *RF2* files. It is important to note the *RF2* features are not supported by the converted *RF1* files and, eventually *RF1* support will be withdrawn.

5.4 SNOMED CT - File Naming Conventions

The file naming convention specified in this section applies to all *IHTSDO release files* starting with the January 2010 *International release*.

The specification provides the following benefits:

- A consistent naming convention across the *International edition* and each *National edition*.
- Predictable file naming, providing a stable structure for naming over time between releases.
- A standard way to identify the source country and *namespace* by which a *release file* is owned.
- A consistent versioning mechanism.
- An easy human readable way to identify the content of a file, at a summary level.
- A mechanism for identifying the type of information stored in a *release file* (e.g. documentation, tooling, etc.).
- Guidance on file naming for *release files* in non-English *extensions*.
- Assurance that names will be unique across the *International release* and releases from individual *National release* centers and across separate releases from each center over time.
- An upgrade path, to enable use of the same naming convention with the new *release format* (*RF2*), while enabling easy identification of whether a file is in *RF1* or *RF2* format, and avoiding naming clashes.

Quality Assurance checks, to ensure that this naming convention is enforced, will be performed as part of the *International release* process. It is expected that equivalent checks will be performed as part of each *National Release* Center 's release process.

Note: Prior to January 2010 other naming strategies were used. Implementers who need to review earlier releases should consult the documentation that accompanied the release that they need to review.

5.4.1 File Naming Convention - Overview

5.4.1.1 General File Naming Pattern

The basic pattern for *SNOMED CT release file* names consists of five elements, each separated by an underscore ("_ ") and followed by a full stop (" .") and a file *extension*:

<FileType>_<ContentType>_<ContentSubType>_<Country|Namespace>_<VersionDate>.<Extension>

Each element in the above structure is described in more detail in subsequent sections.

5.4.1.2 General Naming Rules

The following rules apply generally to all elements of the file name:

- All elements are mandatory and may not have a null value;
- Elements of the file name may only contain alphanumeric characters, with the exception of hyphens (" **-**") used in connection with *language* codes (see detail for the ContentSubType element below);
- All text should be in US English, except as explicitly allowed below;
- Abbreviations should not be used, except for specified codes or tags;
- The maximum length of a file name (including separators and *extension*) is 128 characters.

5.4.1.3 Rules for "Readme" Files

Readme files distributed as part of a *SNOMED CT release* have their own specific naming convention, as shown below:

Language is the *language* code for the *language* of the Readme file, as specified below for the ContentSubType element, and the VersionDate corresponds to the version date of the release.

5.4.2 FileType element

5.4.2.1 Description

The FileType element of the filename designates the type and intended use of the *release file*. It consists of a 3-5 letter code and must be lowercase.

5.4.2.2 Rules

Allowable FileType codes are shown in the table below:

Table 33: Allowable File Type Codes

Code	File Type *Description*
"sct" + <format tag>	Terminology Data File
"der" + <format tag>	Derivative Work Data File
"res" (+ <format tag>)	Implementation Resource Data File
"tls" (+<format tag>)	Implementation Resource Tool
"doc" (+<format tag>)	Documentation
"z" + Code	Archival/Unsupported File (e.g. zsct)
"x" + Code	Test/Beta Release File (e.g. xder)

The allowable file types are described in more detail below:

- **Terminology Data File**(" **sct**") - the set of data files that make up the *SNOMED CT* terminology. These are:

    - *Concepts file*
    - *Descriptions file*
    - *Relationship File*
    - *Stated Relationships File*
    - *Identifier file* (currently empty)

- **Derivative Work Data File**(" **der**") - data files that make up a *SNOMED CT* "*derivative* work" (a product for use in conjunction with *SNOMED CT* that cannot be effectively used without the terminology - such as *subsets* or *maps*). Examples of the files within this group include:
    - *Reference Set Files*.
- **Implementation Resource Data File**(" **res**") - data files intended to support developers with the implementation of *SNOMED CT*, but that are not necessarily useful to end-users. Examples from the current *International Release* include:
- **Implementation Resource Tool**(" **tls**") - software tools or other files that do not contain original *SNOMED CT* content (i.e. that is not also held elsewhere in the release), but can be of use to implementers. If such files cannot comply with this naming convention (for example, if some other standard applies), then those files should be distributed as part of a ZIP file archive that does conform to this file naming convention.
- **Documentation**(" **doc**") - documents defining *SNOMED CT* standards, policies and guidelines, as well as documentation for files or products included in a *SNOMED CT release*. Most, but not all, files in this group are released in a PDF format.
- **Archival/Unsupported File**("**zsct**", "**zder**", "**zres**", "**ztls**") - files that are not currently supported or updated, but may be of some use to implementers. These files should only be used with caution and after appropriate review and validation. The letter "**z**" is inserted in front of the usual FileType code for these files (i.e. "**z**" + "**sct**", "**der**", "**res**" or " **tls**"). Examples from the *current International Release* include:
    - *SNOMED* 3.5 to *SNOMED RT* bridge file;
    - *SNOMED* 2 to *SNOMED RT* bridge file.
- **Test/Beta *Release File***("**xsct**", "**xder**", "**xres**", "**xtls**") - files distributed as part of a test/beta release, or as a "technology preview". These files should only be used for review and evaluation purposes. The letter "**x**" is inserted in front of the usual FileType code for these files (i.e. "**x**" + "**sct**", "**der**", "**res**" or "**tls**").

5.4.2.3 Format Tags

A *Release Format* tag must be appended at the end of the three-letter FileType code if the file named is dependent on a particular *Release Format* specification. The allowable *Release Format* tags are:

- For files that are part of the current *Release Format* (*RF2*), or applicable only to the *RF2 Release Format*, the number "**2** " is appended to the FileType code (e.g. "**sct2**", "**der2**", "**res2**").
- For files that are part of the now obsolete *RF1 Release Format*, or applicable only to that *Release Format*, the number "**1** " is appended to the FileType code (e.g. "**sct1**", "**der1**", "**tls1**").
- If the file is not specific to either *Release Format*, the three-letter FileType code should be used without a *Release Format* tag (e.g. "**res**", "**tls**" or "**doc**").
- The FileType code for all terminology and *Derivative* Work data files ("**sct** " or "**der**") must include a *Release Format* tag ("**1**" or "**2**"). For other file types, the *Release Format* tag is optional.

5.4.3 ContentType element

5.4.3.1 Description

The ContentType element of the filename describes the content and purpose of the file. It consists of 2-48 alphanumeric characters in camel case.

5.4.3.2 Rules

The content of this element depends on the first element (FileType) of the filename, as described below:

Possible values for the *Release Format 2* are:

- *Concept*
- *Description*
- *Relationship*
- *StatedRelationship*
- *Identifier*;
- *Refset*

For Data files where the ContentType element is " *Refset* ", the ContentType element will also describe the format and content of the *reference set* member file. Each file of ContentType " *Refset*" will hold zero or more additional columns, each having one of the following types:

- *Component*
- *String*
- *Integer*

Lower case "**c**", "**s**" and " **i**" will be used as abbreviations (for *component*, *String* and *Integer* respectively) to describe the format of the additional columns that will be appended to the end of each row in the *Refset* file. These abbreviations will prefix the ContentType element, as shown in the examples below:

- **cRefset** - a *Refset* file with one additional column, holding *component* values;
- **ssRefset** - a *Refset* file with two additional columns, both holding *String* values;
- **ciRefset** - a *Refset* file also with two additional columns, the first holding component values and the second holding *integer* values;
- **For ImplementationResource Tools**("**tls**") - the value of the ContentType element may be determined on a case-by-case basis but, in conjunction with the ContentSubType element, should be adequate to identify the content and purpose of the file;
- **For Documentation**("**doc**") - the title of the document, which may be abridged but should not be abbreviated, should be used as the value for the ContentType element;
- **For Archival & Test/Beta Files**("**z**"+ code or "**x**"+ code) - the value of the ContentType element should be determined according to the rules for a normal file of the same type ("**sct**", "**der**", "**res**" or "**tls**").

5.4.4 ContentSubType element

5.4.4.1 Description

The ContentSubType element of the filename provides additional information to describe the content and purpose of the file, including the *language / dialect*, where appropriate. Its format is 2-48 alphanumeric characters in camel case (except for the capitalization rules specified below for *language* code). Hyphen ("**-**") is a permitted character in conjunction with a *language* code, as described below.

5.4.4.2 Rules

The content of this element depends on the first element type (FileType), and these rules are described in more detail below:

- **Data Files**("**sct**", "**der**" or " **res**") - as a result of *RF2*'s state-valid history tracking capability, it is possible to perform a number of different releases of *SNOMED CT* content in the *RF2* format:
  - A "**Full**" release of each file containing every version of every component ever released.
  - A "**Snapshot**" release, containing only the most recent version of every component ever released (both *active* and *inactive components*).

- A "**Delta**" release, containing only component versions created since the last release. Each component version represents a new component or a change in an existing component.

Each *RF2* ContentSubType element must be postfixed with a *Release Type* flag with a value of: "**Full**", "**Snapshot**" or "**Delta**". This flag should always appear at the end of the ContentSubType element, unless ContentSubType includes a *language* code (see below). If a *language* code is present in the ContentSubType element, the *Release Type* flag will appear immediately before the *language* code.

- **Implementation Resource Tool** ("**tls**") - the value of this element may be determined on a case-by-case basis but, in conjunction with the ContentType element, should be adequate to identify the content and purpose of the file. If appropriate, the element may contain a *status* tag with one of the values described below under Documentation.
- **Documentation** ("**doc**") - the element should contain at least two components: a *status* tag and a *language* code (see above). Additional components may be added to this element if necessary to fully identify the document. Possible values for the document *status* tag are:
  - "**Current**" - indicates that the document is up-to-date and complete for the current release of *SNOMED CT*, as indicated by the VersionDate element;
  - "**Draft**" - indicates that the document is a draft version; it may be incomplete and has not been approved in a final version;
  - "**Review**" - indicates that the document has been released for review and comments from *SNOMED CT* users and other stakeholders.
- **Archival** & **Test/Beta File** ("**z**"+ code or "**x**"+ code) - the value of the element should be determined according to the rules for a normal file of the same type ("**sct**", "**der**", "**res**" or "**tls**").

5.4.4.3 Language Usage

For files released as part of a National or local release, and which do not appear in the *SNOMED CT International Release*, the value of the ContentSubType element may be given in a *language* other than English, with the following limitation:

- Any of the four sets of defined values for the ContentSubType element that are present in the file name may not be translated, but must appear as specified herein. These are: *language* code, *Release Type* flags ("**Full**", "**Snapshot**", "**Delta**"), placeholders ("**Core**", "**National**", "**Local**"), and *status* tags ("**Current**", "**Draft**", "**Review**").

5.4.4.4 Language Codes

Where it is necessary to specify the *language* of a file, a *language* code must be included in the ContentSubType element. A *language* code is a *string* identifying the *language* and, if appropriate, the *dialect* of a file, and consists of a code and optionally a sub-code. If a sub-code is present it is separated from the code by a hyphen ("**-** ").

The code is the two-character *ISO* 639-1 *language* code. *ISO* 639 is the International Standard for "Codes for the representation of names of *languages*". The sub code is a *string* of upper-case letters that represent the *dialect*. This deliberately mirrors the W3C approach and will either be:

- If the *dialect* is general to an entire country, the two-letter *ISO* 3166 country code is used. *ISO* 3166 is the International Standard for "Codes for the representation of names of countries".
- If *dialects* are used that are less common or not country or *language* linked, the IANA approach is used; this code consists of a *string* of more than two letters. IANA is the Internet Assigned Numbers Authority.

This structure follows Internet conventions. Examples: "**en**" for English, "**es**" for Spanish, " **en-US** " for United States English, " **en-GB** " for British English.

If the ContentSubType includes more than one component (e.g. document *status* and a *language* code), the *language* code must be the last component in theContentSubType element and should be preceded by a hyphen ("**-**") placed before the *language* code.

5.4.5 Country|Namespace element

5.4.5.1 Description

The Country | *Namespace* element of the filename helps to identify the organization responsible for developing and maintaining the file. Its format is 2-10 alphanumeric characters consisting of 0, 2 or 3 upper-case letters followed by 0 or 7 digits.

5.4.5.2 Rules

The following rules apply to the content of this element:

- Letters, if present, are either the *ISO* -3166 2-character country code for an *IHTSDO Member* country or "INT" for files that are part of the *IHTSDO*'s *International Release* of *SNOMED CT*;
- Digits, if present, are a *SNOMED CT Namespace Identifier*.

Valid combinations are:

- 2 characters only - the file is part of a Member *National Release*, but not part of a specific *Namespace* - this combination is not valid for Data Files ("**sct** ","**der**" or "**res**");
- 3 characters only ("**INT**") - the file is part of the *IHTSDO International Release* and belongs to the International *Namespace*;
- 2 characters and 7 digits - the file is part of a Member *National Release* and belongs to the specified *Namespace*;
- 3 characters and 7 digits - the file is an optional part of the *IHTSDO International Release* and belongs to the specified *Namespace*; or;
- 7 digits only - the file has been developed and released by a 3rd party, identified by the specified *Namespace*.

5.4.6 VersionDate element

5.4.6.1 Description

The VersionDate element of the filename identifies the *SNOMED CT* version with which the file is intended to be used. Its format is an 8-digit number in the pattern "YYYYMMDD", in compliance with the *ISO* 8601 standard.

5.4.6.2 Rules

The following rules apply to the content of this element:

- For Data Files("**sct**","**der**" or "**res**"),and for Documentation (" **doc**") with a *status* tag value of "**Current**", the value of this element should always be the same as the *SNOMED CT* version date with which the file is associated.
- For other file types, the VersionDate element will identify the (past) date of the *SNOMED CT release* for which the file was intended. A file distributed with a past version date has not been updated to reflect changes to *SNOMED CT* since that date, nor has it been validated as correct or appropriate for current use.

VersionDate refers to the official, published date of a *SNOMED CT International Release*, or of the *National Release* of an *IHTSDO Member* country, and may not always correspond to the actual date of distribution of any particular release.

5.4.7 Extension element

The *extension* element of the filename identifies the file format (encoding convention) of the file, such as "**txt** ", "**pdf**" or "**zip**". It has a format of 1-4 alphanumeric characters.

5.5 Release Format 2 - Core Component Guide

This guide describes *SNOMED CT Release Format 2* (*RF2*), to be used for official production releases of *SNOMED CT*. This format is not mandated for internal terminology development usage or as an interchange mechanism between terminology development systems.

The purpose of *RF2* is to provide a format that is flexible, unambiguous and useful. Its primary aim is to strengthen *SNOMED CT* by providing a format that is simple and stable, while enabling innovation through adaptations to cater for changing requirements.

This specification was developed by harmonizing proposals reviewed by the *IHTSDO* Enhanced *Release Format* Project Group, including:

- The “Enhanced *Release Format* Specification” (International Health Terminology Standards Development Organisation. *SNOMED Clinical Terms ® Enhanced Release Format Proposed Specification* , 21 June 2007).
- The “*Reference Set* Specification” (International Health Terminology Standards Development Organisation. *SNOMED Clinical Terms ® Reference Sets - Proposed Specification* , 31 July 2007).
- The “Alternate *Release Format*” proposed by NEHTA in coordination with their Australian Affiliates.

5.5.1 General

5.5.1.1 File Naming and Layout

In *RF2*, *release files* will be named predictably and in such a way as to avoid naming clashed between files in the *International release* and *National releases*.The basic pattern for *SNOMED CT release file* names consists of five elements, separated by an underscore ("_"), followed by a full stop (".") and a file *extension*:

Full details of the file naming convention can be found in the " *SNOMED CT* File Naming Convention" document(see associated documentation).All *release files*:

- are *UTF-8* encoded, tab delimited text files.
- contain a column header row, providing field names for each column within the file. Lower camel case is used for the field names (e.g. *moduleId*, *effectiveTime*).
- use DOS style line termination. Each line is terminated with a carriage return character followed by a line feed character.
- Should have a last line that ends with a line terminator (CR/LF) before the end of file.

5.5.1.2 Field Data Types

The following data types are used in the *release files*:

Table 34: Data Types Used in Release Files

Data Type	*Description*
SCTID	A *SNOMED CT identifier*, between 6 and 18 digits long, as described in *SCTID Representation*.
UUID	Universally Unique *Identifier*, 128-bit unsigned *integer*
Integer	32-bit signed *integer*.

Data Type	***Description***
String	*UTF-8* text of a specified length.
Boolean	*Boolean* value, represented as one of two possible *integer* values ('1' for true, '0' for false).
Time	For *release files*, a time format down to day of the year is used, having an *ISO* 8601 basic representation of *YYYYMMDD*. For development interchange formats, an ASCII text field in the *ISO* 8601 basic format *YYYYMMDDThhmmss* Z will be used. The time zone will always be UTC, as indicated by the trailing "Z". (e.g. 20080602T223000Z represents 10:30pm June 2 2008 UTC.)

5.5.1.3 Metadata and Enumerated Values

Concept enumerations are used across all *release files*. A *concept* enumeration simply uses *concepts* in a metadata *hierarchy* to represent an enumerated *value set* rather than using arbitrary *integer* values directly. A *concept* enumeration will therefore use an *SCTID* data type.

Non-clinical metadata is separated from the *SNOMED CT* clinical content by holding the two types of data in two separate hierarchies. The *concept* named |SNOMED CT Model Component|, which is a child of the *root concept* |SNOMED CT Concept|, contains the metadata model that supports each release.

Underneath the |SNOMED CT Model Component| *hierarchy*, the |core metadata concept| *sub-hierarchy* contains *concepts* that are referenced from fields within other *International Release* files (the *Concept*, *Description*, *Relationship*, *Identifier files*).

The |foundation metadata concept| *sub-hierarchy* also sits below the |SNOMED CT Model Component| *hierarchy*. This *sub-hierarchy* contains the metadata that supports the extensibility mechanism, and is discussed in more detail in the *Reference Sets Guide*.

The third and forth *sub-hierarchy* under |SNOMED CT Model Component| are the |linkage concept| *sub-hierarchy*, which holds details of *relationship types*, and the |namespace concept| *sub-hierarchy*, which holds details of *Namespaces*.

For more information, see *Concept Enumerations*.

5.5.1.4 Identification of Source Module

A *moduleId* field, assigned to each component, helps identify the origin of content and dependencies in a release. This enables release centers to compose a unified release from a number of different modules, yet still identify the origin of content within the release. For example, module ids may be used to differentiate *SNOMED CT* International content, Australian Medicines terminology and Pathology content within the Australian *National release*.

Each component within a *SNOMED CT release* references a *moduleId*. This is the module in which the component is currently maintained. A module is simply a collection of *SNOMED CT components* that are maintained as a unit by a single organization . It is the organization 's responsibility to organize the components in each *extension* that it is responsible for into one or more modules, in a way that best fits its business needs.

5.5.1.5 Meaning of the active field

Each *component* in *RF2* has an associated *active* field, which can take values of true ('1') or false ('0'). The meaning of this flag is described by *component* type in the following table:

Table 35: Behavior of Active and Inactive Components

***Component* Type**	***active* value**	***Description* of behavior when most recent row representing a *component* has the specified *active* value**
Concept	True	• The *Concept* is intended for *active* use. • All *active Descriptions* for which the conceptId refers to this *Concept* are valid. Visibility of these *active Descriptions* depends on information contained in applicable RefsetMembers (for example, whether the *Description* is in a *language dialect reference set* that is currently enabled in the vendor's system). • All *active Relationships* of which it is the *sourceId* or *destinationId* are applicable.
Concept	False	• The *Concept* is not intended for *active* use. However, it remains a valid *concept* for historical purposes as part of the *SNOMED CT* commitment to the principle of '*concept* permanence'. • Valid *Descriptions* of the *Concept* remain *active* allowing it to be appropriately viewed in human-readable form. • An *inactive Concept* cannot be the *sourceId*, *destinationId* or *typeId* of an *active Relationship*.
Description	True	• The *Description* contains a *Term* that is a valid *description* of the *Concept* referred to by the conceptId. • An *active Description* may refer to an *inactive Concept*, in which case the *Term* provides a valid *description* of that *inactive Concept*. Text based searches should (by default) include only *active Descriptions* that refer to *active Concepts*.
Description	False	• The *Description* is not a valid and the associated *Term* should no longer be regarded as being associated with the *Concept* referred to by conceptId.
Relationship	True	• The *Relationship* represents a valid association of the type specified by the *typeId*, between two *Concepts* referred to by the *sourceId* and *destinationId*; • An *inactive Concept* cannot be the *sourceId*, *destinationId* or *typeId* of an *active Relationship*.
Relationship	False	• The *Relationship* is not valid. An *inactive Relationship* should be ignored as it does not apply. • This does not necessarily mean that the association indicated by the *Relationship* does not apply. The *Relationship* may be *inactive* because it is redundant and inferable based on other *active Relationships*. • An *inactive Relationship* may refer to either *active* or *inactive components*.
RefsetMember	True	• The RefsetMember contains valid information applicable to the *component* referred to by the *referencedComponentId*. • The *component* referred to by the *referencedComponentId* may be *active* or *inactive*. An *active* RefsetMember cannot make an *inactive component active* but may provide related information that continues to be relevant (e.g. the reason for inactivation).

***Component* Type**	***active* value**	***Description* of behavior when most recent row representing a *component* has the specified *active* value**
RefsetMember	False	• The RefsetMember is not valid. An *inactive* RefsetMember should be ignored. The information it contains is not applicable to the *component* referred to by *referencedComponentId*.

5.5.1.6 History Mechanism

The *effectiveTime* and *active* fields in the *release file* enable the use of a "log style" append-only data model to track all changes to each *component*, providing full traceability. Once released, a row in any of these files will always remain unchanged. Historic data is supplied in the *RF2 release files*, dating back to the first release in *RF1* format in 2002.

In order to change the properties of a current *component* (and, therefore, to create a new version of it), a new row is added to the applicable file, containing the updated fields, with the *active* field set to true and the timestamp in the *effectiveTime* field indicating the nominal date on which the new version was released.

To inactivate a *component*, a new row is added, containing the same data as the final valid version of the *component*, but with the *active* field set to false and the timestamp in the *effectiveTime* field indicating the nominal date of the release in which the final version ceased being valid.

Where editorial policy does not allow a particular property of a *component* to be changed whilst keeping the same *Identifier*, the *component* as a whole is inactivated (as described above), and a new row added with a new id, the *effectiveTime* set to the nominal date of the release in which this version of the *component* became valid, and the *active* field set to true.

It is thus possible to see both the current values and any historical values of a *component* at any point in time.

Content will not be future dated with respect to the release that it appears in, although a release itself may be released a few days before its nominal release date. Where there is a business requirement for specifying a future activation date for some *components*, this may be modeled using *reference sets*.

The following example demonstrates how the *history mechanism* works on the *Concept file*, but the same rules apply equally well to the *Description*, *Relationship* and *Reference set* member files. In this example, the *descriptions* associated with the *moduleId* and *definitionStatusId* have been shown in place of their *SCTID* values.

A new *concept* (101291009) is added on the 1st July 2007:

Table 36: History Example - Concept Added

<table>
<tr><th>Id</th><th>effectiveTime</th><th>active</th><th>moduleId</th><th>definitionStatusId</th></tr>
<tr><td>101291009</td><td>20070701</td><td>1</td><td>| Module 1 |</td><td>900000000000074008 | Primitive |</td></tr>
</table>

In the next release (on 1 st January 2008), the *concept* is moved from |Module 1| to |Module 2|. Because the moduleIdfield is not immutable, the *concept* may be updated simply by adding a new record with the same Id.

Table 37: History Example - Module Change

Id	*effectiveTime*	*active*	*moduleId*	*definitionStatusId*
101291009	*20070701*	1	| Module 1 |	900000000000074008 | Primitive |
101291009	*20080101*	1	| Module 2 |	900000000000074008 | Primitive |

In the next release (on 1st July 2008), the *concept* is changed from being | *Primitive* | to being | *Fully defined* |.

Table 38: History Example - Definition Status Changed

Id	*effectiveTime*	*active*	*moduleId*	*definitionStatusId*
101291009	*20070701*	1	| Module 1 |	|900000000000074008 | Primitive |
101291009	*20080101*	1	| Module 2 |	900000000000074008 | Primitive |
101291009	*20080701*	1	| Module 2 |	900000000000073002 | Defined |

In the next release (on 1 st January 2009), the *concept* is deactivated:

Table 39: History Example - Concept Made Inactive

Id	*effectiveTime*	*active*	*moduleId*	*definitionStatusId*
101291009	*20070701*	1	| Module 1 |	900000000000074008 | Primitive |
101291009	*20080101*	1	| Module 2 |	900000000000074008 | Primitive |
101291009	*20080701*	1	| Module 2 |	900000000000073002 | Defined |
101291009	*20090101*	0	| Module 2 |	900000000000074008 | Primitive |

Notes:

1. At no stage in this process are previously written records ever amended. Once a record has been released in a *release file*, it will continue to be released in exactly the same form in future *release files*.
2. Changes are only recorded at the point of release in the *RF2 release files*. If a *component* record is changed a number of times between releases (during an edit and review process), only the most recently amended record will be appended to the *release file*, not individual records showing each separate edit to the released *component*.

3. In the last example, as well as inactivating the concept (active=0), the *definitionStatusId* is changed from 900000000000073002 | Defined | to 900000000000074008 | Primitive |. In practice this change is not essential since the value of data columns is ignored when a *component* is inactive. Although the change is unnecessary and insignificant, it typically occurs since all the relationships of an inactive concept must also be inactive, and as a result, from the perspective of the authoring environment the concept cannot be regarded as 900000000000073002 | Defined |.

5.5.1.7 Release Types

Given the *RF2*'s history tracking capability, it is possible to perform a number of different releases of content:

Table 40: SNOMED CT Release Types

Release Type	*Description*
Full	The files representing each type of component contain **every** version of every component ever released.
Snapshot	The files representing each type of component contain **one** version of every component released up to the time of the snapshot. The version of each component contained in a snapshot is the most recent version of that component at the time of the snapshot.
Delta	The files representing each type of component contain only component versions created since the previous release. Each component version in a *delta release* represents either a new component or a change to an existing component.

There are valid use cases for each type of *Release Type*. Each *International release* will incorporate all three of these *Release Types*, allowing users to choose the most appropriate format for their needs.

A *full release* will always be available from release centers . Optionally, other *Release Formats* may also be made available. Where out of cycles releases are made, these will follow the same format as standard cycle releases.

5.5.2 Relationships between files

The *relationships* between the records in the *core files* in the *RF2 Release Format* are depicted in the following diagram.

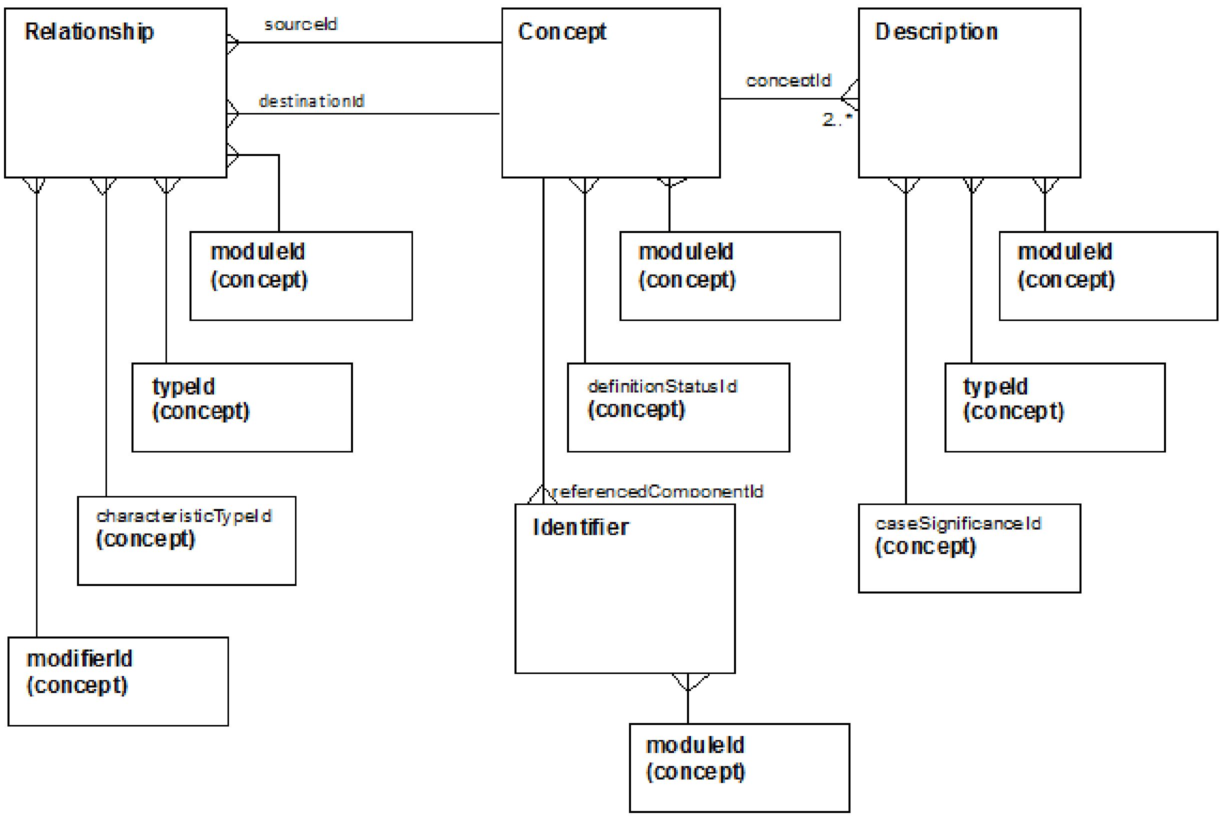


Figure 37: Relationships between files

Each *SNOMED CT concept* is held as a single row in the *Concept file*.Each row represents a clinical *concept*.

Each *concept* has two or more *descriptions* associated with it (at least one *synonym* and at least one *Fully Specified Name*). Each *description* is held as a single row in the *Description file*, and may only refer to a single *concept*.

Each *relationship*, from a source *concept* to a destination *concept*, is held as a single row in the *Relationship file*. The type of each *relationship* is defined by reference to a linkage *concept*, also held within the *Concept file*.

The most basic form of *relationship* is the subsumption *relationship*, identifying that one *concept* is a kind of another *concept*. For example, an |Outpatient procedure| |Is a| |Procedure|. All the *concepts* in *SNOMED CT* form an |Is a| *hierarchy*, with a parent *concept* connected to each *child concept* by an |Is a| *relationship*. In this *hierarchy*, a *child concept* may have more than one parent *concept*. The root of the *hierarchy* is the | *SNOMED CT Concept* |, which has 19 top level *children*, each forming its own *sub-hierarchy*. There are no |Is a| *relationships* that cross from one of these sub-hierarchies to another (e.g. from a *concept* in the Procedures *sub-hierarchy* to a *concept* in the Substances *hierarchy*.

In addition to the | is a | *relationships*, other *relationship types* are also held within the *Relationship file*, such as |Finding site| or |Laterality|. *Relationships* types are specified under the |Linkage| *sub-hierarchy* in the | *SNOMED CT* Model *component* | *hierarchy*.

5.5.3 File formats

The following sections provide details of the format of the *release files*. An SQL schema, which represents the content of each of theses files as a relational table, is provided as part of the *Terminology Service Guide*.

5.5.3.1 Concept file

The *Concept file* holds the clinical *concepts* that make up *SNOMED CT*. A *concept* is given meaning by its *Fully Specified Name*, which is held in the *Description file*. A *concept* may be distinguished from or refined by association with other *concepts* using *relationships*, which are held in the *Relationship file*.

Table 41: Concept file - Detailed Specification

Field	Data type	Immutable	Purpose
id	*SCTID*	Y	Uniquely identifies the *concept*.
effectiveTime	*Time*	N	Specifies the inclusive date at which the component version's state became the then current valid state of the component
active	*Boolean*	N	Specifies whether the *concept*'s state was *active* or *inactive* from the nominal release date specified by the *effectiveTime*
moduleId	*SCTID*	N	Identifies the *concept* version's module. Set to a *descendant* of 900000000000443000 | Module | within the metadata *hierarchy*.
definitionStatusId	*SCTID*	N	Specifies if the *concept* version is *primitive* or *fully defined*. Set to a *descendant* of 900000000000444006 | Definition status | in the metadata *hierarchy*.

Only one *concept* record with the same id field is current at any point in time. The current record will be the one with the most recent *effectiveTime* before or equal to the date under consideration. If the *active* field of this record is false ('0'), then the *concept* is *inactive* at that point in time.

When a *concept* is made *inactive*, the following operations take place:

- A new row is added to the *Concepts* file for the *concept*, with the *active* flag set to *inactive* (as described in the section on the *History Mechanism*);
- All *relationships* that have as source the *concept* to be inactivated will themselves be inactivated by adding a new row to the *Relationship file* for each *relationship*, with the *active* flag set to *inactive*;
- All *active descriptions* associated with the *concept* will remain unchanged unless incorrect for the *concept*;
- Rows will be added as needed to the *Historical Association Reference Set*, to model associations from the *inactive concept* to other *concepts*;
- *Active descriptions* that are still associated with the *inactive concept* will be added to the | *Description* inactivation indicator *reference set* |, with an associated value of | *Concept* non-current|.

5.5.3.2 Description file

The *Description file* holds *descriptions* that describe *SNOMED CT concepts*. A *description* is used to give meaning to a *concept* and provide well-understood and standard ways of referring to a *concept*.

Table 42: Description file - Detailed Specification

Field	Data type	Immutable	Purpose
id	*SCTID*	Y	Uniquely identifies the *description*.
effectiveTime	*Time*	N	Specifies the inclusive date at which the component version's state became the then current valid state of the component
active	*Boolean*	N	Specifies whether the *description*'s state was *active* or *inactive* from the nominal release date specified by the *effectiveTime*.
moduleId	*SCTID*	N	Identifies the *description* version's module. Set to a *child* of 900000000000443000 | Module | within the metadata *hierarchy*.
conceptId	*SCTID*	Y	Identifies the *concept* to which this *description* belongs. Set to an *Identifier* of a *concept* in the 138875005 | SNOMED CT Concept | *hierarchy* within the *Concept file*. Note that versions of *descriptions* and *concepts* don't belong to each other. Which version of any given *description* is combined with which version of its owning *concept* depends on the point in time at which they are accessed.
languageCode	*String*	Y	Specifies the *language* of the *description* text using the two character *ISO* -639-1 code. Note that this specifies a *language* level only, not a *dialect* or country code.
typeId	*SCTID*	Y	Identifies whether the *description* is an FSN, *Synonym* or other *description* type. This field is set to a *child* of 900000000000446008 | Description type | in the Metadata *hierarchy*.
term	*String*	N	The *description* version's text value, represented in *UTF-8* encoding.

Field	Data type	Immutable	Purpose
caseSignificanceId	*SCTID*	N	Identifies the *concept* enumeration value that represents the case significance of this *description* version. For example, the *term* may be completely case sensitive, case insensitive or initial letter case insensitive. This field will be set to a *child* of 900000000000447004 | Case significance | within the metadata *hierarchy*.

Only one *description* record with the same id field will be current at any point in time. The current record will be the one with the most recent *effectiveTime* before or equal to the point in time under consideration.

If the *active* field of this record is false ('0'), then the *description* is *inactive* at that point in time. If the *active* field is true ('1'), then the *description* is associated with the *concept* identified by the conceptId field.

The conceptId field, the languageCode field and the *typeId* field will not change between two rows with the same id, in other words they are immutable. Where a change is required to one of these fields, then the current row will be de-activated (by appending a row with the same id and the *active* field set to false) and a new row with a new id will be appended. Only limited changes may be made to the ' *term*' field, as defined by editorial rules.

Each *concept* will have at least one *active description* with a *typeId* of | *Synonym* | for a given languageCode (like "en"). Where a *concept* only has one *active description* with a *typeId* of | *Fully Specified Name* | for a given *language* code, then that *Description* can be taken as the *Fully Specified Name* for that *language* and each of its *dialects*, and need not therefore be explicitly included in *language reference sets* for that *language*. Where a *concept* only has one *active description* with a *typeId* of | *Fully Specified Name* | across all *language* codes within a release, then that *Description* can be taken as the *Fully Specified Name* for all *languages* and *dialects*, and need not therefore be explicitly included in any *language reference sets* in that release.

The *Term* field will be restricted as follows:

- to an overall maximum length of 32Kb;
- to a maximum length, configurable for each *description* type (as defined by the *Description* Type *reference set* member associated with that *description* type - see the "*SNOMED CT Release Format 2 - Reference Set* Specifications" document for more details);
- The format of the *term* field (plain text, limited HTML, XHTML, *DITA*) will also be configurable for each *description* type, using the same mechanism as above;
- Control characters (including TABs, CRs and LFs) will not appear in |Plain text| and |Limited HTML| format types.

5.5.3.3 Relationship file

The *Relationship file* holds one *relationship* per row. Each *relationship* is of a particular type, and has a source *concept* and a destination *concept*. An example of a *relationship* is given below:

| Outpatient procedure | | Is a | | Procedure | where:

- | Outpatient procedure | is the source *concept*;
- | Is a | is the *relationship type concept*; and;
- | Procedure | is the destination *concept*.

Table 43: Relationship file - Detailed Specification

Field	Data type	Immutable	Purpose
id	*SCTID*	Y	Uniquely identifies the *relationship*.
effectiveTime	*Time*	N	Specifies the inclusive date at which the component version's state became the then current valid state of the component
active	*Boolean*	N	Specifies whether the *relationship*'s state was *active* or *inactive* from the nominal release date specified by the *effectiveTime* field.
moduleId	*SCTID*	N	Identifies the *relationship* version's module. Set to a *child* of 900000000000443000 | Module | within the metadata *hierarchy*.
sourceId	*SCTID*	Y	Identifies the source *concept* of the *relationship* version, i.e., the *concept* the *relationship* version emanates from. Set to an *Identifier* of a *concept* in the *Concept* file.
destinationId	*SCTID*	Y	Identifies the *concept* that is the destination of the *relationship* version. Set to an *Identifier* of a *concept* in the *Concept* file.
relationshipGroup	*Integer*	N	Groups together *relationship* versions that are part of a logically associated *relationship group*. All *active Relationship* records with the same *relationshipGroup* number and *sourceId* are grouped in this way.
typeId	*SCTID*	Y	A *concept* enumeration value from the metadata *hierarchy* that identifies the semantic type of the *relationship* version. 116680003 | Is a | or a subtype of 410662002 | Concept model attribute |. For example | Is a |, or | associated morphology |.

Field	**Data type**	**Immutable**	**Purpose**
characteristicTypeId	*SCTID*	N	A *concept* enumeration value that identifies the characteristic type of the *relationship* version (i.e. whether the *relationship* version is defining, qualifying, etc.) This field is set to a *descendant* of 900000000000449001 | Characteristic type | in the metadata *hierarchy*.
modifierId	*SCTID*	N	A *concept* enumeration value that identifies the type of *Description Logic* (DL) restriction (some, all, etc.). Set to a *child* of 900000000000450001 | Modifier | in the metadata *hierarchy*.

Only one *relationship* record with the same id field will be current at any point in time. The current record will be the one with the most recent effectiveTimebefore or equal to the point in time under consideration.

If the *active* field of this record is false ('0'), then the *relationship* is *inactive* at that point in time. If the *active* field is true ('1'), then there is a *relationship* between the *SNOMED CT concepts* identified by *sourceId* and *destinationId*.

The *sourceId*, *destinationId*, *relationshipGroup*, *typeId*, characteristicTypeIdand *modifierId* will not change between two rows with the same id, in other words they are immutable. Where a change is required to one of these fields, then the current row will be de-activated (by appending a row with the same id and the *active* field set to false) and a new row with a new id will be appended.

The *relationshipGroup* field is used to group *relationships* with the same *sourceId* field into one or more logical sets. A *relationship* with a *relationshipGroup* field value of '0' is considered not to be grouped. All *relationships* with the same *sourceId* and non-zero *relationshipGroup* are considered to be logically grouped.

The *relationshipGroup* field will be an unsigned *integer*, and will not be limited to a single digit value. There is no guarantee that they will be assigned sequentially, and the values will not be unique across *concepts*.

The *modifierId* field will initially be set to 900000000000451002 | Some | to keep compatibility with the *RF1* release. Widening the range of this field to include other values (such as |All|) will in future increase the expressive power of *SNOMED CT*. However, this is likely to come at the cost of an increase in reasoning complexity, leading to potential issues for classification tooling.

Notes:

1. The modiferId field has been included at this stage as the *RF2* format is likely to be stable for at least a five year period, without addition or deletion of fields. Within that period it is anticipated that other *modifierId* values will be added. Therefore, although not fully implemented at this stage, this field has been included in the initial RF2 specification as it represents an integral part of the *Description Logic* used by *SNOMED CT*.
2. Any expansion of *SNOMED CT* to include *relationships* with a *modifierId* set to a value other than 900000000000451002 | Some | will be discussed with *Members* first and approved by the Technical Committee.
3. Changes have been made to the "Immutability" values shown in the above table in the 2014-07-31 version. These changes reflect the fact that the values in the following columns of a uniquely identified relationship have occurred in historical data and in these cases tracking the history of these changes is of greater value that insisting on immutability.

- relationshipGroup: The number can change though the logical content of the group represented should not change. Additionally no significance should be read into the relationshipGroup value of an inactive *relationship*;
- characteristicType: This has changed in historical data but should not change in future;
- modifierId: Since there is currently only one value for this no changes are possible but if the permitted values are extended as suggested above then it is likely that changes would be required.

5.5.3.4 Identifier file

This file provides a standardized way of associating alternate *Identifiers* from various schemes with *SNOMED CT components*.

At any point in time, an alternate *Identifier* within a particular scheme will be associated with one and only one *SNOMED CT component*. A *SNOMED CT component* may be associated with zero or more alternate *Identifiers* within a single scheme.

It is important to note that the *SNOMED CT component* and its alternate *Identifiers* all identify precisely the same real-world object.

Note: The Identifier file is not currently used in the *SNOMED CT International Release* as use of the more flexible *Simple map type references set* structure is preferred for links to alternative codes. The only known current use of this file is for internal identification of components during the content development process.

Table 44: Identifier file - Detailed Specification

Field	Data type	Immutable	Purpose
identifierSchemeId	*SCTID*	Y	*Identifier* of the *concept* enumeration value from the Metadata *hierarchy* that represents the scheme to which the *Identifier* value belongs. Set to a *descendant* of 900000000000453004 | Identifier scheme | within the metadata *hierarchy*.
alternateIdentifier	*String*	Y	*String* representation of the alternate *Identifier* in its native scheme.
effectiveTime	*Time*	N	Specifies the inclusive date at which the alternate *Identifier* was associated with the *SNOMED CT component*.
active	*Boolean*	N	Specifies whether the association was *active* or *inactive* from the point in time specified by the *effectiveTime*.
moduleId	*SCTID*	N	Identifies the source module that this association was created in. Set to a *child* of 900000000000443000 | Module | within the metadata *hierarchy*.

Field	Data type	Immutable	Purpose
referencedComponentId	*SCTID*	Y	Uniquely identifies the *SNOMED CT component* with which the alternate *Identifier* is associated.

Only one record with the same *identifierSchemeId* and *alternateIdentifier* fields will be current at any point in time. The current record will be the one with the most recent effectiveTimebefore or equal to the point in time under consideration.

If the *active* field of this record is false ('0'), then the association is *inactive* at that point in time. If the *active* field is true ('1'), then there is an identity at that point in time between the *referencedComponentId* (a *SNOMED CT component*) and the alternateIdentifier in the scheme identified by *identifierSchemeId*.

5.5.3.5 Transitive Closure History File

The *Transitive Closure* is the complete set of *relationships* between every *concept* and each of its super-type *concepts*, in other words both its parents and *ancestors*. A *Transitive Closure* History file can be generated from the *SNOMED CT* content using scripts provided with each release. The generated file will be of the following format and contain the valid states of the *transitive closure* of each *concept* across all previous releases:

Table 45: Transitive Closure History File - Detailed Specification

Field	Data type	Purpose
subtypeId	*SCTID*	Id of the *concept* playing the *subtype* role. Set to an *Identifier* of a *concept*.
supertypeId	*SCTID*	Id of the *concept* playing the supertype role. Set to an *Identifier* of a *concept*.
effectiveTime	*Time*	Specifies the inclusive date at which the *transitive closure* record became valid.
active	*Boolean*	Specifies whether the *Identifier* version's state was *active* or *inactive* from the point in time specified by the *effectiveTime*.

5.5.4 Extensibility Mechanism

Reference set data structures provide the foundation pieces for *RF2*'s generic extensibility mechanism. These building blocks provide a common foundation for *extension* owners to build on *SNOMED CT*. They also enable the *Release Format* to support changing requirements.

Conventions applied to the *RF2* files such as field names, field *order* and history tracking have also been applied to the *reference set* specification. This has been done to provide consistency across all components in the *Release Format*.

5.5.4.1 The basic reference set member file format

The basic *reference set* data structure consists of the following fields:

Table 46: Basic Reference Set Data Structure

Field	Data type	Immutable	Purpose
id	*UUID*	Y	A 128 bit unsigned *integer*, uniquely identifying the *reference set* member.
effectiveTime	*Time*	N	Specifies the inclusive date at which this change becomes effective.
active	*Boolean*	N	Specifies whether the member's state was *active* or *inactive* from the nominal release date specified by the *effectiveTime* field.
moduleId	*SCTID*	N	Identifies the member version's module. Set to a *child* of 900000000000443000 | Module | within the metadata *hierarchy*.
refsetId	*SCTID*	Y	Uniquely identifies the *reference set* that this *extension* row is part of. Set to a *descendant* of 900000000000455006 | Reference set | within the metadata *hierarchy*.
referencedComponentId	*SCTID* or *UUID*	Y	Uniquely identifies the component that this row relates to, thus defining membership of this component in the *Reference Set*. This field can be set to the *Identifier* of a record within the *Concept*, *Description*, *Relationship* or *Reference Set* member file. However, the content of this field can be further restricted for each *reference set* by the *reference set* descriptor (see the "*SNOMED CT Release Format 2 - Reference Set* Specifications" document for more details).
Zero or more other fields	*SCTID*, *String*, or *Integer*	N	Optional field
...	*SCTID*, *String*, or *Integer*	N	Optional field

Each *reference set* will be defined as a *concept* in the metadata *hierarchy*.

There will be one *active* row in the above table for each member of the *reference set*. Individual *reference set* members will be uniquely identified using a *UUID*. Each *Reference Set* member will belong to a single *Reference Set* (referred to by the refsetId field) and will also reference the member component that belongs to that *reference set* (using the *referencedComponentId* field). The member component may be a *Concept*, *Description*, *Relationship* or a *RefSet* member itself.

Only one *reference set* member record with the same id field will be current at any point in time. The current record will be the one with the most recent effectiveTimebefore or equal to the point in time under consideration.

If the *active* field of this record is false ('0'), then the *reference set* member is *inactive* at that point in time. If the *active* field is true ('1'), then the component referenced by the *referencedComponentId* field is deemed to be a member of the *reference set* identified by the refsetId field.

The *refsetId* and *referencedComponentId* fields will not change between two rows with the same id, in other words they are immutable. Where a change is required to one of these fields, then the current row will be de-activated (by appending a row with the same id and the *active* field set to false) and a new row with a new id will be appended.

A component may belong to any number of *reference sets* and to each *reference set* more than once. In the latter case, there will be more than one row with the same refsetId and *referencedComponentId*, each having different id fields, so co-existing at the same time.

5.5.4.2 Extending the basic reference set member file format

The *reference set* member file structure may be extended by addition of one or more fields.
Each of these fields will hold additional values specific to each member. Data types that are supported in the additional columns are:

- *Integer*
- *String*
- *Component* (a reference to a *SNOMED CT component*).

Finer grained interpretation of the values is based on the 900000000000456007 I Reference set descriptor I. Further details can be found in the *Reference set specifications*.

The different *Reference Set* patterns that are supported will depend on a documented set of use cases. The supported patterns will expand over time as further use cases are identified.

5.5.5 Metadata hierarchy

As the *release file* formats contain a number of *concept* enumerations, it is necessary to define sets of *concepts* that represent the allowed values. As well as the enumerated values, other metadata supporting the extensibility mechanism and the *concept model* is required.

The *concept* I SNOMED CT Model Component (metadata) I is a *subtype* of the *root concept* (I SNOMED CT Concept I), and contains the metadata, supporting the release.

The *subtypes* of I SNOMED CT Model Component (metadata) I are described in *Table 47* and the top three levels of the hierarchy are shown in *Figure 38*.

Table 47: SNOMED CT Model Component (metadata) (900000000000441003)

Id	Term	Comment
106237007	Linkage concept	*Concepts* that specify • Semantic *Relationships* between *concepts* (I Attribute I); and • Asserted associations between statements in a record (I Link assertion I)

Id	Term	Comment
370136006	Namespace concept	*Concepts* that specify the *Extension* Namespaces allocated by the *IHTSDO*.
900000000000442005	Core metadata concept	*Concepts* that are referenced from enumerated fields within the *International Release* files (the *Concept*, *Description*, *Relationship*, *Identifier files*).
900000000000454005	Foundation metadata concept	The metadata that supports the extensibility mechanism, and is discussed in more detail in the *Reference Sets Guide*.

- 138875005 | SNOMED CT Concept |
    - *... (content hierarchies) ...*
    - 900000000000441003 | SNOMED CT Model Component |
        - 106237007 | linkage concept |
            - 246061005 | attribute | ...
            - 416698001 | link assertion | ...
        - 370136006 | namespace concept | ...
        - 900000000000442005 | core metadata concept |
            - 900000000000443000 | module |
                - 900000000000445007 | IHTSDO maintained module | ...
            - 900000000000444006 | definition status |
                - 900000000000073002 | defined |
                - 900000000000074008 | primitive |
            - 900000000000446008 | description type |
                - 900000000000003001 | fully specified name |
                - 900000000000013009 | synonym |
                - 900000000000550004 | definition |
            - 900000000000447004 | case significance |
                - 900000000000017005 | case sensitive |
                - 900000000000020002 | only initial character case insensitive |
                - 900000000000448009 | case insensitive |
            - 900000000000449001 | characteristic type |
                - 900000000000006009 | defining relationship | ...
                - 900000000000225001 | qualifying relationship |
                - 900000000000227009 | additional relationship |
            - 900000000000450001 | modifier |
                - 900000000000451002 | some |
                - 900000000000452009 | all |

    - 900000000000453004 | identifier scheme |
        - 900000000000002006 | SNOMED CT UUID |
        - 900000000000294009 | SNOMED CT integer identifier |
- 900000000000454005 | foundation metadata concept |
    - 900000000000455006 | reference set |
        - 900000000000456007 | reference set descriptor |
        - 900000000000480006 | attribute value type | ...
        - 900000000000496009 | simple map | ...
        - 900000000000506000 | language type | ...
        - 900000000000512005 | query specification type | ...
        - 900000000000516008 | annotation type | ...
        - 900000000000521006 | association type | ...
        - 900000000000534007 | module dependency |
        - 900000000000538005 | description format |
    - 900000000000457003 | reference set attribute |
        - 900000000000458008 | attribute description |
        - 900000000000459000 | attribute type | ...
        - 900000000000479008 | attribute order |
        - 900000000000491004 | attribute value | ...
        - 900000000000499002 | scheme value |
        - 900000000000500006 | map source concept |
        - 900000000000501005 | map group |
        - 900000000000502003 | map priority |
        - 900000000000503008 | map rule |
        - 900000000000504002 | map advice |
        - 900000000000505001 | map target |
        - 900000000000510002 | description in dialect |
        - 900000000000511003 | acceptability | ...
        - 900000000000514006 | generated reference set |
        - 900000000000515007 | query |
        - 900000000000518009 | annotated component |
        - 900000000000519001 | annotation | ...
        - 900000000000532006 | association source component |
        - 900000000000533001 | association target component |
        - 900000000000535008 | dependency target |
        - 900000000000536009 | source effective time |
        - 900000000000537000 | target effective time |
        - 900000000000539002 | description format | ...
        - 900000000000544009 | description length |

Figure 38: SNOMED CT Metadata Hierarchy (2010-07-31)

5.6 Release Format 2- Reference Sets Guide

5.6.1 Introduction

This guide describes the *reference set* specifications released as part of the *SNOMED CT Release Format 2*. This format is not mandated for internal terminology development usage or as an interchange mechanism between terminology development systems.

The purpose of *RF2* is to provide a format that is flexible, unambiguous and useful. Its primary aim is to strengthen *SNOMED CT* by providing a format that is simple and stable, while enabling innovation through adaptations to cater for changing requirements.

This format specification was developed by harmonizing proposals reviewed by *IHTSDO* Enhanced *Release Format* Project Group, including:

- Enhanced *Release Format* Specification (International Health Terminology Standards Development OrganizationOrganisation. *SNOMED Clinical Terms ® Enhanced Release Format Proposed Specification* , 21 June 2007).
- *Reference Set* Specification (International Health Terminology Standards Development OrganizationOrganisation. *SNOMED Clinical Terms ® Reference Sets - Proposed Specification* , 31 July 2007).
- Alternate *Release Format* proposed by NEHTA in coordination with their Australian Affiliates.

5.6.1.1 Associated Quality Measures

Although the definition of quality measures to monitor the implementation of this standard do not fall under the scope of this guide, they will be covered by the documentation covering the QA and *Release* process for the *Workbench*.

5.6.1.2 Separation of Reference Sets into Release files

Separation of *reference sets* into files may be done in a number of ways. Each *reference set* will have a particular structure for the optional fields that are appended to each member. For example, a simple *reference set* will have no additional fields; a CSI *reference set* will have three additional fields - the first a *component*, the second a *String*, and the third an *Integer*. There must be at least one *reference set* member file for each different *reference set* structure, as defined above. *Reference sets* may be further split, if required, by the owner of the *reference sets*. The naming conventions for the *reference set* files provide more detail on the naming convention to be used in this case (see the " *SNOMED CT* File Naming Convention" document).

Each *reference set* file will have a header row containing field names for each of the columns. The names of the standard fields will be the field names as detailed in the "*SNOMED CT Release Format 2* - Data Structures Specification" document.

In *release files* the additional fields will be named in accordance with the relevant row in "*Reference Set* Descriptor *reference set*".

Note: When imported by an application the *release file* names of additional attributes may be substituted by a more generic name to allow *reference sets* with a similar pattern of additional fields to be represented in a single table. In this case the "*Reference Set* Descriptor *reference set*" provides a link to the original name of the field in the distribution file.

5.6.2 Reference Set Specifications

This section first details how *reference sets* themselves are described in a machine readable form, using a set of | *Reference set* descriptor| member records (called a Descriptor, for short). It then describes a number of standard *reference set* patterns. Each of these patterns is also described in a machine readable form using a set of | *Reference set* descriptor| member records (called a Descriptor Template, for short). Each pattern may be used to define a number of *reference sets*. At the end of the section, a number of individual *reference sets* are described that do not conform to a particular pattern.

In each subsection, each *reference set* or *reference set* pattern is described in turn:

- The purpose of each *reference set* is first described;
- The format of the *reference set* member record is detailed in a table;
- The metadata supporting the *reference set* is described;
- The machine readable *reference set* descriptor member records for the *reference set* pattern (the Descriptor Template, for short) are then shown;
- Examples of usage are given, providing example Descriptors, where appropriate.

The first *reference set* to be described is the *reference set* descriptor. Subsequent sections describe a number of *reference set* patterns.

5.6.2.1 Overview

5.6.2.1.1 Descriptors, Descriptor Templates and Patterns

The purpose of the | *Reference set* descriptor| is to describe the format of all other *reference sets* that may be included in a release. A Descriptor held within the | *Reference set* Descriptor| describes the referencedComponentIdfield and the additional fields for the *reference sets* it describes. Each field is described using a *concept* in the metadata. The type of each field is also described in the same way.

Patterns allow a number of different types of *reference set* to be defined, each of which will conform to the specified pattern, having the same *release file* format. The file format of each *reference set* pattern is described by a Descriptor Template. This Descriptor Template describes the format and number of additional fields held against members of *reference sets* conforming to the pattern, and provides an envelope within which those additional fields may be further refined for each *reference set* conforming to the pattern. The Descriptor Template for each pattern is provided in the section describing that pattern.

Each defined *reference set* that conforms to a pattern will have its own Descriptor, that describes its own specific properties, and although *reference set* field types must still conform to the Descriptor Template for the pattern, each field type may be further constrained using data sub-types specified in the metadata *hierarchy*. This provides some level of *refinement* to the *constraints* that may be applied to a *reference set* conforming to a particular pattern.

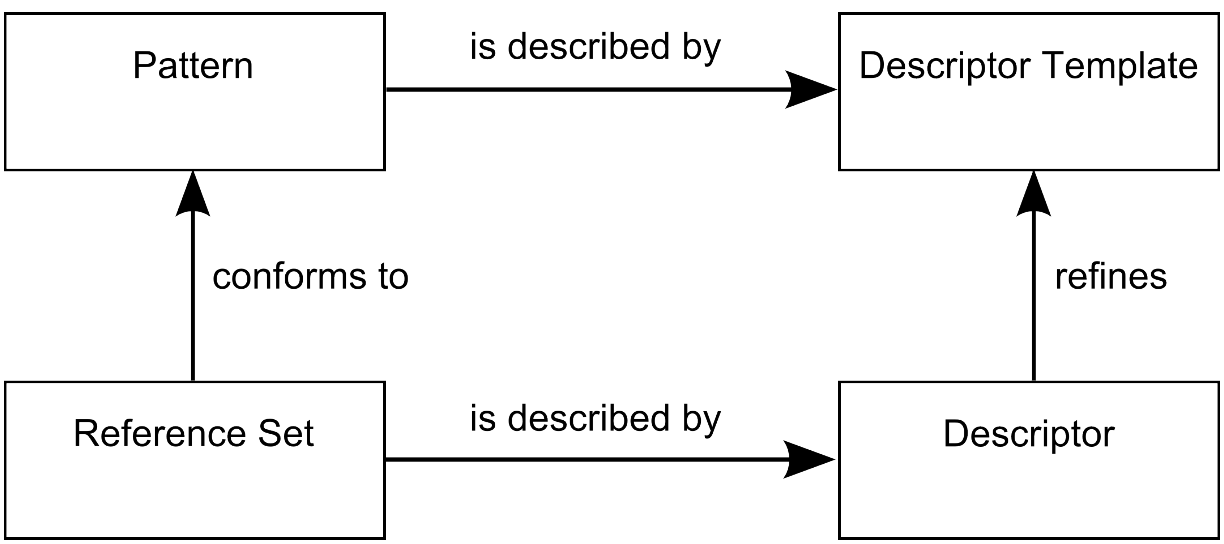


Figure 39: Graphical view of Relationships between patterns, reference sets, Descriptor Templates and Descriptors

5.6.2.1.2 Patterns and Use Cases

The next table summarizes the use cases for *reference sets* (one per row) that are described in the following sections, and shows which *reference set* patterns are used in each case:

Table 48: Reference Set Use Case Summary

	Patterns								
Use cases	**I Attribute value typeI (C)**	**I Simple map I (S)**	**I Complex map I (IISSSC)**	**I Language I (C)**	**I Query Specification I(CCS)**	**I Annotation I (S)**	**I Association I (C)**	**I Ordered I (IC)**	**I Simple I**
Refinability of *relationships*	*								
ICD-10 mapping	*		*						
Inactivation indicator	*								
CVT3 map		*							
SNOMED RT map		*							
Language dialect				*					
Language dialect with context				*					
Intension *reference set* specification					*				
Image *annotation*						*			
Short *annotation*						*			
Descriptive *annotation*						*			
reason for inactivation							*		

	Patterns								
Use cases	**I Attribute value typeI (C)**	**I Simple map I (S)**	**I Complex map I (IISSSC)**	**I Language I (C)**	**I Query Specification I(CCS)**	**I Annotation I (S)**	**I Association I (C)**	**I Ordered I (IC)**	**I Simple I**
RF1 Subset representation				*				*	

Note: The letters shown after each pattern indicate the type and number of additional fields held against each member of a *reference set* conforming to that pattern, where 'C' is short for *component*, 'S' is short for *String* and 'I' is short for *Integer*.

Example: *Reference sets* conforming to the | *Attribute value* type| (C) pattern will have one additional field held against each member, a component reference; *reference sets* conforming to the |Simple type| pattern will have no additional fields held against each member.

5.6.2.1.3 Metadata Supporting Reference Sets

Reference sets are described by *concepts* under the | *Reference set* | *sub-hierarchy*.

- 900000000000455006 | reference set |
    - 900000000000456007 | reference set descriptor |
    - 900000000000480006 | attribute value type | ...
    - 900000000000496009 | simple map | ...
    - 900000000000506000 | language type | ...
    - 900000000000512005 | query specification type | ...
    - 900000000000516008 | annotation type | ...
    - 900000000000521006 | association type | ...
    - 900000000000534007 | module dependency |
    - 900000000000538005 | description format |

Figure 40: Reference Sets in the Metadata Hierarchy

Values that can be used within *reference set* fields are described in the | *Reference set* attribute | *sub-hierarchy*.

- 900000000000457003 | reference set attribute |
    - 900000000000458008 | attribute description |
    - 900000000000459000 | attribute type | ...
    - 900000000000479008 | attribute order |
    - 900000000000491004 | attribute value | ...
    - 900000000000499002 | scheme value |
    - 900000000000500006 | map source concept |
    - 900000000000501005 | map group |
    - 900000000000502003 | map priority |
    - 900000000000503008 | map rule |
    - 900000000000504002 | map advice |
    - 900000000000505001 | map target |
    - 900000000000510002 | description in dialect |
    - 900000000000511003 | acceptability | ...
    - 900000000000514006 | generated reference set |
    - 900000000000515007 | query |
    - 900000000000518009 | annotated component |
    - 900000000000519001 | annotation | ...
    - 900000000000532006 | association source component |
    - 900000000000533001 | association target component |
    - 900000000000535008 | dependency target |
    - 900000000000536009 | source effective time |
    - 900000000000537000 | target effective time |
    - 900000000000539002 | description format | ...

- 900000000000544009 I description length I

Figure 41: Reference Set Attributes Metadata Hierarchy

The way that each of the *concepts* shown in this metadata *hierarchy* is used is described in each of the following sections.

5.6.2.1.4 Naming Conventions for Reference Sets

National Release Centres and others may create additional *reference sets*. A *namespace* is required to create a new *reference set*, as each *reference set* is defined by a *Concept*. The *Concept*'s FSN and a *Synonym* are used to name the *reference set*. Where a new *reference set* is created against an existing pattern, then the following naming convention should be used (where the text "My particular" should be replaced by the name of the *reference set*):

***Attribute value* type *reference set* (pattern)**

FSN = My particular *attribute value reference set* (foundation metadata *concept*)

PT = My particular *reference set*

Simple Map type *reference set* (pattern)

FSN = My particular simple map *reference set* (foundation metadata *concept*)

PT = My particular simple map

Complex Map type *reference set* (pattern)

FSN = My particular complex map *reference set* (foundation metadata *concept*)

PT = My particular complex map

Language* type *reference set* (pattern) - for a *Language refset

FSN = English - *ISO* 639-1 code 'en'*language reference set* (foundation metadata *concept*)

PT = English

Language* type *reference set* (pattern) - for a *Dialect RefSet

FSN = GB English *language reference set* (foundation metadata *concept*)

PT = GB English

***Query* specification type *reference set* (pattern)**

FSN = My particular *query* specification *reference set* (foundation metadata *concept*)

PT = My particular *query* specification *reference set*

***Annotation* type *reference set* (pattern)**

FSN = My particular *annotation reference set* (foundation metadata *concept*)

PT = My particular *annotation reference set*

Association type *reference set* (pattern)

FSN = My particular association *reference set* (foundation metadata *concept*)

PT = My particular association *reference set*

5.6.2.2 Reference Set Descriptor

5.6.2.2.1 Purpose

The 900000000000456007 I Reference set descriptor I is a *reference set* that used to specify the format of all *reference sets* included in a release. The data type and meaning of the referenced component and each additional field within each *reference set* is described by this *reference set*.

Reference set descriptor can be used to define

- The order of appearance of additional attributes (other than those mandatory for all *reference sets*);
- The name and purpose of the additional attributes;
- The data types for the additional attributes.

This allows for a *reference set* to be validated using the metadata embedded within the *reference set* descriptor in the following ways:

- the data type of its attributes may be validated against the data type declared in the *reference set* descriptor;
- the column order can be checked against the *reference set* descriptor.

5.6.2.2.2 Data structure

The *Reference set descriptor reference set* is a *Component* - *Component* - *Integer reference set* that specifies the structure of reference sets. Its structure is shown in the following table.

Table 49: Reference set descriptor reference set - Data structure

Field	Data type	Purpose
id	*UUID*	A 128 bit unsigned *integer*, uniquely identifying this *reference set member*. Different versions of a *reference set member* share the same *id* but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.
effectiveTime	*Time*	The inclusive date or time at which this version of the identified *reference set member* became the current version. The current version of this *reference set member* at time *T* is the version with the most recent *effectiveTime* prior to or equal to time *T*.
active	*Boolean*	The state of the identified *reference set member* as at the specified *effectiveTime*. If *active*= 1 (true) the *reference set member* is part of the current version of the set, if *active*= 0 (false) the *reference set member* is not part of the current version of the set.
moduleId	*SCTID*	Identifies the *SNOMED CT module* that contains this *reference set member* as at the specified *effectiveTime*. The value must be a *subtype* of 900000000000443000 | Module (core metadata concept) | within the metadata *hierarchy*.

Field	Data type	Purpose
refsetId	*SCTID*	Identifies the *reference set* to which this *reference set member* belongs. In this case, set to 900000000000456007 | Reference set descriptor |
referencedComponentId	*SCTID*	Identifies the *reference set* (or type of *reference set*) that is specified by this descriptor. Set to a *descendant* of 900000000000455006 | reference set (foundation metadata concept) | in the metadata *hierarchy*.
attributeDescription	*SCTID*	Specifies the name of an attribute that is used in the *reference set* to which this descriptor applies. Set to a *descendant* of 900000000000457003 | Reference set attribute (foundation metadata concept) | in the metadata *hierarchy*, that describes the additional attribute extending the *reference set*.
attributeType	*SCTID*	Specifies the data type of this attribute in the *reference set* to which this descriptor applies. Set to a *descendant* of 900000000000459000 | attribute type (foundation metadata concept) | in the metadata *hierarchy*, that describes the type of the additional attribute extending the *reference set*.
attributeOrder	*Integer*	Specifies the position of this attribute in the *reference set* to which this descriptor applies. A zero value identifies the *referencedComponentId* within the *reference set*. Other values specify an additional attributes by its position relative to the *referencedComponentId*. Within a particular descriptor, *attributeOrder* values for a particular *referencedComponentId* must be contiguous. An unsigned *integer*, providing an ordering for the additional attributes extending the *reference set*.

At least one row must exist for each *reference set* included in a release. This row must have an *attributeOrder* value of '0' and an *attributeType* of ' *component* type' (or one of its *descendants*). The *referencedComponentId* identifies the *reference set* defined by the descriptor.

There is one additional row for each additional column present in the specified *reference set*.

Creation of *Reference set* descriptor data is mandatory when creating a new *reference set* in the *International Release* or in a *National Extension*.

Creation of a *Reference set* descriptor is optional when creating a *reference set* in another *Extension*. If a descriptor is not created, the descriptor of the closest *ancestor* of the *reference set* is used when validating *reference set* member records.

5.6.2.2.3 Metadata

The following metadata in the |Foundation metadata *concept* | *hierarchy* supports the *reference set* descriptor *reference set*.

The *Reference Set* Descriptor *Reference Set* is specified by the 900000000000456007 | Reference set descriptor | *concept* in the metadata hierarchy.

- 900000000000441003 | SNOMED CT Model Component |
    - 900000000000454005 | Foundation metadata concept |
        - 900000000000455006 | Reference set |
            - 900000000000456007 | Reference set descriptor |

Figure 42: Reference Set Descriptor Concept in the Metadata Hierarchy

Values in the *Reference Set* are populated from:

- 900000000000454005 | Foundation metadata concept |
    - 900000000000457003 | Reference set attribute |
        - 900000000000458008 | Attribute description |
        - 900000000000459000 | Attribute type |
            - 900000000000460005 | Component type |
                - 900000000000461009 | Concept type component |
                - 900000000000462002 | Description type component |
                - 900000000000463007 | Relationship type component |
                - 900000000000464001 | Reference set member type component |
            - 900000000000465000 | String |
                - 900000000000466004 | Text |
                    - 900000000000467008 | Single character |
                    - 900000000000468003 | Text < 256 bytes |
                - 900000000000469006 | URL |
                    - 900000000000470007 | HTML reference |
                    - 900000000000471006 | Image reference | ...
                - 900000000000474003 | UUID |
                - 900000000000475002 | Time |
            - 900000000000476001 | Integer |
                - 900000000000477005 | Signed integer |
                - 900000000000478000 | Unsigned integer |
            - 900000000000460005 | Component type | ...
            - 900000000000465000 | String | ...
            - 900000000000476001 | Integer | ...

- 900000000000479008 | Attribute order |
- 900000000000491004 | Attribute value | ...

Figure 43: Reference Set Attribute Metadata Hierarchy

5.6.2.2.4 Descriptor

The table below shows the descriptor that defines the structure of the 900000000000456007 | Reference set descriptor |.

Table 50: Refset Descriptor rows for 900000000000456007 | Reference set descriptor |.

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 | Reference set descriptor |	900000000000456007 | Reference set descriptor |	449608002 | Referenced component |	900000000000461009 | Concept type component |	0
900000000000456007 | Reference set descriptor |	900000000000456007 | Reference set descriptor |	900000000000458008 | Attribute description |	900000000000461009 | Concept type component |	1
900000000000456007 | Reference set descriptor |	900000000000456007 | Reference set descriptor |	900000000000459000 | Attribute type |	900000000000461009 | Concept type component |	2
900000000000456007 | Reference set descriptor |	900000000000456007 | Reference set descriptor |	900000000000479008 | Attribute order |	900000000000478000 | Unsigned integer |	3

Note: The table above omits the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the table above also includes the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.3 Simple Reference Set

5.6.2.3.1 Purpose

A 446609009 I Simple type reference set I allows a set of *components* to be specified for inclusion or exclusion for a specified purpose. This type of *reference ret* represents an extensional definition of a subset of SNOMED CT *components*. Thus it can be used to fully enumerate a subset of *concepts*, *descriptions* or *relationships*.

See also *Query specification reference set*, which can be used to represent an intensional definition of a subset of SNOMED CT *components*. In an intensional definition, the members of the subset are specified by rules rather than by enumerations (e.g. all subtypes of a specified *concepts*).

5.6.2.3.2 Reference Set Data Structure

A Simple *reference set* does not have any addition fields.

Table 51: Simple Reference Set - Data Structure

Field	Data type	Purpose
id	*UUID*	A 128 bit unsigned *integer*, uniquely identifying this *reference set member*. Different versions of a *reference set member* share the same *id* but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.
effectiveTime	*Time*	The inclusive date or time at which this version of the identified *reference set member* became the current version. The current version of this *reference set member* at time *T* is the version with the most recent *effectiveTime* prior to or equal to time *T*.
active	*Boolean*	The state of the identified *reference set member* as at the specified *effectiveTime*. If *active*= 1 (true) the *reference set member* is part of the current version of the set, if *active*= 0 (false) the *reference set member* is not part of the current version of the set.
moduleId	*SCTID*	Identifies the *SNOMED CT module* that contains this *reference set member* as at the specified *effectiveTime*. The value must be a *subtype* of 900000000000443000 I Module (core metadata concept) I within the metadata *hierarchy*.

Field	Data type	Purpose
refsetId	*SCTID*	Identifies the *reference set* to which this *reference set member* belongs. In this case, set to 900000000000456007 | Reference set descriptor |
id	*UUID*	A 128 bit unsigned *integer*, uniquely identifying the *reference set* member.
effectiveTime	*Time*	Specifies the inclusive date at which this change becomes effective.
active	*Boolean*	Specifies whether the member's state was *active* or *inactive* from the nominal release date specified by the *effectiveTime* field.
moduleId	*SCTID*	Identifies the member version's module. Set to a *subtype* of 900000000000443000 | Module (core metadata concept) | within the metadata *hierarchy*.
refsetId	*SCTID*	Set to a *child* of |Simple type| in the metadata *hierarchy*.
referencedComponentId	*SCTID*	A reference to the *SNOMED CT component* to be included in the *reference set*.

5.6.2.3.3 Metadata

Simple References Sets are *subtypes* of 446609009 | Simple type reference set | in the metadata hierarchy.

- 900000000000441003 | SNOMED CT Model Component |
  - 900000000000454005 | Foundation metadata concept |
    - 900000000000455006 | Reference set |
      - 446609009 | Simple type reference set |

Figure 44: Simple Reference Sets in the Metadata Hierarchy

5.6.2.3.4 Descriptor template

The table below shows the descriptor for a specific *reference sets* that follows the 446609009 | Simple type reference set | pattern.

Table 52: Refset Descriptor rows for 447566000 | Virtual medicinal product simple reference set |.

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 | Reference set descriptor |	447566000 | Virtual medicinal product simple reference set |	449608002 | Referenced component |	900000000000461009 | Concept type component |	0

Note: The table above omits the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the table above also shows the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.3.5 Simple reference set example

Table 53: Sample content from 447565001 I Virtual therapeutic moiety simple reference set I.

refsetId	referencedComponentId (Referenced component)
447565001 I Virtual therapeutic moiety simple reference set I	211009 I Norethandrolone preparation I
447565001 I Virtual therapeutic moiety simple reference set I	302007 I Spiramycin I
447565001 I Virtual therapeutic moiety simple reference set I	449005 I Penicillin G procaine I
447565001 I Virtual therapeutic moiety simple reference set I	544002 I Melphalan I
447565001 I Virtual therapeutic moiety simple reference set I	669007 I Vaccinia virus vaccine I
447565001 I Virtual therapeutic moiety simple reference set I	796001 I Digoxin I
447565001 I Virtual therapeutic moiety simple reference set I	847003 I D-thyroxine preparation I
447565001 I Virtual therapeutic moiety simple reference set I	922004 I Pralidoxime I
447565001 I Virtual therapeutic moiety simple reference set I	1039008 I Mercaptopurine I
447565001 I Virtual therapeutic moiety simple reference set I	1148001 I Ticarcillin I

5.6.2.4 Ordered Reference Set

5.6.2.4.1 Purpose

An 447258008 I Ordered type reference set I allows a collection of *components* to be defined with a specified given a priority ordering. This type of *reference ret* can also be used to specify ordered associations between different *components*. These can be used to specify several interrelated subsets of components and to define alternative hierarchies for navigation and selection of *concepts* or *descriptions*.

5.6.2.4.2 Data structure

An *Ordered reference set* is an *Integer Component reference set* is used to represent ordered lists and alternative hierarchies. Its structure is shown in the following table.

Table 54: Ordered reference set - Data structure

Field	Data type	Purpose
id	*UUID*	A 128 bit unsigned *integer*, uniquely identifying this *reference set member*. Different versions of a *reference set member* share the same *id* but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.
effectiveTime	*Time*	The inclusive date or time at which this version of the identified *reference set member* became the current version. The current version of this *reference set member* at time *T* is the version with the most recent *effectiveTime* prior to or equal to time *T*.
active	*Boolean*	The state of the identified *reference set member* as at the specified *effectiveTime*. If *active*= 1 (true) the *reference set member* is part of the current version of the set, if *active*= 0 (false) the *reference set member* is not part of the current version of the set.
moduleId	*SCTID*	Identifies the *SNOMED CT module* that contains this *reference set member* as at the specified *effectiveTime*. The value must be a *subtype* of 900000000000443000 | Module (core metadata concept) | within the metadata *hierarchy*.
refsetId	*SCTID*	Identifies the *reference set* to which this *reference set member* belongs. In this case, set to a *subtype* of 447258008 | Ordered type reference set |
referencedComponentId	*SCTID*	The identifier of a *SNOMED CT component* that is included in the ordered list of alternative hierarchy.

Field	Data type	Purpose
order	*Integer*	Specifies the sort *order* of the list. The list is ordered by applying an ascending sort of the *order* value. The value of *order*=1 represents the highest priority. A value of '0' is not allowed. Duplicate values are permitted and the sort order between two members with the same order value is not defined. If the *linkedToId* value is not 0, sorting occurs within subgroups that share the same *linkedToId value*. **Note:** The name "order" is a reserved word in some database environments. Please consider this when using this column.
linkedToId	*SCTID*	The identifier of a *SNOMED CT component* that acts as a grouper or hierarchy node, collecting together a subgroup from within the list. This field either enables *reference set member* linked into a number of subgroups. These subgroups can be nested allowing representation of alternative hierarchies. To link members into a subgroup, all components in the same subgroup should reference the same *component*. This can either be a component that represents the name of that subgroup or the first member of the subgroup. In the latter case, the first row of each subgroup will contain the same identifier in *referencedComponentId* and *linkedToId* and with *order*=1. To link a number of *children concepts* to a single parent *concept*, one member record should exist per *child*, with the *referencedComponentId* field referencing the parent and this field referencing the *child concept*. The *order* field is then used to *order* the *children concepts* under the parent *concept*. For ordered lists that do not require grouping or hierarchical arrangement the value of *linkedToId* should be the digit zero (0).

5.6.2.4.3 Metadata

The following metadata in the "Foundation metadata *concept*" *hierarchy* supports this *reference set*:

- 900000000000454005 | Foundation metadata concept |
  - 900000000000455006 | Reference set |

- 447258008 | Ordered type reference set | ...

Figure 45: Ordered References Sets in the Metadata Hierarchy

5.6.2.4.4 Descriptor template and examples

The tables below show the descriptor that defines the structure of the 447258008 I Ordered type reference set I pattern and an example of descriptor for a specific *reference set* that follows this pattern.

Table 55: Refset Descriptor rows for 447258008 I Ordered type reference set I

refsetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
900000000000456007 I Reference set descriptor I	447258008 I Ordered type reference set I	449608002 I Referenced component I	900000000000460005 I Component type I	0
I900000000000456007 I Reference set descriptor I	447258008 I Ordered type reference set I	447255006 I Priority order reference set attribute I	900000000000478000 I Unsigned integer I	1
900000000000456007 I Reference set descriptor I	447258008 I Ordered type reference set I	447257003 I "Linked to" reference set attribute I	900000000000460005 I Component type I	2

Table 56: Refset Descriptor rows for 447570008 I SNOMED CT top level navigation hierarchy ordered reference set I.

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 I Reference set descriptor I	447570008 I SNOMED CT top level navigation hierarchy ordered reference set I	449608002 I Referenced component I	900000000000461009 I Concept type component I	0
900000000000456007 I Reference set descriptor I	447570008 I SNOMED CT top level navigation hierarchy ordered reference set I	900000000000479008 I Attribute order I	900000000000478000 I Unsigned integer I	1
900000000000456007 I Reference set descriptor I	447570008 I SNOMED CT top level navigation hierarchy ordered reference set I	447257003 I "Linked to" reference set attribute I	900000000000461009 I Concept type component I	2

Note: The table above omits the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the table above also shows the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.4.5 Ordered reference set example

Table 57: Sample content from 447570008 | SNOMED CT top level navigation hierarchy ordered reference set |.

refsetId	referencedComponentId (Referenced component)	order (Attribute order)	linkedTo ("Linked to" reference set attribute)
447570008 | SNOMED CT top level navigation hierarchy ordered reference set |	64572001 | Disease |	1	123946008 | Disorder by body site |
447570008 | SNOMED CT top level navigation hierarchy ordered reference set |	64572001 | Disease |	2	370117001 | Disorder of system |
447570008 | SNOMED CT top level navigation hierarchy ordered reference set |	64572001 | Disease |	3	278919001 | Communication disorder |
447570008 | SNOMED CT top level navigation hierarchy ordered reference set |	64572001 | Disease |	4	74732009 | Mental disorder |
447570008 | SNOMED CT top level navigation hierarchy ordered reference set |	64572001 | Disease |	5	39898005 | Sleep disorder |
447570008 | SNOMED CT top level navigation hierarchy ordered reference set |	64572001 | Disease |	6	370118006 | Disorder of pregnancy / labor / delivery / puerperium |
447570008 | SNOMED CT top level navigation hierarchy ordered reference set |	64572001 | Disease |	7	370119003 | Fetal / neonatal / perinatal disorder |
447570008 | SNOMED CT top level navigation hierarchy ordered reference set |	64572001 | Disease |	8	370120009 | Endocrine / nutritional / metabolic disorder |

refsetId	referencedComponentId (Referenced component)	order (Attribute order)	linkedTo ("Linked to" reference set attribute)
447570008 | SNOMED CT top level navigation hierarchy ordered reference set |	64572001 | Disease |	9	370121008 | Disorder of blood / lymphatics / immune system |
447570008 | SNOMED CT top level navigation hierarchy ordered reference set |	64572001 | Disease |	10	281867008 | Multisystem disorder |

5.6.2.5 Attribute Value Reference Set

5.6.2.5.1 Purpose

An 900000000000480006 | Attribute value type reference set | allows a value from a specified range to be associated with a *component*. This type of *reference set* can be use for a range of purposes where there is a requirement to provide additional information about particular *concepts*, *descriptions* or *relationships*. For example, an 900000000000480006 | Attribute value type reference set | is used to indicate the reason why a *concepts* has been *inactivated*.

5.6.2.5.2 Data Structure

A *Attribute value reference set* is a *component reference set* used to apply a tagged value to a *SNOMED CT component*. Its structure is shown in the following table.

Table 58: Attribute value reference set - Data structure

Field	Data type	Purpose
id	*UUID*	A 128 bit unsigned *integer*, uniquely identifying this *reference set member*. Different versions of a *reference set member* share the same *id* but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.
effectiveTime	*Time*	The inclusive date or time at which this version of the identified *reference set member* became the current version. The current version of this *reference set member* at time *T* is the version with the most recent *effectiveTime* prior to or equal to time *T*.

Field	Data type	Purpose
active	*Boolean*	The state of the identified *reference set member* as at the specified *effectiveTime*. If *active*= 1 (true) the *reference set member* is part of the current version of the set, if *active*= 0 (false) the *reference set member* is not part of the current version of the set.
moduleId	*SCTID*	Identifies the *SNOMED CT module* that contains this *reference set member* as at the specified *effectiveTime*. The value must be a *subtype* of 900000000000443000 | Module (core metadata concept) | within the metadata *hierarchy*.
refsetId	*SCTID*	Identifies the *reference set* to which this *reference set member* belongs. In this case, set to a *subtype* of 900000000000480006 | Attribute value type reference set |
referencedComponentId	*SCTID*	A reference to the *SNOMED CT component* being tagged with a value.
valueId	*SCTID*	The tagged value applied to the *referencedComponentId*. A *subtype* of 900000000000491004 | Attribute value |.

5.6.2.5.3 Metadata

The following metadata in the "Foundation metadata *concept*" *hierarchy* supports this *reference set*:

- 900000000000454005 | Foundation metadata concept |
  - 900000000000455006 | Reference set |
    - 900000000000480006 | Attribute value type |
      - 900000000000488004 | Relationship refinability reference set |
      - 900000000000489007 | Concept inactivation indicator reference set |
      - 900000000000490003 | Description inactivation indicator reference set |
      - 900000000000547002 | Relationship inactivation indicator reference set |

Figure 46: Attribute Value Reference Sets in the Metadata Hierarchy

5.6.2.5.4 Descriptor template and examples

The tables below show the descriptors that define examples of *reference sets* that follow the I Attribute value type reference set I pattern.

Table 59: Refset Descriptor rows for 900000000000489007 I Concept inactivation indicator reference set I.

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 I Reference set descriptor I	900000000000489007 I Concept inactivation indicator reference set I	449608002 I Referenced component I	900000000000461009 I Concept type component I	0
900000000000456007 I Reference set descriptor I	900000000000489007 I Concept inactivation indicator reference set I	900000000000481005 I Concept inactivation value I	900000000000461009 I Concept type component I	1

Table 60: Refset Descriptor rows for 900000000000490003 I Description inactivation indicator reference set I.

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 I Reference set descriptor I	900000000000490003 I Description inactivation indicator reference set I	449608002 I Referenced component I	900000000000462002 I Description type component I	0
900000000000456007 I Reference set descriptor I	900000000000490003 I Description inactivation indicator reference set I	900000000000493001 I Description inactivation value I	900000000000461009 I Concept type component I	1

Note: The tables above omit the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the tables above also show the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.5.5 Attribute Value reference set example

Table 61: Sample content from 900000000000489007 | Concept inactivation indicator reference set |.

refsetId	referencedComponentId (Referenced component)	valueId (Concept inactivation value)
900000000000489007 | Concept inactivation indicator reference set |	105000 | Poisoning by pharmaceutical excipient |	900000000000482003 | Duplicate |
900000000000489007 | Concept inactivation indicator reference set |	123008 | Channel catfish virus disease |	900000000000487009 | Moved elsewhere |
900000000000489007 | Concept inactivation indicator reference set |	141000 | Glaucoma as birth trauma |	900000000000482003 | Duplicate |
900000000000489007 | Concept inactivation indicator reference set |	157000 | AIDS with low vision |	900000000000484002 | Ambiguous |
900000000000489007 | Concept inactivation indicator reference set |	190000 | Partial hysterectomy |	900000000000484002 | Ambiguous |
900000000000489007 | Concept inactivation indicator reference set |	203004 | Replacement of pacemaker in brain |	900000000000484002 | Ambiguous |
900000000000489007 | Concept inactivation indicator reference set |	212002 | Salmonella III arizonae 53:k:z |	900000000000483008 | Outdated |
900000000000489007 | Concept inactivation indicator reference set |	215000 | Operative procedure on fingers |	900000000000482003 | Duplicate |
900000000000489007 | Concept inactivation indicator reference set |	220000 | Unspecified monoarthritis |	900000000000486000 | Limited |
900000000000489007 | Concept inactivation indicator reference set |	236003 | Incision of vein |	900000000000484002 | Ambiguous |

5.6.2.6 Simple Map Reference Set

5.6.2.6.1 Purpose

A 900000000000496009 | Simple map reference set | allows representation of simple maps between *SNOMED CT concepts* and values in other code systems. No constrains are put on the number of coding schemes supported, the number of codes within a particular scheme mapped to by a single *SNOMED CT concept* or the number of *SNOMED CT concepts* mapping to a particular code. However, this type of *reference set* is usually only appropriate where there is a close "one-to-one" mapping between *SNOMED CT concepts* and coded values in anothe code system.

5.6.2.6.2 Data structure

A *Simple map reference set* is a *String reference set* used to represent one-to-one maps between SNOMED CT concepts and codes in another terminology, classification or code system. Its structure is shown in the following table.

Table 62: Simple map reference set - Data structure

Field	Data type	Purpose
id	*UUID*	A 128 bit unsigned *integer*, uniquely identifying this *reference set member*. Different versions of a *reference set member* share the same *id* but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.
effectiveTime	*Time*	The inclusive date or time at which this version of the identified *reference set member* became the current version. The current version of this *reference set member* at time *T* is the version with the most recent *effectiveTime* prior to or equal to time *T*.
active	*Boolean*	The state of the identified *reference set member* as at the specified *effectiveTime*. If *active*= 1 (true) the *reference set member* is part of the current version of the set, if *active*= 0 (false) the *reference set member* is not part of the current version of the set.
moduleId	*SCTID*	Identifies the *SNOMED CT module* that contains this *reference set member* as at the specified *effectiveTime*. The value must be a *subtype* of 900000000000443000	Module (core metadata concept)	within the metadata *hierarchy*.

Field	Data type	Purpose
refsetId	*SCTID*	Identifies the *reference set* to which this *reference set member* belongs. In this case, set to a *subtype* of 900000000000496009 | Simple map type reference set |
referencedComponentId	*SCTID*	The identifier of the *SNOMED CT concept* being mapped.
mapTarget	*String*	The equivalent code in the other terminology, classification or code system.

5.6.2.6.3 Metadata

The following metadata *hierarchy* supports this *reference set*:

- 900000000000454005 | Foundation metadata concept |
  - 900000000000455006 | Reference set |
    - 900000000000456007 | Reference set descriptor | ...
    - 900000000000480006 | Attribute value type | ...
    - 900000000000496009 | Simple map |
      - 900000000000497000 | CTV3 simple map |
      - 900000000000498005 | SNOMED RT ID simple map |

Figure 47: Simple Map Reference Sets in the Metadata Hierarchy

5.6.2.6.4 Descriptor template and examples

The tables below show the descriptors that define examples of *reference sets* that follow the 900000000000496009 | Simple map reference set | pattern.

Table 63: Refset Descriptor rows for 446608001 | ICD-O simple map reference set |.

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 | Reference set descriptor |	446608001 | ICD-O simple map reference set |	900000000000500006 | Map source concept |	900000000000461009 | Concept type component |	0
900000000000456007 | Reference set descriptor |	446608001 | ICD-O simple map reference set |	900000000000499002 | Scheme value |	900000000000465000 | String |	1

Table 64: Refset Descriptor rows for 900000000000498005 | SNOMED RT ID simple map |.

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 | Reference set descriptor |	900000000000498005 | SNOMED RT ID simple map |	900000000000500006 | Map source concept |	900000000000461009 | Concept type component |	0
900000000000456007 | Reference set descriptor |	900000000000498005 | SNOMED RT ID simple map |	900000000000499002 | Scheme value |	900000000000465000 | String |	1

Note: The tables above omit the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the tables above also show the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.6.5 Simple Map Refset Examples

Table 65: Sample content from 900000000000498005 | SNOMED RT ID simple map |.

refsetId	referencedComponentId (Map source concept)	mapTarget (Scheme value)
900000000000498005 | SNOMED RT ID simple map |	100005 | SNOMED RT Concept |	G-3000
900000000000498005 | SNOMED RT ID simple map |	101009 | Quilonia ethiopica |	L-55535
900000000000498005 | SNOMED RT ID simple map |	102002 | Hemoglobin Okaloosa |	F-D5972
900000000000498005 | SNOMED RT ID simple map |	103007 | Squirrel fibroma virus |	L-37904
900000000000498005 | SNOMED RT ID simple map |	104001 | Excision of lesion of patella |	P1-18376
900000000000498005 | SNOMED RT ID simple map |	105000 | Poisoning by pharmaceutical excipient |	DD-82950
900000000000498005 | SNOMED RT ID simple map |	106004 | Structure of posterior carpal region |	T-D8602
900000000000498005 | SNOMED RT ID simple map |	107008 | Structure of fetal part of placenta |	T-F1102
900000000000498005 | SNOMED RT ID simple map |	108003 | Entire condylar emissary vein |	T-49723
900000000000498005 | SNOMED RT ID simple map |	109006 | Anxiety disorder of childhood OR adolescence |	D9-12000

Table 66: Sample content from 446608001 | ICD-O simple map reference set |.

refsetId	referencedComponentId (Map source concept)	mapTarget (Scheme value)
446608001 | ICD-O simple map reference set |	106004 | Structure of posterior carpal region |	C76.4
446608001 | ICD-O simple map reference set |	107008 | Structure of fetal part of placenta |	C58.9
446608001 | ICD-O simple map reference set |	108003 | Entire condylar emissary vein |	C49.0

refsetId	referencedComponentId (Map source concept)	mapTarget (Scheme value)
446608001 | ICD-O simple map reference set |	110001 | Structure of visceral layer of Bowman's capsule |	C64.9
446608001 | ICD-O simple map reference set |	111002 | Parathyroid structure |	C75.0
446608001 | ICD-O simple map reference set |	116007 | Subcutaneous tissue structure of medial surface of index finger |	C49.1
446608001 | ICD-O simple map reference set |	124002 | Structure of coronoid process of mandible |	C41.1
446608001 | ICD-O simple map reference set |	155008 | Structure of deep circumflex iliac artery |	C49.5
446608001 | ICD-O simple map reference set |	167005 | Structure of supraclavicular part of brachial plexus |	C47.1
446608001 | ICD-O simple map reference set |	202009 | Structure of anterior division of renal artery |	C49.4

5.6.2.7 Complex and Extended Map Reference Sets

5.6.2.7.1 Purpose

A 447250001 | Complex map type reference set | enables representation of maps where each *SNOMED CT concept* may map to one or more codes in a *target scheme*. The type of *reference set* supports the general set of mapping data required to enable a *target code* to be selected at run-time from a number of alternate codes. It supports *target code* selection by accommodating the inclusion of machine readable rules and/or human readable advice. An 609331003 | Extended map type reference set | adds an additional field to allow categorization of maps.

5.6.2.7.2 Data structure

A *Complex map reference set* is an *Integer-Integer-String-String-String-Component reference set*. The pattern is currently used for the map to ICD-9-CM. Its structure is as shown in the following table, with one exception - the table includes an additional field (*mapCategoryId*) which is not used for this type of map.

An *Extended map reference set* follows the same pattern but adds one additional column. It is an *Integer-Integer-String-String-String-Component-Component reference set* and this pattern is currently used for maps to ICD-10. Its structure is shown in the following table.

Table 67: Complex and Extended map reference sets - Data structures

Field	Data type	Purpose
id	*UUID*	A 128 bit unsigned *integer*, uniquely identifying this *reference set member*. Different versions of a *reference set member* share the same *id* but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.
effectiveTime	*Time*	The inclusive date or time at which this version of the identified *reference set member* became the current version. The current version of this *reference set member* at time *T* is the version with the most recent *effectiveTime* prior to or equal to time *T*.
active	*Boolean*	The state of the identified *reference set member* as at the specified *effectiveTime*. If *active*= 1 (true) the *reference set member* is part of the current version of the set, if *active*= 0 (false) the *reference set member* is not part of the current version of the set.
moduleId	*SCTID*	Identifies the *SNOMED CT module* that contains this *reference set member* as at the specified *effectiveTime*. The value must be a *subtype* of 900000000000443000 | Module (core metadata concept) | within the metadata *hierarchy*.
refsetId	*SCTID*	Identifies the *reference set* to which this *reference set member* belongs. In this case, a *subtype* of 447250001 | Complex map type reference set | or 447250001 | Complex map type reference set |.
referencedComponentId	*SCTID*	A reference to the *SNOMED CT concept* being mapped.
mapGroup	*Integer*	An *integer*, grouping a set of complex map records from which one may be selected as a *target code*. Where a *SNOMED CT concept* maps onto 'n' *target codes*, there will be 'n' groups, each containing one or more complex map records.

Field	**Data type**	**Purpose**
mapPriority	*Integer*	Within a *mapGroup*, the mapPriority specifies the *order* in which complex map records should be checked. Only the first map record meeting the run - time selection criteria will be taken as the *target code* within the group of alternate codes.
mapRule	*String*	A machine-readable rule, (evaluating to either 'true' or 'false' at run-time) that indicates whether this map record should be selected within its mapGroup.
mapAdvice	*String*	Human-readable advice, that may be employed by the software vendor to give an end-user advice on selection of the appropriate *target code* from the alternatives presented to him within the group.
mapTarget	*String*	The *target code* in the target terminology, classification or code system.
correlationId	*SCTID*	A *child* of 447247004 I SNOMED CT source code to target map code correlation value I in the metadata *hierarchy*, identifying the correlation between the *SNOMED CT concept* and the *target code*.
The following additional field only applies to Extended Map Reference Sets		
mapCategoryId	*SCTID*	Identifies the *SNOMED CT concept* in the metadata hierarchy which represents the MapCategory for the associated map member. The categories vary for different *target code* systems, each set of categories is represented by a *subtype* of 609330002 I Map category value I. In the case of *ICD-10* the individual category values are *subtypes* of: 447634004 I ICD-10 map category value I.

Values for *mapGroup* are allocated on a sequential basis (for each *refsetId* and *referencedComponentId* combination) during authoring starting at 1. However, distributed *mapGroup* are not necessarily sequential, as some *mapGroups* may be created and removed during a mapping process between releases. For maps where each *SNOMED CT concept* only maps to at most one of a group of alternate *target codes*, the mapGroup field are usually be set to '1'.

Values for *mapPriority* will be allocated on a sequential basis (within each map group) starting from '1'. For maps that do not require run - time alternatives, the *mapPriority* field is set to '1'.

The *mapRule* and *mapAdvice* fields enable run-time selection (within vendor's software) from a number of alternative map records within a *mapGroups*. Where there are no alternatives maps these columns of the release files will be empty (zero length string). Where alternative maps exist one or both of columns will be populated where relevant information is available.

Where both fields are populated, and a vendor's system is capable of processing a machine readable rule, this should take priority over the human readable advice. Where neither field is populated, a vendor's system should allow the end-user to select the appropriate *target code* from the alternates.

5.6.2.7.3 Metadata

The following metadata supports this *reference set*:

- 900000000000454005 | Foundation metadata concept |
  - 900000000000455006 | Reference set |
    - 447250001 | Complex map type reference set | ...
    - 609331003 | Extended map type reference set | ...

Figure 48: Complex Map References Sets in the Metadata Hierarchy

5.6.2.7.4 Descriptor templates

The tables below examples of the descriptors for specific *reference sets* that follow the 447250001 I Complex map type reference set I and 609331003 I Extended map type reference set I patterns.

Table 68: Refset Descriptor rows for 447563008 I ICD-9-CM equivalence complex map reference set I.

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 I Reference set descriptor I	447563008 I ICD-9-CM equivalence complex map reference set I	900000000000500006 I Map source concept I	900000000000461009 I Concept type component I	0
900000000000456007 I Reference set descriptor I	447563008 I ICD-9-CM equivalence complex map reference set I	900000000000501005 I Map group I	900000000000478000 I Unsigned integer I	1
900000000000456007 I Reference set descriptor I	447563008 I ICD-9-CM equivalence complex map reference set I	900000000000502003 I Map priority I	900000000000478000 I Unsigned integer I	2
900000000000456007 I Reference set descriptor I	447563008 I ICD-9-CM equivalence complex map reference set I	900000000000503008 I Map rule I	900000000000465000 I String I	3
900000000000456007 I Reference set descriptor I	447563008 I ICD-9-CM equivalence complex map reference set I	900000000000504002 I Map advice I	900000000000465000 I String I	4
900000000000456007 I Reference set descriptor I	447563008 I ICD-9-CM equivalence complex map reference set I	900000000000505001 I Map target I	900000000000465000 I String I	5
900000000000456007 I Reference set descriptor I	447563008 I ICD-9-CM equivalence complex map reference set I	447247004 I SNOMED CT source code to target map code correlation value I	900000000000461009 I Concept type component I	6

Table 69: Refset Descriptor rows for 447562003 | ICD-10 complex map reference set | (Extended map type)

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 | Reference set descriptor |	447562003 | ICD-10 complex map reference set |	900000000000500006 | Map source concept |	900000000000461009 | Concept type component |	0
900000000000456007 | Reference set descriptor |	447562003 | ICD-10 complex map reference set |	900000000000501005 | Map group |	900000000000478000 | Unsigned integer |	1
900000000000456007 | Reference set descriptor |	447562003 | ICD-10 complex map reference set |	900000000000502003 | Map priority |	900000000000478000 | Unsigned integer |	2
900000000000456007 | Reference set descriptor |	447562003 | ICD-10 complex map reference set |	900000000000503008 | Map rule |	900000000000465000 | String |	3
900000000000456007 | Reference set descriptor |	447562003 | ICD-10 complex map reference set |	900000000000504002 | Map advice |	900000000000465000 | String |	4
900000000000456007 | Reference set descriptor |	447562003 | ICD-10 complex map reference set |	900000000000505001 | Map target |	900000000000465000 | String |	5
900000000000456007 | Reference set descriptor |	447562003 | ICD-10 complex map reference set |	447247004 | SNOMED CT source code to target map code correlation value |	900000000000461009 | Concept type component |	6
900000000000456007 | Reference set descriptor |	447562003 | ICD-10 complex map reference set |	609330002 | Map category value |	900000000000461009 | Concept type component |	7

Note: The tables above omit the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the tables above also show the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.7.5 Mapping rule specifications

The specific grammar and content of the rules for resolving complex mapping cases depends on the nature of the *target code* system or classification. In general, each map is accompanied by a rule which is tested against other data and can be evaluated to return one of the following values:

- **True** - in which case the map target applies;
- **False** - in which case the map target does not apply;
- **Indeterminate** - in cases where there is insufficient accessible data to determine whether the map target applies. In this case manual resolution of the map using the map advice provided will be required.

The mapping rules assume access to a number of variables, that can be bound to appropriate attributes in the vendor's system information model. These include the age and gender of the patient and information about coexisting situations (e.g. records of other disorders, procedures or events in the same patient record).

Detailed definitions of the mapping rules used forms part of individual specifications for maps to particular *target code* systems and classifications. This will initially be provided separately and will accompany the release of the relevant mapping files. For example, the set of rules used for mapping to *ICD-10* are currently included in a document released with those maps.

5.6.2.7.6 Extended map reference set example

Table 70: Sample content from 447562003 | ICD-10 complex map reference set |.

refSetId	referencedComponentId (Map source concept)	mapGroup (Map group)	mapPriority (Map priority)	mapRule (Map rule)	mapAdvice (Map advice)	mapTarget (Map target)	correlationId (SNOMED CT source code to target map code correlation value)	mapCategoryId (Map category value)
447562003 | ICD-10 complex map reference set |	127009 | Miscarriage with laceration of cervix |	1	1	TRUE	ALWAYS O03.8	O03.8	447561005 | SNOMED CT source code to target map code correlation not specified |	447637006 | Map source concept is properly classified |
447562003 | ICD-10 complex map reference set |	127009 | Miscarriage with laceration of cervix |	2	1	TRUE	ALWAYS O08.6	O08.6	447561005 | SNOMED CT source code to target map code correlation not specified |	447637006 | Map source concept is properly classified |
447562003 | ICD-10 complex map reference set |	140004 | Chronic pharyngitis |	1	1	IFA 90979004 | Chronic tonsillitis (disorder) |	IF CHRONIC TONSILLITIS CHOOSE J35.0 | MAP OF SOURCE CONCEPT IS CONTEXT DEPENDENT	J35.0	447561005 | SNOMED CT source code to target map code correlation not specified |	447639009 | Map of source concept is context dependent |

refSetId	referencedComponentId (Map source concept)	mapGroup (Map group)	mapPriority (Map priority)	mapRule (Map rule)	mapAdvice (Map advice)	mapTarget (Map target)	correlationId (SNOMED CT source code to target map code correlation value)	mapCategoryId (Map category value)
447562003 | ICD-10 complex map reference set |	140004 | Chronic pharyngitis |	1	2	IFA 232406009 | Chronic pharyngeal candidiasis (disorder) |	IF CHRONIC PHARYNGEAL CANDIDIASIS CHOOSE B37.8 | MAP OF SOURCE CONCEPT IS CONTEXT DEPENDENT	B37.8	447561005 | SNOMED CT source code to target map code correlation not specified |	447639009 | Map of source concept is context dependent |
447562003 | ICD-10 complex map reference set |	140004 | Chronic pharyngitis |	1	3	OTHERWISE TRUE	ALWAYS J31.2	J31.2	447561005 | SNOMED CT source code to target map code correlation not specified |	447637006 | Map source concept is properly classified |
447562003 | ICD-10 complex map reference set |	162004 | Severe manic bipolar I disorder without psychotic features |	1	1	TRUE	ALWAYS F31.1	F31.1	447561005 | SNOMED CT source code to target map code correlation not specified |	447637006 | Map source concept is properly classified |
447562003 | ICD-10 complex map reference set |	177007 | Poisoning by sawfly larvae |	1	1	TRUE	ALWAYS T63.4	T63.4	447561005 | SNOMED CT source code to target map code correlation not specified |	447637006 | Map source concept is properly classified |

refSetId	referencedComponentId (Map source concept)	mapGroup (Map group)	mapPriority (Map priority)	mapRule (Map rule)	mapAdvice (Map advice)	mapTarget (Map target)	correlationId (SNOMED CT source code to target map code correlation value)	mapCategoryId (Map category value)
447562003 | ICD-10 complex map reference set |	177007 | Poisoning by sawfly larvae |	2	1	TRUE	ALWAYS X25 | POSSIBLE REQUIREMENT FOR PLACE OF OCCURRENCE	X25	447561005 | SNOMED CT source code to target map code correlation not specified |	447637006 | Map source concept is properly classified |
447562003 | ICD-10 complex map reference set |	181007 | Hemorrhagic bronchopneumonia |	1	1	TRUE	ALWAYS J18.0	J18.0	447561005 | SNOMED CT source code to target map code correlation not specified |	447637006 | Map source concept is properly classified |
447562003 | ICD-10 complex map reference set |	183005 | Autoimmune pancytopenia |	1	1	TRUE	ALWAYS D61.8	D61.8	447561005 | SNOMED CT source code to target map code correlation not specified |	447637006 | Map source concept is properly classified |

5.6.2.8 Language Reference Set

5.6.2.8.1 Purpose

A 900000000000506000 | Language type reference set | supports the representation of *language* and *dialects* preferences for the use of particular *descriptions*. The most common use case for this type of *reference set* is to specify the acceptable and preferred terms for use within a particular country or region. However, the same type of *reference set* can also be used to represent preferences for use of *descriptions* in a more specific context such as a clinical specialty, organization or department.

5.6.2.8.2 Data structure

A *Language reference set* is a *Component reference set* that is used to indicate which *descriptions* contain *terms* that are acceptable or preferred in a particular *language* or *dialect*. Its structure is shown in the following table.

Table 71: Language reference set - Data structure

Field	Data type	Purpose
id	*UUID*	A 128 bit unsigned *integer*, uniquely identifying this *reference set member*. Different versions of a *reference set member* share the same *id* but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.
effectiveTime	*Time*	The inclusive date or time at which this version of the identified *reference set member* became the current version. The current version of this *reference set member* at time *T* is the version with the most recent *effectiveTime* prior to or equal to time *T*.
active	*Boolean*	The state of the identified *reference set member* as at the specified *effectiveTime*. If *active*= 1 (true) the *reference set member* is part of the current version of the set, if *active*= 0 (false) the *reference set member* is not part of the current version of the set.
moduleId	*SCTID*	Identifies the *SNOMED CT module* that contains this *reference set member* as at the specified *effectiveTime*. The value must be a *subtype* of 900000000000443000	Module (core metadata concept)	within the metadata *hierarchy*.
refsetId	*SCTID*	Identifies the *reference set* to which this *reference set member* belongs. In this case, a *subtype* of 900000000000506000	Language type reference set	.

Field	Data type	Purpose
referencedComponentId	*SCTID*	The identifier of a *description* included in the *language reference set*.
acceptabilityId	*SCTID*	A *subtype* of 900000000000511003 | Acceptability | indicating whether the *description* is acceptable or preferred for use in the specified *language* or *dialect*.

In a *Language reference set*:

- No more than one *description* of a specific *description type* associated with a single *concept* may have the *acceptabilityId* value 900000000000548007 | Preferred |.
- At least one *description* of *description typesynonym* associated with each *concept* should have the *acceptabilityId* value 900000000000548007 | Preferred |. This is the *preferred term* for that *concept* in the specified *language* or *dialect*.

5.6.2.8.3 Metadata

The following metadata supports this *reference set*:

- 900000000000454005 | Foundation metadata concept |
  - 900000000000455006 | Reference set |
    - 900000000000456007 | Reference set descriptor |
    - 900000000000480006 | Attribute value type | ...
    - 900000000000496009 | Simple map | ...
    - 900000000000506000 | Language type |
      - 900000000000507009 | English |
        - 900000000000508004 | GB English |
        - 900000000000509007 | US English |

Figure 49: Language References Sets in the Metadata Hierarchy

The immediate *children* of | *Language* type| will be *languages*. This level may be used to represent the "correct" *language*, where a *language* authority exists. In most cases, however, this level is likely to be empty.

5.6.2.8.4 Descriptor template

The table below shows an example of the descriptor for a specific *reference sets* that follows the 900000000000506000 I Language type reference set I pattern.

Table 72: Refset Descriptor rows for 900000000000508004 I GB English I.

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 I Reference set descriptor I	900000000000508004 I GB English I	900000000000510002 I Description in dialect I	900000000000462002 I Description type component I	0
900000000000456007 I Reference set descriptor I	900000000000508004 I GB English I	900000000000511003 I Acceptability I	900000000000461009 I Concept type component I	1

Note: The tables above omit the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the tables above also show the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.8.5 Examples of language reference sets

Table 73: Sample content from 900000000000509007 I US English I.

refsetId	referencedComponentId (Description in dialect)	acceptabilityId (Acceptability)
900000000000509007 I US English I	42969009 I Cauterization of skin {id:71693012} I	900000000000548007 I Preferred I
900000000000509007 I US English I	42969009 I Fulguration of subcutaneous tissue {id:71695017} I	900000000000549004 I Acceptable I
900000000000509007 I US English I	80146002 I Appendectomy {id:132967011} I	900000000000548007 I Preferred I
900000000000509007 I US English I	80146002 I Excision of appendix {id:132972019} I	900000000000549004 I Acceptable I
900000000000509007 I US English I	271737000 I Anemia {id:406636013} I	900000000000548007 I Preferred I
900000000000509007 I US English I	271737000 I Absolute anemia {id:406640016} I	900000000000549004 I Acceptable I

Table 74: Sample content from 900000000000508004 I GB English I.

refsetId	referencedComponentId (Description in dialect)	acceptabilityId (Acceptability)
900000000000508004 I GB English I	42969009 I Cauterisation of skin {id:493493018} I	900000000000548007 I Preferred I
900000000000508004 I GB English I	42969009 I Fulguration of subcutaneous tissue {id:71695017} I	900000000000549004 I Acceptable I
900000000000508004 I GB English I	80146002 I Excision of appendix {id:132972019} I	900000000000549004 I Acceptable I
900000000000508004 I GB English I	80146002 I Appendicectomy {id:132973012} I	900000000000548007 I Preferred I
900000000000508004 I GB English I	271737000 I Anaemia {id:406638014} I	900000000000548007 I Preferred I
900000000000508004 I GB English I	271737000 I Absolute anaemia {id:406641017} I	900000000000549004 I Acceptable I

In the above examples, | Excision of appendix | is acceptable in both US and GB English. However, | Appendectomy | is preferred in US English and | Appendicectomy | is preferred in GB English.

Note: Any *description* which is not referenced by an active row in the relevant language *reference set* is regarded as unacceptable (i.e. not a valid *synonym* in the language or *dialect*).

5.6.2.9 Query Specification Reference Set

5.6.2.9.1 Purpose

A 900000000000512005 | Query specification type reference set | allows a serialised query to represent the membership of a subset of *SNOMED CT components*. A *query* contained in the *reference set* is run against the content of *SNOMED CT* to produce a subset of *concepts*, *descriptions* or *relationships*. The query is referred to an intensional definition of the subset. It can be run against future releases of *SNOMED CT* to generate an updated set of subset members.

The members of the resulting subset may also be represented in an enumerated form as a *Simple reference set*. An enumerated representation of a subset is referred to as an extensional definition.

5.6.2.9.2 Data structure

A *Query specification reference set* is a *String reference set* containing *queries* that represent intensional definitions of subsets of *components*. The result of running the *query* is an extensional representation of the subset of *components* which can be represented as a *Simple reference set*. Its structure is shown in the following table.

Table 75: Query specification reference set - Data structure

Field	Data type	Purpose
id	*UUID*	A 128 bit unsigned *integer*, uniquely identifying this *reference set member*. Different versions of a *reference set member* share the same *id* but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.
effectiveTime	*Time*	The inclusive date or time at which this version of the identified *reference set member* became the current version. The current version of this *reference set member* at time *T* is the version with the most recent *effectiveTime* prior to or equal to time *T*.
active	*Boolean*	The state of the identified *reference set member* as at the specified *effectiveTime*. If *active*= 1 (true) the *reference set member* is part of the current version of the set, if *active*= 0 (false) the *reference set member* is not part of the current version of the set.

Field	Data type	Purpose
moduleId	*SCTID*	Identifies the *SNOMED CT module* that contains this *reference set member* as at the specified *effectiveTime*. The value must be a *subtype* of 900000000000443000 I Module (core metadata concept) I within the metadata *hierarchy*.
refsetId	*SCTID*	Identifies the *reference set* to which this *reference set member* belongs. In this case, set to a subtype of 900000000000512005 I Query specification type I
referencedComponentId	*SCTID*	The identifier (*refsetId*) of the *reference set* for which members are to be generated.
query	*String*	The serialised *query* that can be used to (re-)generate the *reference set* members. A standard syntax for use in these queries is currently under development and is due for publication in late 2014.

5.6.2.9.3 Metadata

The following metadata in the "Foundation metadata *concept*" *hierarchy* supports this *reference set*:

- 900000000000454005 I Foundation metadata concept I
  - 900000000000455006 I Reference set I
    - 900000000000512005 I Query specification type I
      - 900000000000513000 I Simple query specification I

Figure 50: Hierarchy of Foundation metadata concept

5.6.2.9.4 Descriptor template

The table below shows the descriptor that defines the structure of the 900000000000512005 I Query specification type reference set I pattern.

Table 76: Descriptor Template for Query Specification Reference Sets

refsetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
900000000000456007 I Reference set descriptor I	900000000000512005 I Query specification type reference set I	900000000000514006 I Generated reference set I	900000000000461009 I Concept type component I	0
900000000000456007 I Reference set descriptor I	900000000000512005 I Query specification type reference set I	900000000000515007 I Query I	900000000000465000 I String I	1

Note: The tables above omit the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the tables above also show the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.9.5 Example usage

In the example below, "serialised *query* 1" is a text *string* that can be used to generate members for *Reference set 1*, which is a simple member *reference set* (without any additional fields within its member records).

Table 77: Example rows from Query Specification Reference Set

refsetId	referencedComponentId	query
| Simple query specification |	| Target reference set |	Serialized text of the query ...

5.6.2.9.6 Query language specification

The specification of the *query language* has yet to be defined / selected, but it should be capable of:

- Selecting *concepts* using primary fields, subsumption testing, *relationships*, *relationship groups*, set operators (*union*, *intersection*, excludes), and lexical *query*;
- Selecting *descriptions*, *relationships* and *reference sets* using similar mechanisms;
- Calculation of values for the *reference set*'s extended fields. Identifying the version of the syntax and any *language* syntax variations.
- Queries that support definitions for terminologies other than *SNOMED CT* should also be supported. For example, queries to link or include codes in ICD-10, ICD-11, ICPC and *LOINC*.

Note: During 2014 work is underway to develop and pilot a standard approach to representation of queries including queries for generation of subset of *SNOMED CT concepts*.

5.6.2.10 Annotation Reference Set

5.6.2.10.1 Purpose

An 900000000000516008 | Annotation type reference set | allows *text strings* to be associated with components for any specified purpose.

5.6.2.10.2 Data structure

An annotation reference set *String reference set* used to apply text *annotations* to selected *SNOMED CT components*.

Table 78: Annotation reference set - Data structure

Field	Data type	Purpose
id	*UUID*	A 128 bit unsigned *integer*, uniquely identifying this *reference set member*. Different versions of a *reference set member* share the same *id* but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.

Field	**Data type**	**Purpose**
effectiveTime	*Time*	The inclusive date or time at which this version of the identified *reference set member* became the current version. The current version of this *reference set member* at time *T* is the version with the most recent *effectiveTime* prior to or equal to time *T*.
active	*Boolean*	The state of the identified *reference set member* as at the specified *effectiveTime*. If *active*= 1 (true) the *reference set member* is part of the current version of the set, if *active*= 0 (false) the *reference set member* is not part of the current version of the set.
moduleId	*SCTID*	Identifies the *SNOMED CT module* that contains this *reference set member* as at the specified *effectiveTime*. The value must be a *subtype* of 900000000000443000 I Module (core metadata concept) I within the metadata *hierarchy*.
refsetId	*SCTID*	Identifies the *reference set* to which this *reference set member* belongs. In this case, set to 900000000000516008 I Annotation type reference set I.
referencedComponentId	*SCTID*	The identifier of the *component* to be annotated.
annotation	*String*	The text *annotation* to attach to the *component* identified by *referencedComponentId*.

5.6.2.10.3 Metadata

The following metadata in supports this *reference set*:

- 900000000000454005 I Foundation metadata concept I
    - 900000000000455006 I Reference set I
        - 900000000000516008 I Annotation type I
            - 900000000000517004 I Associated image I

Figure 51: Annotation References Sets in the Metadata Hierarchy

5.6.2.10.4 Descriptor template

The tables below show the descriptors that define the structure of the 900000000000516008 | Annotation type reference set | pattern and examples of the descriptors for specific *reference sets* that follow this pattern.

Table 79: Descriptor Template for Annotation Reference Sets

refsetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
900000000000456007 | Reference set descriptor |	900000000000516008 | Annotation type |	900000000000518009 | Annotated component |	900000000000461009 | Concept type component |	0
900000000000456007 | Reference set descriptor |	900000000000516008 | Annotation type |	900000000000519001 | Annotation |	900000000000465000 | String |	1

The *attributeType* for the *Annotation* field can be any *descendant* of the "*string*" *concept* in the metadata *hierarchy*. This *hierarchy* is described in more detail under the "*Reference set* descriptor" section.

The table below holds the Descriptor for the "Associated image" *annotation reference set*, which allows URLs to be associated with *concepts*:

Table 80: Descriptor for 900000000000517004 | Associated image reference set |

refsetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
900000000000456007 | Reference set descriptor |	900000000000517004 | Associated image |	| Annotated component |	900000000000461009 | Concept type component |	0
900000000000456007 | Reference set descriptor |	900000000000517004 | Associated image |	| Image |	900000000000469006 | URL |	1

Note that in the table above, the | URL | *concept* is a *descendant* of | *string* | *concept* in the metadata.

Note: The tables above omit the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the tables above also show the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.10.5 Annotation reference set example

As no annotation reference sets are included in the International Release, these sample rows are for illustration only.

Table 81: Example of "Associated image" Annotation Reference Set

refsetId	referencedComponentId	Annotation
900000000000517004 I Associated image I	80891009 I Heart structure I	http://en.wikipedia.org/wiki/Heart#mediaviewer/File:Wiki_Heart_Antomy_Ties_van_Brussel.jpg
900000000000517004 I Associated image I	86174004 I Laparoscope I	http://www.educationaldimensions.com/eLearn/endoscope/bigScope.html

In the above example, the two URLs have been used to annotate two *SNOMED CT concepts* with images on the web. It is not recommended that this mechanism be used to annotate *concepts* with text that may require translation to other *languages*. Instead, such text should be included under an appropriate *description* type within the *Description file*.

Note: The tables above omit the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the tables above also show the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.11 Association Reference Set

5.6.2.11.1 Purpose

An 900000000000521006 I Association type reference set I represents a set of unordered associations of a particular type between *components*.

5.6.2.11.2 Data structure

An *Association reference set* is a *Component reference set* used to represent associations between *component*. Its structure is shown in the following table.

Table 82: Association reference Set - Data structure

Field	Data type	Purpose
id	*UUID*	A 128 bit unsigned *integer*, uniquely identifying this *reference set member*. Different versions of a *reference set member* share the same *id* but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.
effectiveTime	*Time*	The inclusive date or time at which this version of the identified *reference set member* became the current version. The current version of this *reference set member* at time *T* is the version with the most recent *effectiveTime* prior to or equal to time *T*.

Field	Data type	Purpose
active	*Boolean*	The state of the identified *reference set member* as at the specified *effectiveTime*. If *active*= 1 (true) the *reference set member* is part of the current version of the set, if *active*= 0 (false) the *reference set member* is not part of the current version of the set.
moduleId	*SCTID*	Identifies the *SNOMED CT module* that contains this *reference set member* as at the specified *effectiveTime*. The value must be a *subtype* of 900000000000443000 | Module (core metadata concept) | within the metadata *hierarchy*.
refsetId	*SCTID*	Identifies the *reference set* to which this *reference set member* belongs. In this case, set to a *subtype* of 900000000000521006 | Association type reference set |. This indicates the nature of the association between the source and target *components*.
referencedComponentId	*SCTID*	The identifier of the source *component* of the association.
targetComponentId	*SCTID*	The identifier of the target *component* of the association.

5.6.2.11.3 Metadata

The following metadata in the "Foundation metadata *concept*" *hierarchy* supports this *reference set*:

- 900000000000455006 | Reference set |
  - 900000000000521006 | Association type |
    - 900000000000522004 | Historical association |
      - 900000000000523009 | POSSIBLY EQUIVALENT TO association reference set |
      - 900000000000524003 | MOVED TO association reference set |
      - 900000000000525002 | MOVED FROM association reference set |
      - 900000000000526001 | REPLACED BY association reference set |
      - 900000000000527005 | SAME AS association reference set |
      - 900000000000528000 | WAS A association reference set |
      - 900000000000529008 | SIMILAR TO association reference set |
      - 900000000000530003 | ALTERNATIVE association reference set |

- 900000000000531004 | REFERS TO concept association reference set |

Figure 52: Association Reference Sets in the Metadata Hierarchy

5.6.2.11.4 Notes on usage

Each member of a | Historical association | *reference set* represents a Reference from an *inactive component* to other equivalent or related *components* that were current in the *Release Version* in which that *component* was inactivated.

Each | Historical association | *reference set* holds *Relationships* of a different nature between the *components*. The |Historical association| *reference sets* contains associations:

- from each *inactive description* to one or more other *Descriptions* that are current in the *release Version* in which the *description* was inactivated;
- from each inactive *reference set* for which there is a current replacement to the replacement *reference set*;
- from an *inactive description* to a *concept* that is current in the *Release Version* in which the *description* was inactivated, and which is correctly described by the Term of the *inactive description*;
- From each *inactive concept* to one or more *concepts* that replace it.

The *component* identified by the *targetComponentId* must be an instance of the same class of *component* as the *component* identified by the referencedComponentIdfor all |Historical association| *reference sets* apart from the |REFERS TO *concept* association *reference set* |.

Within the |REFERS TO *concept* association *reference set* |, the referenced ComponentId field must be a *description* and the targetComponentIdmust be a *concept*.

The *targetComponentId* is used differently in the |MOVED TO association *reference set* |. In this case, the *targetComponentId* does not refer directly to a replacement *component* , but rather to the namespace to which the *component* was moved to. The *targetComponentId* actually refers to the *concept* that represents the namespace. This approach is used since the organization sourcing the *component* may not always be able to determine the precise reference that is applicable in the receiving organization (namespace). Thus the responsibility for these references lies with the new responsible (receiving) organization .

5.6.2.11.5 Descriptor template and examples

The tables below show examples of the descriptors for specific *reference sets* that follow the 900000000000521006 | Association type reference set | pattern.

Table 83: Refset Descriptor rows for 900000000000527005 | SAME AS association reference set |.

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 | Reference set descriptor |	900000000000527005 | SAME AS association reference set |	900000000000532006 | Association source component |	900000000000460005 | Component type |	0
900000000000456007 | Reference set descriptor |	900000000000527005 | SAME AS association reference set |	900000000000533001 | Association target component |	900000000000460005 | Component type |	1

Table 84: Refset Descriptor rows for 900000000000531004 | REFERS TO concept association reference set |.

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 | Reference set descriptor |	900000000000531004 | REFERS TO concept association reference set |	900000000000532006 | Association source component |	900000000000460005 | Component type |	0
900000000000456007 | Reference set descriptor |	900000000000531004 | REFERS TO concept association reference set |	900000000000533001 | Association target component |	900000000000460005 | Component type |	1

Note: The tables above omit the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the tables above also show the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.11.6 Association reference set example

The following table holds example entries for the | Replaced by | *reference set*.

Table 85: Sample content from 900000000000526001 | REPLACED BY association reference set |.

refsetId	referencedComponentId (Association source component)	targetComponentId (Association target component)
900000000000526001 | REPLACED BY association reference set |	100005 | SNOMED RT Concept |	138875005 | SNOMED CT Concept |
900000000000526001 | REPLACED BY association reference set |	212002 | Salmonella III arizonae 53:k:z |	398450001 | Salmonella IIIb 53:k:z |
900000000000526001 | REPLACED BY association reference set |	225005 | Special care of patient with contagious disease |	133895001 | Care of patient with infectious disease |
900000000000526001 | REPLACED BY association reference set |	244003 | Evans and Lloyd-Thomas syndrome |	66659007 | Normal variation in position |
900000000000526001 | REPLACED BY association reference set |	278009 | Epidural injection of neurolytic substance, lumbar |	17753007 | Epidural injection of neurolytic solution, lumbar |
900000000000526001 | REPLACED BY association reference set |	558000 | Other disorder of the neurohypophysis, NEC |	72442006 | Disorder of posterior pituitary |
900000000000526001 | REPLACED BY association reference set |	659001 | Peptostreptococcus anaerobius |	413524006 | Anaerococcus tretradius |
900000000000526001 | REPLACED BY association reference set |	696005 | Chronobiologic disorder |	387605007 | Abnormal chronobiologic state |
900000000000526001 | REPLACED BY association reference set |	700002 | Salmonella III arizonae 50:z4,z23,z32:-- |	404619004 | Salmonella IIIa 50:z4,z23,z32:- |
900000000000526001 | REPLACED BY association reference set |	822000 | Salmonella arizonae 53:z4,z23:-- |	13998005 | Salmonella IV 53:z4,z23:-- |

Note: The tables above omit the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the tables above also show the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.12 Module Dependency Reference Set

5.6.2.12.1 Purpose

The 900000000000534007 | Module dependency reference set | represents dependencies between different *SNOMED CT* release *modules*. In each case, the dependency indicates which *version* of each particular *module* a given *version* of the dependent *module* requires.

5.6.2.12.2 Data structure

The *Module dependency reference set* is a *String-String reference set* which is used to represent dependencies between *modules*, taking account of *module* versioning. Its structure is shown in the following table.

Table 86: Module dependency reference set - Data Structure

Field	Data type	Purpose
id	*UUID*	A 128 bit unsigned *integer*, uniquely identifying this *reference set member*. Different versions of a *reference set member* share the same *id* but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.
effectiveTime	*Time*	The inclusive date or time at which this version of the identified *reference set member* became the current version. The current version of this *reference set member* at time *T* is the version with the most recent *effectiveTime* prior to or equal to time *T*.
active	*Boolean*	The state of the identified *reference set member* as at the specified *effectiveTime*. If *active*= 1 (true) the *reference set member* is part of the current version of the set, if *active*= 0 (false) the *reference set member* is not part of the current version of the set.

Field	Data type	Purpose
moduleId	*SCTID*	Identifies the *SNOMED CT module* that contains this *reference set member* as at the specified *effectiveTime*. The value must be a *subtype* of 900000000000443000 | Module (core metadata concept) | within the metadata *hierarchy*. In this *reference set*, *moduleId* also refers to the dependent source *module*. Thus each *module* contains the rows of the 900000000000534007 | Module dependency reference set | that represent its dependencies on other *modules*.
refsetId	*SCTID*	Identifies the *reference set* to which this *reference set member* belongs. In this case, set to 900000000000534007 | Module dependency reference set |
referencedComponentId	*SCTID*	The identifier of a target *module* on which the dependent *module* (identified by *moduleId*) depends. Thus must be a *subtype* of 900000000000443000 | Module (core metadata concept) |.
sourceEffectiveTime	*Time*	The effective time of the dependent source *module* (identified by *moduleId*). This specifies a version of that *module*, consisting of all *components* that have the same *moduleId* as this *refset member* in their states as at the specified *targetEffectiveTime*.
targetEffectiveTime	*Time*	The effective time of the target *module* required to satisfy the dependency (identified by *referencedComponentId*). This specifies a version of that *module*, consisting of all *components* with the *moduleId* specified by *referencedComponentId* in their states as at the specified *targetEffectiveTime*.

5.6.2.12.3 Notes on usage

Module version dependencies are represented using a single 900000000000534007 | Module dependency reference set |. Thus all module dependency rows have the same refsetId (900000000000534007).

It is the responsibility of the organization owning and maintaining a dependent module to identify all modules on which it depends. They do this by adding rows to the 900000000000534007 | Module dependency reference set | within the dependent module. Because these added member must be in the dependent module, the *moduleId* of the *reference set* member record is also the identifier of the dependent (source) module. The target module on which the source module depends is identified by the *referencedComponentId*.

A module version may depend on one or more other module versions, and many module versions may have a dependency on a single module version. Cyclic module version dependencies are not allowed. If module-A depends on module-B, then module-B cannot depend on module-A.

Dependencies are not transitive and this means that dependencies cannot be inferred from a chain of dependencies. If module-A depends on module-B and module-B depends on module-C, the dependency of module-A on module-C must still be stated explicitly.

Any release should consist of a set of module versions that are certified as being compatible. Each release should also identify other module versions that it is dependent on even when these are outside the scope of the release. For example, the dependencies of modules in an Extension on the International Release must be stated.

Dependencies are specified between module versions, not just dependencies between modules. Therefore, it is possible to specify a dependency from a module released on one date to an earlier version of another module. The version of the dependent module is specified by the *sourceEffectiveTime* and the version of the module on which it depends is specified by the *targetEffectiveTime*.

Note: Current practice assumes the refset.id column contains the same identifier for all versions of the dependencies between the same pair of modules. This approach means that at any given time only one version of each module has effective dependencies. Therefore, to review the dependencies of an earlier version, a snapshot for an earlier time must be checked. An alternative approach has been suggested by some people in which a new identifier is allocated to each dependency of each module. This would then mean that all past dependencies would be visible in a snapshot view. It would also mean that it would be possible release updated dependencies for an existing module version while also releasing more up-to-date versions of the same module with different dependencies. This added flexibility comes at the price of additional complexity and for the time-being the *International Release* continues to use the simpler approach in which each new version of a dependency supersedes the dependency between earlier versions of the same pair of modules.

5.6.2.12.4 Metadata

The following metadata in the "Foundation metadata *concept*" *hierarchy* supports this *reference set*:

- 900000000000454005 | Foundation metadata concept |
  - 900000000000455006 | Reference set |
    - 900000000000534007 | Module dependency |

Figure 53: Module Dependency Reference Set in the Metadata Hierarchy

Each component within a *SNOMED CT release* references a *moduleId*. This is the *module* that the component is currently mastered in (from the *effectiveTime* held on the component record). A module is simply a collection of *SNOMED CT components* that are maintained as a unit by a single organization . It is the organization 's responsibility to organize the components in each *extension* that it is responsible for into one or more modules, in a way that best fits its business needs.

A module is modeled by a *descendant* of 900000000000443000 | Module | in the metadata *hierarchy*. The 900000000000443000 | Module | sub - *hierarchy* is organized by a maintainingorganization into a number of groups. For example, all modules maintained by *IHTSDO* will be *children* of 900000000000445007 | IHTSDO maintained module |. The 900000000000443000 | Module | *sub-hierarchy* models modules maintained by each organization and does NOT model module dependencies. Instead, module dependencies aremodeled using the 900000000000534007 | Module dependency reference set | .

At the point of release, if any *component* within a module has changed, then a new row will be added to 900000000000534007 | Module dependency reference set | for the module's *concept*, with the *effectiveTime* set to the date of the new release, irrespective of whether the other fields in the module *concept* record itself have changed. The updated |Module| *concept* record identifies that some components within the module have been updated in this release. Where no components within a module have been updated, then a new module record will not be added and the module's *effectiveTime* field will not change from the previous release.

Each *component* will be in one, and only one *component*. The module that a component is mastered in may change over time, and when this happens, the component's *moduleId* field will be updated (in the usual way by appending a row for the component).

Each module will be in one and only one *extension*. Modules will not straddle *extensions*. The *extension* that a module resides in is defined by the *SCTID* of the module. A module may not move from one *extension* to another over time. If the components within a module are to be moved to another *extension*, then a new module must be created within the destination *extension* to host the components that are to be transferred.

There may be more than one module in an *extension*.

5.6.2.12.5 Descriptor

The table below shows the descriptor that defines the structure of the 900000000000534007 | Module dependency reference set |.

Table 87: Refset Descriptor rows for 900000000000534007 | Module dependency |.

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 | Reference set descriptor |	900000000000534007 | Module dependency |	900000000000535008 | Dependency target |	900000000000461009 | Concept type component |	0
900000000000456007 | Reference set descriptor |	900000000000534007 | Module dependency |	900000000000536009 | Source effective time |	900000000000475002 | Time |	1
900000000000456007 | Reference set descriptor |	900000000000534007 | Module dependency |	900000000000537000 | Target effective time |	900000000000475002 | Time |	2

Note: The table above omits the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the table above also shows the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.12.6 Module dependency reference set example

The table below holds example entries for the 900000000000534007 | Module dependency reference set | in a *snapshot view* of the January 2014 *SNOMED CT International Release*.

This *SNOMED CT International Release* contains three modules:

- 900000000000012004 | SNOMED CT model component | which has no dependencies;
- 900000000000207008 | SNOMED CT core | which depends on the 900000000000012004 | SNOMED CT model component |; and
- 449080006 | SNOMED CT to ICD-10 rule-based mapping module | which depends on both the other modules.

In this case all the 2014-01-31 modules depend on 2014-01-31 versions of the other modules. However, in some case a module may depend on an earlier version of another model (e.g. an extension module may be releases after the *SNOMED CT International Release* to which it applies).

Dependencies are not transitive. The fact that 449080006 | SNOMED CT to ICD-10 rule-based mapping module | is dependent on 900000000000207008 | SNOMED CT core | may seem to imply a dependency on 900000000000012004 | SNOMED CT model component |. However, in practice all dependencies must be explicitly specified, not just immediate dependencies.

Table 88: Sample content from 900000000000534007 | Module dependency |.

moduleId	refsetId	referencedComponentId (Dependency target)	sourceEffectiveTime (Source effective time)	targetEffectiveTime (Target effective time)
900000000000207008 | SNOMED CT core |	900000000000534007 | Module dependency |	900000000000012004 | SNOMED CT model component |	20140131	20140131
449080006 | SNOMED CT to ICD-10 rule-based mapping module |	900000000000534007 | Module dependency |	900000000000012004 | SNOMED CT model component |	20140131	20140131
449080006 | SNOMED CT to ICD-10 rule-based mapping module |	900000000000534007 | Module dependency |	900000000000207008 | SNOMED CT core |	20140131	20140131

Note: The tables above omit the initial three columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*. Additionally, to aid understanding, the tables above also show the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.13 Description Format Reference Set

5.6.2.13.1 Purpose

The 900000000000538005 l Description format reference set l specifies the text format and maximum length of each supported *description type*. This permits additional *description types* to be specified in future in addition to the three existing *description types* (*synonym*, *fully specified name* and *textual definition*).

5.6.2.13.2 Data structure

The 900000000000538005 l Description format reference set l is a C-I (*component* - *integer*) *reference set* which is used to specify the length and format of the *terms* in *descriptions* of this *description type*. Its structure is shown in the following table.

Table 89: Description format reference set - Data structure

Field	Data type	Purpose
id	*UUID*	A 128 bit unsigned *integer*, uniquely identifying this *reference set member*. Different versions of a *reference set member* share the same *id* but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.
effectiveTime	*Time*	The inclusive date or time at which this version of the identified *reference set member* became the current version. The current version of this *reference set member* at time *T* is the version with the most recent *effectiveTime* prior to or equal to time *T*.
active	*Boolean*	The state of the identified *reference set member* as at the specified *effectiveTime*. If *active*= 1 (true) the *reference set member* is part of the current version of the set, if *active*= 0 (false) the *reference set member* is not part of the current version of the set.
moduleId	*SCTID*	Identifies the *SNOMED CT module* that contains this *reference set member* as at the specified *effectiveTime*. The value must be a *subtype* of 900000000000443000 l Module (core metadata concept) l within the metadata *hierarchy*.
refsetId	*SCTID*	Identifies the *reference set* to which this *reference set member* belongs. In this case, set to 900000000000538005 l Description format reference set (foundation metadata concept) l

Field	Data type	Purpose
referencedComponentId	*SCTID*	A reference to the *subtype* of 900000000000446008 | Description type (core metadata concept) | which specifies the *description type* to which this format applies.
descriptionFormat	*SCTID*	A reference to a *subtype* of 900000000000539002 | Description format (foundation metadata concept) | attribute which specifies the format of *terms* in *descriptions* of this *description type*.
descriptionLength	*Integer*	The maximum length in bytes of the *terms* in *descriptions* of this *description type*.

5.6.2.13.3 Metadata

The following metadata supports the *description* format *reference set*:

- 900000000000454005 | Foundation metadata concept |
    - 900000000000455006 | Reference set |
        - 900000000000538005 | Description format |

Figure 54: Description Format Reference Sets in the Metadata Hierarchy

5.6.2.13.4 Descriptor

The table below shows the descriptor that defines the structure of the 900000000000538005 I Description format reference set I

Table 90: Refset Descriptor rows for 900000000000538005 I Description format I.

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 I Reference set descriptor I	900000000000538005 I Description format I	900000000000462002 I Description type component I	900000000000461009 I Concept type component I	0
900000000000456007 I Reference set descriptor I	900000000000538005 I Description format I	900000000000539002 I Description format I	900000000000461009 I Concept type component I	1
900000000000456007 I Reference set descriptor I	900000000000538005 I Description format I	900000000000544009 I Description length I	900000000000478000 I Unsigned integer I	2

Note: The table above omits the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the table above also shows the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

5.6.2.13.5 Description format reference set example

This example holds the all the members of the 900000000000538005 I Description format reference set I in the *SNOMED CT International Release* for July 2014. Other members may added to future versions of the *International Release* if new *description types* are introduced. Owners of Extensions that support additional *description types* must also add members to the 900000000000538005 I Description format reference set I.

Table 91: Sample content from 900000000000538005 I Description format I.

refsetId	referencedComponentId (Description type component)	descriptionFormat (Description format)	descriptionLength (Description length)
900000000000538005 I Description format I	900000000000003001 I Fully specified name I	900000000000540000 I Plain text I	255
900000000000538005 I Description format I	900000000000013009 I Synonym I	900000000000540000 I Plain text I	255
900000000000538005 I Description format I	900000000000550004 I Definition I	900000000000540000 I Plain text I	4096

Note: The tables above omit the initial four columns of data present in the release file. These follow the standards versioning pattern *id*, *effectiveTime*, *active*, *moduleId*. Additionally, to aid understanding, the tables above also show the *term* from one of the *descriptions* associated with each of the identified *concept*. The release file only contains the *identifier*.

Chapter 6

6 Concept Model Guide

This part of the guide explains the *SNOMED CT Concept Model*. This is the model used to specify logical definitions of *SNOMED CT concepts*. It is based on a combination of formal logic and a set of editorial rules that determined the permitted sets of *attributes* and values that may applied to particular types of *concepts*.

6.1 Essential Features of the Concept Model

This section describes key features of the *Concept Model* that underpin the definitions of all *SNOMED CT concepts*.

6.1.1 Root and top-level Concepts

6.1.1.1 The Root Concept

The *Concept file* includes a special *concept* referred to as the *Root Concept* . It is the "root" of the main hierarchy that contain all the *Concepts* in *SNOMED CT*.

All other *Concepts* are descended from this "root" *concept* via at least one series of *Relationships* of the *Relationship Type* | is a | (i.e. all other *Concepts* are regarded as subclasses of this *Concept*).

The *Root Concept* Code is 138875005 and is named | SNOMED CT Concept |.

6.1.1.1.1 Features of the root Concept

All other *SNOMED CT Concepts* are *subtypes* of the root *concept*.

Unlike other *SNOMED CT Concepts*, the root *concept* is not a *subtype* of any other *concept*.

6.1.1.1.2 Release information in the root Concept

The root *Concept* has a current *Synonym* that contains information about the release. The *Synonyms*, representing earlier releases, are distributed as *Inactive Descriptions*. The release information is represented in the *term* text of the *Synonym* as indicated in *Table 92*.

Table 92: Representation of release information in the root Concept

Example	*SNOMED Clinical Terms* version: 20020131 [R] (first release)	
Stylized form	*SNOMED Clinical Terms* version: yyyymmdd [*status*] (*description*)	
	yyyymmdd	The release date in *ISO* format.
	Status	R (release), D (developmental) or E (evaluation).
	Description	An optional free text *description* of the release.

6.1.1.2 Top-level Concepts

Concepts that are directly related to the *Root Concept* by a single *Relationship* of the *Relationship Type* | is a | are referred to as "Top Level *Concepts*". All other *concepts* are descended from at least one Top Level *Concept* via at least one series of *Relationships* of the *Relationship Type* | is a | (i.e. all other *concepts* represent subclasses of the meaning of at least one Top Level *Concept*).

Many Top-level *Concepts* are intended to represent things outside of *SNOMED CT* (including processes, events, and material entities) in the real world. These include:

Table 93: Top Level Concepts

• *| Clinical finding |* • *| Procedure |* • *| Observable entity |* • *| Body structure |* • *| Organism |* • *| Substance |* • *| Pharmaceutical / biologic product |* • *| Specimen |* • *| Special concept |* • *| SNOMED CT Model Component |*	• *| Physical force |* • *| Event |* • *| Environment or geographical location |* • *| Social context |* • *| Situation with explicit context |* • *| Staging and scales |* • *| Physical object |* • *| Qualifier value |* • *| Record artifact |*

6.1.1.2.1 Representation of top-level Concepts

Awareness of the top-level *Concepts* is likely to be particularly important when developing technical implementations.

A top-level *Concept* can be identified by the fact that it has a single *subtype relationship* referring to the *Root Concept*. However, to minimize processing requirements the top-level *Concepts* have designated *Concept Identifiers* that are documented in this guide as *Important Concept Identifiers*.

6.1.1.3 Top Level Metadata Concepts

Metadata codes represent structural information about the terminology itself. The Top Level Metadata *Concepts* represent broad groups of metadata.

Table 94: Top Level Metadata

• | Core metadata concept | • | Foundation metadata concept | • | Linkage concept | • | Namespace concept |

6.1.2 Subtype Relationships

6.1.2.1 Role of subtype Relationships

Subtype Relationships provide the main semantic *hierarchy* that relates *Concepts* to one another.

All *Active Concepts*, except the root *Concept*, have *subtype Relationships* with one or more *Concepts*. Each of these *Relationships* indicates that a *Concept* is a *subtype* of another *Concept*.

6.1.2.2 Representation of Subtype Relationships

Subtype Relationships are expressed in the same way as all other *SNOMED CT Relationships*. They are identifiable by their RelationshipType, which refers to a *Concept* with the *Fully Specified Name* I is a I.

The *subtype Relationship Concept* has a designated *Concept Identifier* , which is documented in this guide as an *Important Concept Identifier*.

6.1.2.3 Subtype Relationships and the Subtype Hierarchy

Subtype Relationships represent the *subtype hierarchy* of *SNOMED CT*. This is illustrated here using a small sample set of *concepts* and *Relationships* listed in *Table 95*.[8]

Table 95: Subtype Relationships Example

Source	*Relationship Type*	Destination
I bacterial pneumonia I	I is a I	I infective pneumonia I
I bacterial pneumonia I	I is a I	I bacterial infectious disease I
I infective pneumonia I	I is a I	I infectious disease I
I infective pneumonia I	I is a I	I pneumonia I
I pneumonia I	I is a I	I disease of lung I
I disease of lung I	I is a I	I disease of respiratory system I
I disease of respiratory system I	I is a I	I disease I
I bacterial infectious disease I	I is a I	I infectious disease I
I infectious disease I	I is a I	I disease I
I disease I	I is a I	I SNOMED CT Concept I

Only the most proximate I is a I *relationships* are represented in the distribution files. These *Relationships* are shown by the blue lines in *Figure 55*. However, a *Concept* is a *subtype* of any *concept* to which it has a direct or indirect I is a I *Relationship*.

- Thus the *Concept* I bacterial pneumonia I is a *subtype* of all the other *concepts* shown in the diagram.

Example:

I Bacterial pneumonia I is a *subtype* of I pneumonia I because it is a *subtype* of I infective pneumonia I which is a *subtype* of I pneumonia. I

[8] Only a small sample of concepts and relationships have been included to produce a simple illustration. Some concept have been omitted and direct relationships have been included where in the release data the relationships pass via additional intermediate concepts.

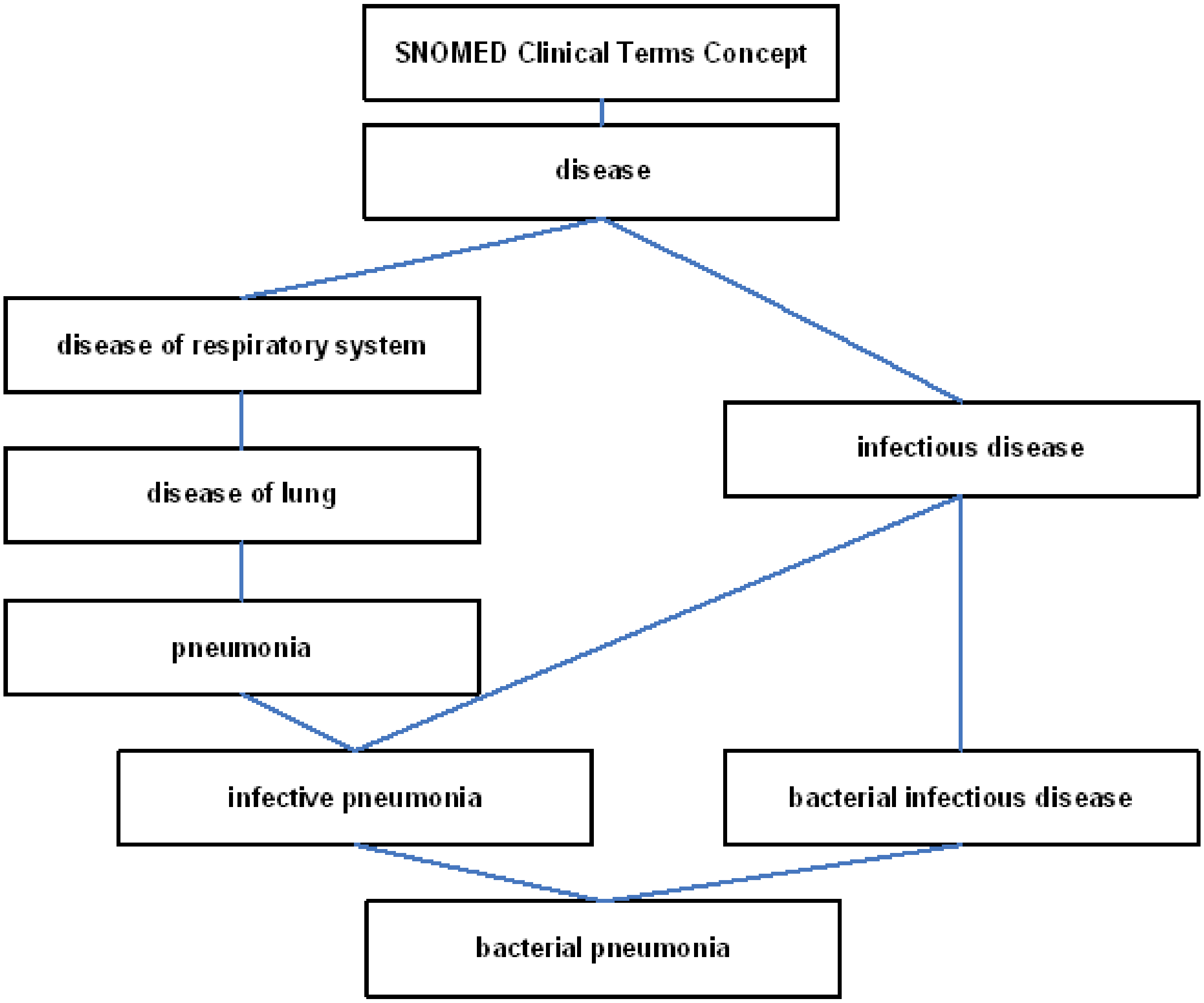


Figure 55: Graphical view of the of the Supertypes of | bacterial pneumonia |

The number of links in the chain of | is a | *Relationships* between two *Concepts* does not alter the logical meaning of the *relationship* between them. The number of | is a | *Relationships* between two *Concepts* may change between releases of *SNOMED CT* as a result of the addition of an intermediate *Concept*. This does not alter the semantic *relationship* between them.

Some technical implementation issues are affected by whether a pair of *Concepts* is linked by a single *subtype Relationship* or by a sequence of several *subtype Relationships*. In this guide, the following *terms* are used where this distinction is technically significant:

A given *Concept* (*Concept* -x) may have:

- *Subtype children* - *Concepts* with a *subtype Relationship* referring to *Concept*-x:
  - | bacterial pneumonia | is a *subtype child* of:
    - | bacterial infectious disease |
    - | infective pneumonia |
- *Supertype parents*\ - *Concepts* referred to by a *subtype Relationship* from *Concept*-x:
  - | infectious disease | is a *supertype parent* of:
    - | bacterial infectious disease |
    - | infective pneumonia |

- *Subtype descendants* - *Concepts* with *subtype Relationships* that refer to other *Concepts* that are either *child* or *subtype descendants* of *Concept*-x:
  - | bacterial pneumonia | is a *subtype descendant* of:
    - All other *Concepts* shown in the example.
- *Supertype ancestors* - *Concepts* referred to by *subtype Relationships* from other *Concepts* that are either parent or *supertype ancestors* of *Concept*-x:
  - | disease | is an *supertype ancestor* of:
    - All other *Concepts* shown in the example, except for | SNOMED CT Concept |.

- | bacterial pneumonia |
  - | bacterial pneumonia | | is a | | infective pneumonia |
    - | infective pneumonia | | is a | | infectious disease |
      - | infectious disease | | is a | | disease |
        - | disease | | is a | | SNOMED CT Concept |
    - | infective pneumonia | | is a | | pneumonia |
      - | pneumonia | | is a | | disease of lung |
        - | disease of lung | | is a | | disease of respiratory system |
          - | disease of respiratory system | | is a | | disease |
            - | disease | | is a | | SNOMED CT Concept |
  - | bacterial pneumonia | | is a | | bacterial infectious disease |
    - | bacterial infectious disease | | is a | | infectious disease |
      - | infectious disease | | is a | | disease |
        - | disease | | is a | | SNOMED CT Concept |

Figure 56: Inverted hierarchical view of the *Supertypes of | bacterial pneumonia |*

6.1.3 Defining characteristics

6.1.3.1 Role of defining characteristics

Subtype relationships contribute the hierarchical type based aspect of a *Concept* definition. This is augmented by *defining characteristics* that represent the values of a range of relevant attributes. Depending on the nature of the *concept* these may include including etiology, topography, method, etc.

The range of attributes applicable depends on the type of *Concept*. For example, a procedure may have a method, and a disorder may have an etiology, but a procedure cannot have an etiology, and disorder cannot have a method.

Defining characteristics using a particular attribute will be applied consistently to all *Concepts* to which it is relevant. Note that this design principle may not be fully realized for all attributes in each release.

6.1.3.2 Representation of defining characteristics

Defining characteristics are represented as *Relationships*. The fields are used as follows:

- SourceId refers to the *Concept* to which a *defining characteristic* applies;
- *TypeId* indicates the nature of the defining attribute;
- DestinationId refers to the *Concept* that represents the value of that attribute.

In each release the supported *defining characteristics* for every *Concept* are distributed in the *relationship file*. The supported *defining characteristics* are *descendants* of the *concept* 410662002 I concept model attribute I. The list of supported defining attributes is provided in *Defining Attributes by Hierarchy and Domain*[9].

Table 96: Defining characteristics applied to an example concept

I Disease I		
	I is a I ***SNOMED CT Concept***	
	Primitive	Not all *SNOMED CT Concepts* are diseases. No *defining characteristics* are included to specify what makes something a disease.
I infectious disease I		
	I is a I I disease I	
	I causative agent I I infectious agent I	
	Primitive	Not all diseases with causative agent I infectious agent I are I bacterial infectious disease I. For example, I rheumatic heart disease I has I causative agent I I streptococcus I but is not an I infectious disease I.
I bacterial infectious disease I		
	I is a I I infectious disease I	
	I causative agent I I bacteria I	
	fully defined	All infectious diseases with causative agent I bacteria I are I Bacterial infectious disease I
disease of respiratory system		
	I is a I I disease I	
	I finding site I I respiratory system structure I	

9 Note that the Relationships shown in the table and diagram are not the definitive released Relationships of these Concepts. They have been simplified to illustrate particular points in the text.

	fully defined	All diseases with | finding site | | respiratory system structure | are | Disorder of respiratory system |.
disease of lung		
	| is a | **disease of respiratory system**	
	finding site | lung structure |	
	fully defined	All diseases of respiratory system with | finding site | | lung | are | Disorder of lung |.
| Pneumonia |		
	| is a | **disease of lung**	
	finding site | lung structure |	
	Primitive	Not all diseases of lung are | pneumonia |. No additional characteristics specify what attributes are needed to specify | pneumonia |
| infective pneumonia |		
	| is a | | infectious disease |	
	| is a | | pneumonia |	
	| causative agent | | infectious agent |	
	finding site | lung structure |	
	fully defined	All pneumonias with causative agent | infectious agent | are | infective pneumonia |.
| bacterial pneumonia |		
	| is a | | bacterial infectious disease |	
	| is a | | infective pneumonia |	
	| causative agent | | bacteria |	
	finding site | lung structure |	
	fully defined	All pneumonias with causative agent | bacteria | are | bacterial pneumonia |.

6.1.4 Qualifiers and refinement

6.1.4.1 Qualifiers and refinable definitions

A *qualifying characteristic* is an attribute that may have one of several possible values for a particular *Concept*. If a particular *qualifier* is applied to a *Concept*, the resulting *expression* represents a more tightly defined *subtype* of that *Concept*.

Example: It might be possible to qualify a disorder such as | bacterial pneumonia | according to its clinical course (| acute | or | chronic |) or severity ("mild," "moderate" or | severe |). With appropriate *qualifiers*, "injury of skin of the left side of face" could then be represented even if a single *Concept Identifier* cannot express this.

A similar tightening of the definition of a *Concept* can be achieved by allowing one or more of the *defining characteristics* associated with a *Concept* to be refined. A *defining characteristic* is refined by an *expression* that applies a specified *subtype* of the value stated in the definition.

Example: | Fracture of bone | could be refined by qualifying it with the finding site"tibia" to represent the *Concept* | Fracture of tibia |.

6.1.5 Primitive and fully-defined Concepts

A *Concept* is considered to be *fully defined* if its *defining characteristics* are sufficient to define it relative to its immediate supertype(s). A *Concept* which is not *fully defined* is *Primitive* and this is indicated by the value of the *definitionStatusId* field.

Example: | Pneumonia | is a lung disease but unless *defining characteristics* are specified that effectively distinguish | pneumonia | from other lung diseases then it is regarded as a *primitive Concept*.

If a *Concept* is *primitive* then the *defining characteristics* for that *Concept* are incomplete. It is not possible to automatically compute that a *Concept* represented as a *postcoordinated* combination of several *Concepts* is or is not a *subtype* of a particular *primitive Concept*.

Example: The *Concept* "lung disease" qualified by | causative agent | = | bacteria | may be | pneumonia | but could also be "bronchitis."

In contrast if a *Concept* is *fully defined* it is possible to state that any *Concept* represented as a combination of the same *defining characteristics* is equivalent to or a *subtype* of that *Concept*.

Example: Assume that the *Concept* | bacterial pneumonia | is *fully defined* as | infective pneumonia | with | causative agent | = | bacteria | and that | pneumococcus | is a | bacteria |. It then follows that the post coordinated representation of | pneumococcal pneumonia | as | infective pneumonia | with | causative agent | = | pneumococcus | is computably a *subtype* of | bacterial pneumonia. |

6.1.6 Important Concept Identifiers

Table 97: Root Concept and Subtype Relationship

Id	*Preferred Term*	Comments
138875005	| SNOMED CT Concept |	All *Active Concepts* are *subtype descendants* of this *Root Concept*. The *Root Concept* has a *current Synonym* representing the release date.

Id	*Preferred Term*	Comments
116680003	| is a |	Relates a *Concept* to its immediate supertype *Concepts*.

Table 98: Top-Level Concepts

Id	*Preferred Term*	
123037004	| body structure |	
404684003	| clinical finding |	
308916002	| environment or geographical location |	
272379006	| event |	
363787002	| observable entity |	
410607006	| organism |	
373873005	| pharmaceutical / biologic product |	
78621006	| physical force |	
260787004	| physical object |	
71388002	| procedure |	
362981000	| qualifier value |	
419891008	| record artefact |	
243796009	| situation with explicit context |	
48176007	| social context |	
123038009	| specimen |	
254291000	| staging and scales |	
105590001	| substance |	
106237007	| linkage concept |	In *Release Format 2* this is a *subtype* of | SNOMED CT Model Component |.

Id	***Preferred Term***	
370115009	| special concept |	Not used in *Release Format 2*. Replaced and extended by | SNOMED CT Model Component |.
900000000000441003	| SNOMED CT Model Component |	Introduced in *Release Format 2*.

Table 99: Special Concepts

Id	***Preferred Term***	Comments
370115009	| special concept |	A top-level *Concept* that has as its immediate *subtypes* a set of *Concepts* that are used to support the functionality of the terminology rather than to represent real-world *Concepts*.
363743006	| navigational concept |	A Special *Concept* that has as its immediate *subtypes* all *active Navigation Concepts*.

Table 100: Valid Relationship Type Concepts - Defining Characteristics

Id	***Preferred Term***	Comment
246061005	| attribute |	
410662002	| concept model attribute |	
260507000	| access |	
246090004	| associated finding |	
116676008	| associated morphology |	
363589002	| associated procedure |	
47429007	| associated with |	
255234002	| after |	*Subtype* of | associated with |.
246075003	| causative agent |	*Subtype* of | associated with |.
42752001	| due to |	*Subtype* of | associated with |.
263502005	| clinical course |	
246093002	| component |	

Id	*Preferred Term*	Comment
363701004	| direct substance |	
246456000	| episodicity |	
408729009	| finding context |	
419066007	| finding informer |	
418775008	| finding method |	
363698007	| finding site |	
127489000	| has active ingredient |	
363705008	| has definitional manifestation |	
411116001	| has dose form |	
363702006	| has focus |	
363703001	| has intent |	
363713009	| has interpretation |	
116686009	| has specimen |	
363714003	| interprets |	
272741003	| laterality |	
370129005	| measurement method |	
260686004	| method |	
246454002	| occurrence |	
123005000	| part of |	
370135005	| pathological process |	
260870009	| priority |	
408730004	| procedure context |	

Id	*Preferred Term*	Comment
405815000	| procedure device |	
363699004	| direct device |	*Subtype* of | procedure device |.
363710007	| indirect device |	*Subtype* of | procedure device |.
424226004	| using device |	*Subtype* of | procedure device |.
425391005	| using access device |	*Subtype* of | using access device |.
405816004	| procedure morphology |	
363700003	| direct morphology |	*Subtype* of | procedure morphology |.
363709002	| indirect morphology |	*Subtype* of | procedure morphology |.
363704007	| procedure site |	
405813007	| procedure site - Direct |	*Subtype* of | procedure site |.
405814001	| procedure site - Indirect |	*Subtype* of | procedure site |.
370130000	| property |	
370131001	| recipient category |	
246513007	| revision status |	
410675002	| route of administration |	
370132008	| scale type |	
246112005	| severity |	
118171006	| specimen procedure |	
118170007	| specimen source identity |	
118168003	| specimen source morphology |	
118169006	| specimen source topography |	
370133003	| specimen substance |	

Id	*Preferred Term*	Comment
131195008	| subject of information |	
408732007	| subject relationship context |	
424876005	| surgical approach |	
408731000	| temporal context |	
370134009	| time aspect |	
424244007	| using energy |	
424361007	| using substance |	

6.2 Concept Model Specification

This section specifies the main hierarchies of the *SNOMED CT Concept Model* and the attributes and values used to define *concepts* in particular hierarchies.

6.2.1 Scope and boundaries

The scope of *SNOMED CT* has been driven by its historical legacy and by the perceived requirements of current user communities. Scope can be defined separately for three dimensions: 1) domain coverage or breadth, 2) granularity or depth, and 3) knowledge representation.

6.2.1.1 Scope of domain coverage

The terminology domains covered by *SNOMED* include:

- Clinical findings, including disorders
- Procedures, broadly defined as including all health related activities such as history taking, physical examination, testing, imaging, surgical procedures, disease-specific training and education, counseling, and so forth.
- Observable entities which, when given a value, provide a specific finding or assertion about health related information. Examples include the names of lab tests, physical exam tests, dates of significant events, and so forth.
- Anatomy, morphology, and other body structures
- Chemicals and other substances of relevance to health and health care, including generic drug ingredient names, generic drug products (virtual medicinal products)
- Generic physical devices relevant to health care, or to broad categories of injury or accident
- Organisms relevant to health and health care of humans and animals
- Other etiologies of disease, including external forces, harmful events, accidents, genetic abnormalities,
- Functions and activities
- Social contexts relevant to health, including general categories of status of employment, education, housing, care provision, family *Relationships*, and so forth.

- Types of clinical records, documents, certificates and other records and record components relevant to health care.
- Staging, scales, classifications, and other miscellaneous health information
- Attributes and values necessary to organize and structure the terminology

Note: Non-human *concepts*: In versions of *SNOMED CT* prior to 2014, there were many strictly non-human *concepts*. These have now been moved to an extension. Many codes are applicable to both human and non-human subjects, and of course these have not been moved.

6.2.1.2 Scope of granularity

Terms and meanings in the terminology can be characterized by a point on a scale from very general to very specific. The degree to which the terminology includes highly specific terms is often referred to as granularity. More properly, terms can be said to be at a level that is coarsely granular, or at a level that is finely granular.

At upper levels, *SNOMED CT* accepts coarsely granular meanings that are useful only for aggregation and are not useful for individual patient data recording. Examples include clinical finding, procedure, measurement procedure, etc. Progressive levels of refinement are allowed to the extent required to meet clinical data requirements. There are, however, limits to the degree of *precoordination* of certain types of complex statements. There is a general rule that the terms in *SNOMED CT* should name things that exist in the real world, and therefore they should tend to be names or short noun phrases, not complete sentences and certainly not paragraphs. The terminology originated as a nomenclature, and it is intended to be used in concert with an information model that can carry full clinical statements along with their attribution, dates, times, and statement inter- *Relationships*. There is an evolving understanding of the boundary between items named in the terminology and more complex statements that should be represented as combined terminology-information elements.

More detailed advice and guidance can be found in the section "Content Inclusion: Principles and Process" below.

6.2.1.3 Scope of knowledge representation

SNOMED logic-based definitions represent terminological knowledge. In other words, they represent what is always necessarily true about the meaning of a code. The logic definitions are not intended to cover the entire range of medical knowledge, and are not intended to include probabilistic or uncertain knowledge. Such knowledge is beyond the scope of *SNOMED CT*'s logic definitions.

For example, consider myocardial infarction (MI). terminological knowledge about this entity includes the fact that it must involve the myocardium, and it must involve an infarction. Additional knowledge that is not terminological, and therefore not included in the *SNOMED* logic definition, might include the fact that an MI is usually associated with crushing substernal chest pain, diaphoresis, arrhythmia, ST-segment elevation on EKG, and elevated levels of cardiac enzymes. Not every case of MI will have chest pain, nor will every case show ST segment elevation, etc. While these are valuable clues to the diagnostician, they are not necessarily always present, and therefore they are not part of the terminological knowledge base. As another example, consider appendicitis. terminological knowledge about this entity includes the fact that it is a kind of inflammatory disorder, and that it involves the appendix. Additional knowledge that is not terminological might include the fact that it often involves central abdominal pain that migrates to the right lower quadrant, and that it is associated with anorexia, nausea, elevated white blood count, and rebound tenderness over McBurneys point. These additional pieces of knowledge are variably present and therefore represent uncertain or probabilistic knowledge. Such variable or probabilistic knowledge is highly valuable for decision support algorithms, but is beyond the scope of *SNOMED CT*'s logic definitions.

6.2.1.4 Content Inclusion Principles and Process

6.2.1.4.1 Content Inclusion - Problem Statement

The basic problem to be addressed is that of deciding whether new content should be added to *SNOMED CT*, and if so, should it be in the *international release* or in an extension. This document attempts to provide basic principles and specific guidelines and examples, as well as a process that can be followed to resolve difficult or contested decisions.

6.2.1.4.1.1 Identifying Acceptable and Unacceptable Content

The range of *concepts*, terms, and other components in *SNOMED CT* is extremely broad in order to support the terminological needs of information systems that support the health and health care of individuals. Nevertheless, within this broad scope there are items for which some groups or individuals may want to have a "code", but which are ruled as unacceptable for *SNOMED CT*. This document aims to identify some rules and mechanisms by which unacceptable content can be identified.

6.2.1.4.1.2 What is the "core"?

The word "core" has been used with several different meanings and therefore should not be used without qualification. In this document we will address the question of whether content should be included in *SNOMED* at all, and also the question of whether it should be in the "core" *international release*. This is the content that is maintained and distributed by *IHTSDO*.

6.2.1.4.1.3 What is an extension?

A *SNOMED CT Extension* is a set of terminology *components* and *derivatives* that add to and are dependent on the *SNOMED CT International Edition*, and are created, structured, maintained and distributed in accordance with *SNOMED CT* specifications and guidelines.

All *extensions* are dependent on the *SNOMED CT International Edition* and can be used to broaden the scope of coverage and/or to configure the terminology for use in a specific language, specialty or jurisdiction. *IHTSDO Members* may create, maintain and distribute *extensions* to address specific national, regional and language requirements. *IHTSDO Affiliates* may also create, maintain and distribute *extensions* to meet the needs of particular software solutions and customers.

Each *component* created as part of an *extension* had an unique identifier (*SCTID*) that includes the *namespace identifier* assigned, by *IHTSDO*, to the organization responsible for that *extension*. The organization responsible for an *extension* is required to create, maintain and distribute *components* and *derivatives* in accordance with *IHTSDO* specifications and guidance. This ensures that *extensions* are compatible with the structure and content of the *SNOMED CT International Edition*.

Namespace identifiers are allocated in response to requests from *IHTSDO Members* and *Affiliates*. For further information about this process and for access to the current SNOMED CT Namespace Register please refer to the *IHTSDO web page on Namespaces*.

It is important to emphasize that the *IHTSDO* definition of *extension* is a narrow one. Other types of add-ons, enhancements or expansions that may of increasing the usefulness of *SNOMED CT* in various application domains are not regarded as *SNOMED CT Extensions*.

Notes:

1. *Components* that are created in an *extension* are identified using *extension SCTIDs*. These identifiers include an *extension namespace* which ensures that they do not collide with other *SCTIDs*, and can be traced to an authorized originator.
2. *Namespace identifiers* are allocated in response to requests from *IHTSDO Members* and *Affiliates*. For further information about this process and for access to the current SNOMED CT Namespace Register please refer to the *IHTSDO web page on Namespaces*.
3. *IHTSDO Members* may create, maintain and distribute *extensions* to address specific national, regional and language requirements. *IHTSDO Affiliates* may also create, maintain and distribute *extensions* to meet the needs of particular software solutions and customers.
4. See also *Edition* which refers to the combination of an *extension* with the *International Release* and, where relevant, any modules from other *extensions*on which it depends.

6.2.1.4.2 Basic principles: Does it belong in SNOMED CT?

Not every possible term or code that is related to health care belongs in *SNOMED*. Some content should be excluded. There has been significant debate about what does and does not belong; these debates are healthy and should continue. This document only attempts to ground the debate in some specific principles that can guide the decisions about what to include and what to exclude.

SNOMED CT is intended to be as reusable as possible, and this generates a tension between being all things to all purposes, versus being custom-fit to a particular purpose. The most general statement that can be made is that *SNOMED CT* is designed to foster semantic interoperability of electronic health applications.

6.2.1.4.2.1 Creation and maintenance of semantic interoperability

The most basic principle for determining whether content belongs in *SNOMED CT* is that it must create and sustain semantic interoperability of clinical information. This ability in turn depends on a reproducible and consistent approach to the incorporation of terminology into electronic records.

6.2.1.4.2.1.1 URU - Understandable, Reproducible, Useful

Beginning in 1996 with the development of *SNOMED RT*, the *SNOMED* modelers began to follow three basic operational criteria that help determine whether new content is following the principle of creating and sustaining semantic interoperability. These tests were summarized with the acronym "URU", standing for:

- **Understandable**: The meaning must be able to be communicated to and understood by an average health care provider without reference to inaccessible, hidden or private meanings
- **Reproducible**: It is not enough for one individual to say they think they understand a meaning. It must be shown that multiple people understand and use the meaning in the same way.
- **Useful**: The meaning must have some demonstrable use or applicability to health or health care.

6.2.1.4.2.2 Coordination with and exposure of information architecture components

The overall semantic interoperability of *electronic health records* is derived from the combined functioning of an information architecture and the terminology that populates it. Yet *SNOMED CT* itself does not produce – and does not dictate the choice of – the information architecture. The best we can do is to make explicit those elements of *SNOMED CT* that would be possible to represent in some alternative ways using the information architecture.

6.2.1.4.2.2.1 SNOMED CT Codes name classes of things

SNOMED CT is a systematic way of naming concepts that uses codes to represent classes or categories of real things. The terms should be names that are human-understandable representations of the codes. With reference to the semiotic triangle, the codes should be considered symbols that refer to classes or categories of real things. New content should be rejected if it consists of full sentences and statements rather than names that can be used in statements.

New content that refers to, or contains references to, a particular instance, should also be rejected. For example, "Doctor Jones' pre-operative order set" can be rejected on the grounds that Doctor Jones is an instance (individual) and not a class.

6.2.1.4.2.2.2 Clinical statements are made using codes within the information architecture

There is a tension between the purity of a nomenclature and the needs of information system implementers. Many system implementers are working with impoverished information models, and to deny them certain coded content is to prevent their use of *SNOMED CT* at all. In the past we have acknowledged this fact and attempted to maximize usefulness while minimizing the *precoordination* of all possible clinical statements.

6.2.1.4.2.3 Comprehensiveness of domain coverage

Within the content currently in *SNOMED CT*, there are some areas that are covered with a great deal of completeness, and others that are not. It is a goal for *SNOMED CT*, and one of the main features of any good clinical terminology, to be comprehensive in those areas that it chooses to cover.

Decisions about inclusion of an individual piece of content should therefore be made on the basis of whether the domain to which it belongs is one that is being comprehensively supported in *SNOMED CT*. For example "organisms affecting human and veterinary health" would be an example of a domain which has previously been included in the core. An organism meeting this criterion can therefore be judged as belonging.

A list of domains currently being maintained is included in *Scope of domain coverage* on page 230.

Even within a particular broad domain, there may be sub-domains that should be added comprehensively rather than piecemeal. For example, there are many new genetic tests not currently included in *SNOMED CT*. The general approach to addition should be to add these as a large batch that is quality assured and reviewed for overlaps and gaps and inconsistencies. They should not be added one or two at a time, because this would be inefficient and error-prone and would not serve users' needs well.

6.2.1.4.3 Principles for accepting content into the international release

The fundamental statement of the scope of the *international release* is that it includes content necessary for international conformance and interoperability.

Content that is defined as being within the scope of the *international release* is restricted to the *international release* and may not be modified or replaced by an extension, unless explicitly permitted by *IHTSDO*. For content that is within scope for *SNOMED CT*, other criteria must be met in order to require that it must be included in the *international release*:

Is the *concept* necessary for health information conformance and interoperability?

Multi-national – Is it useful in more than one national healthcare system?

Conformance – Does it need to be understandable in health information systems within more than one national healthcare system?

Interoperable – Does it need to be shared so that information systems can use it in a reproducible manner beyond a patient's national healthcare system, if a patient were to travel or relocate to a different country?

6.2.1.4.3.1 Use of Proprietary Names and Works

1. Introduction

   This section considers scope issues arising from the incorporation into SNOMED CT of proprietary names (such as names of clinical forms and drugs) and content from clinical forms and tools. The section is divided into two sub-sections. Subsection 2 deals with considerations for the IHTSDO itself. Subsection 3 deals with considerations for third parties (such as IHTSDO Affiliates) who implement SNOMED CT in clinical systems. There is a degree of overlap between these sections, but the issues are not identical. In this section, we refer to the 'owner' of a clinical form or tool. This term is used loosely to refer to whichever person or organization owns whatever intellectual property rights exist in the form or tool. This may be the individual or group of individuals who originally created the form or tool, or it may be the healthcare organization that employed them at the time of creation. We are also aware of cases in which the original creators of a clinical form have assigned their intellectual property rights to a commercial organization which now administers the licensing of the form to organizations who wish to use it.

2. Issues for the IHTSDO

   a. Incorporation of names

      The mere incorporation into SNOMED CT of the name of a clinical form or tool (e.g. the 'XYZ Test'), or the name of the score generated by a form or tool (e.g. the 'XYZ Test Score') will not require a license from the owner. It is possible that the owner holds a trade mark (which may be registered or unregistered) representing the name or score, but simply incorporating that word into SNOMED CT is not an act that would infringe the trade mark.

      This also applies to 'brand name' drugs (as opposed to generic drugs). The IHTSDO does not need to obtain the permission of the trade mark owner simply to include a reference to the drug in SNOMED CT.

   b. Incorporation of questions

A clinical form or tool, including the wording of the individual questions within the form or tool, will generally be a literary work and will qualify for copyright protection. The copying of all or any substantial part of a literary work, without a license from the owner, will infringe the owner's copyright.

It is possible, though unlikely, that the incorporation of the wording of an individual question from a clinical form or tool may infringe the owner's copyright. However, we understand that it is much more likely that the IHTSDO would need to systematically include all of the questions from the form or tool in SNOMED CT. Except in the case of the simplest of forms, that is likely to infringe the owner's copyright without a license from the owner.

c. Incorporation of answers

Certain questions may have a range of pre-determined answers. This could be as simple as 'yes / no' or a number within a specific range (e.g. 0, 5 or 10), but may also be more substantial text (e.g. 'needs help cutting', spreading butter, etc., or 'requires modified diet').

Clearly, incorporating very simple answers into SNOMED CT (such as numbers or 'yes' / 'no') will not require the permission of the owner. However, incorporating more substantial text into SNOMED CT will generally infringe the owner's copyright. This will usually not apply to individual answers, but it will almost always be the case where entire sets of answers (e.g. all possible answers to a question) are incorporated.

d. Incorporation of scores

The principles that apply to individual answers also apply to the overall score generated by a clinical form or tool.

The incorporation of mere numbers will not infringe the owner's copyright. However, in cases where each possible score has an associated textual description and all the possible scores are incorporated into SNOMED CT, together with their associated descriptions, a license would be required from the owner.

e. Incorporation of ‘concepts’ representing questions, answers or scores

We have considered the possibility that, instead of the text of questions, answers or scores being incorporated into SNOMED CT, a concept may be introduced that represents one or all of these.

For example, a form may include a question about a person's ability to dress and a range of possible answers. The IHTSDO might want to incorporate neither the text of the question nor any of the possible answers into SNOMED CT, but instead might want to incorporate a single concept such as 'ability to dress'. Similarly, if the form contains 20 questions, the IHTSDO might want to introduce 20 concepts into SNOMED CT, for 'XYZTest_Result1', 'XYZTest_Result2' and so on, up to 'XYZTest_Result20'.

As with the questions, answers and scores, the incorporation of a single concept into SNOMED CT based on a question, answer or score on a clinical form is highly unlikely to infringe the owner's copyright. However, if the IHTSDO were systematically to introduce a concept into SNOMED CT for every single question on a clinical form, that is likely to amount to copying a substantial amount of the work and would infringe the owner's copyright. The fact that there may be no verbatim or 'literal' copying from the work does not prevent a court from holding that there has been substantial taking from the work, and that there has therefore been an infringement of copyright.

We have also considered the possibility that these concepts (e.g. 'ability to dress') may already exist within SNOMED CT, or be added to it, because they arise in other contexts. (Obviously, this does not apply to concepts that represent specific questions within a form.) Copying is an essential element of any action for breach of copyright – if the owner cannot show that the concepts exist in SNOMED CT because they have been copied (whether or not literally) from the owner's work, the owner will not succeed in an infringement action against the IHTSDO.

3. Issues for Implementers of SNOMED CT

a. Incorporation of names

The mere use in a clinical system of the name of a clinical form or tool, or the name of a 'brand name' drug, will usually not amount to trade mark infringement. We say “usually” here because there is a very wide variety of systems in which SNOMED CT may be implemented, and particular caution should be exercised by implementers who wish to use trade marks in a 'commercial' context (such as systems that enable drugs to be purchased electronically). The IHTSDO avoids giving any specific advice to system implementers on this matter, and advises that implementers who are in any doubt should be encouraged to contact the trade mark owner to discuss whether they require a license.

In any event, system implementers would be advised to make no greater use of a trade mark than is necessary. For example, while displaying the text of a trade mark may not amount to an infringement, it would be preferable to avoid displaying any associated graphical mark (such as a logo) in screen displays or printed output.

b. Incorporation of questions, answers and scores

The considerations that apply to the IHTSDO's incorporation of questions, answers and scores into SNOMED CT also apply to system implementers. In cases where the incorporation of content from a clinical form or tool infringes a third party's copyright, that copyright will also usually be infringed by a system that reproduces that content (such as on a screen display or printed output). This means that, if the IHTSDO requires a license to incorporate the content, that license should ideally also cover use by system implementers.

There are two additional issues for implementers of SNOMED CT.

First, as noted above, it is possible that terms that are found in a clinical form may already exist within SNOMED CT, even though they have not been copied from the form in any way. As noted above, this would not amount to copyright infringement by the IHTSDO, since there would be no copying. However, if a system implementer chooses to arrange a collection of these pre-existing terms in a way that then reproduces all or a substantial part of a clinical form (for example, by populating a drop-down box with all the possible answers to a specific question that appears on the form), that may infringe the owner's copyright, even though those terms existed separately within the SNOMED CT database tables.

Secondly, a system may reproduce the structure and layout of a clinical form on a screen display or printed output (for example, to make the system more accessible to users who are familiar with a paper-based form). This may well infringe the owner's copyright in the form, unless the layout is very trivial (e.g. a bullet-point list). Certainly, any implementer who wishes to emulate the 'look and feel' of a clinical form within its system would be advised to seek a license from the relevant owner.

c. Incorporation of algorithms or ‘logic’

Another issue that would apply specifically to system implementers is the use of the algorithms or 'logic' inherent in a clinical form or tool, such as the method by which an overall score is calculated. For example, a clinical form may instruct the user to perform a mathematical operation on the individual answers to produce the overall score, and the same operation may be carried out by the system.

Of course, since the infringement would result from the use of the algorithm, the fact that terms from the underlying form may also be represented in SNOMED CT would be merely incidental. For this reason, the IHTSDO avoids becoming involved in any such issues, and encourages system implementers to contact the relevant owner if they are in any doubt as to whether the use of an algorithm may infringe the owner's rights.

6.2.1.4.3.2 Management of Non-Human Content

Non-human content may be included in a request for new content via the SNOMED International Request System (SIRS) or may be identified in the international release. Due diligence is required to differentiate content that belongs in the "core", meaning the International Release of SNOMED CT, versus content that can be handled in an extension. The basic principle is that content that may be useful in human medicine belongs in the core. Content that is strictly non-human may be managed in an extension. Examples of non-human content include:

- Egg related coelomitis (disorder)

- Dehorning (procedure)
- Bone structure of wing (body structure)

Criteria for individual types of content that should be in the core include the following:

- Diseases and findings: Anything that can occur in both humans and animals should be in the core.
- Material entities: Every substance that can cause adverse effects should be in the core, with the understanding that poisonings and adverse effects can occur in humans caused by virtually any substance regardless of its intended purpose or origin. As the substance hierarchy begins to differentiate more clearly between molecular entities, collectives, mixture and other material entities, it may be that some material entities will be of interest only in a non-human or veterinary context, and these could be added to or left in a veterinary extension.
- Organisms: Most organisms should be in the core, but there may be exceptions. There are over 20,000 organism codes in the Veterinary Extension maintained by the Veterinary Terminology Services Laboratory (VTSL) at Virginia Tech. As a general rule, these do not need to be transferred into the core unless they are of value in public health or human medicine, or have been requested to be put in the core by more than one IHTSDO Member Country.
    - Requests from veterinary users for organisms that have no known use case in human medicine can be met by adding them to the organism extension content managed by VTSL.

The veterinary extension is publicly available to those in IHTSDO Member Countries and those who hold an Affiliate License. To obtain more information about and access to the veterinay extension, visit http://vtsl.vetmed.vet.edu or contact VTSL at vtsl.extension@gmail.com

6.2.1.4.3.3 Principles for determining National Extension content

The fundamental statement of the scope of a National *Extension* is that it includes content outside of the scope of the *international release*, but is necessary for national conformance and interoperability. The interpretation and application of this fundamental scope will depend on the specifics of each member-state's national healthcare system and is left for each member-state to determine for itself.

Criteria to determine if *concepts* should be included in the National *Extension* terminology include:

Is the *concept* outside the scope of the *international release* but necessary for national conformance and interoperability?

National – Is it useful throughout the national healthcare system?

Conformance – Does it need to be understandable throughout the national healthcare system?

Interoperable – Does it need to be shared in a reproducible manner within the national healthcare system?

If so, then the *concept* may be eligible for the National *Extension* terminology. However, the final decision on inclusion of *concepts* within a National *Extension* lies with each member-state.

6.2.1.4.4 Guidelines for submission of new content

It would be impossible to provide specific rules to apply to every case where new content must be assessed, but it is helpful to list some of the recurring decisions in order to provide consistency and full explanations for the reasons that some content submissions are rejected.

New content should optimally be submitted with *fully specified names* (FSN's), and these should conform to the editorial guidelines for terms, including spelling, language, and term style guidelines. New content should also be submitted with a "parent" code to show where in the hierarchies it belongs. Assignment of this parent should be according to the editorial policy guidelines (see the other style guide documents for details).

Some common errors in past submissions include misspelled words, words submitted in the FSN that use abbreviations or acronyms, or are spelled using British spelling instead of US spelling, FSNs in plural instead of singular form, procedure FSNs in past tense instead of present tense, FSNs containing short forms with hyphens instead of fully unambiguous phrases, mismatch of the FSN tag and the submitted parent code hierarchy, terms that already exist (duplicates), terms containing "or", terms with *precoordinated* numeric ranges, and FSNs that are ambiguous and not fully specified.

Here we add guidelines for content submission that address issues other than the *concept model* or term style, but must also be considered and can be grounds for rejection of submitted content.

6.2.1.4.4.1 Usefulness

Content submitted to the *international release* shall be required to pass a test for "usefulness." The usefulness test can be passed in more than one way. At least one of the following must be satisfied:

1. Content that is used by more than one major user (a *National Release Center* such as NHS, a vendor/supplier of *Clinical Information Systems* with international scope, or a large intra-national system user such as VA or Kaiser) will be considered to have passed the "usefulness" criterion.
2. Data demonstrating significant frequency of use, or frequency of need, by a single user (single national center, or single vendor, or single health care system) can also be used as evidence in support of "usefulness".

Additional means of passing the usefulness test may be added in the future. Submissions that pass the usefulness criterion must also pass understandability and reproducibility tests, and conform to style rules.

6.2.1.4.4.2 Classification-derived phrases

6.2.1.4.4.2.1 Phrases meaningless outside the classification context

New *concept* submissions that contain certain classification-derived phrases in their FSN shall be rejected because they fail the basic tests of *Understandability, Reproducibility and Usefulness* when removed from the narrow constraints of the classification use case. Some such classification-derived phrases are:

NOS (not otherwise specified)

NEC (not elsewhere classified)

Not mentioned

With or without

The basic reason for rejection of these phrases is that they are meaningless within a clinical terminology that assumes a use case based on primary clinical documentation that allows multiple overlapping entries, rather than coding of a pre-existing record into a single best class.

The classification phrases assume that a health care record already exists, and therefore it is meaningful to have codes that depend on what has been specified or mentioned in that record. Likewise the classification phrases assume that there is a fixed set of classes into which the existing record should be placed, and therefore it is meaningful to have codes that depend on what has been classified elsewhere in the system. Lastly, the classifications must have a class for every case, and therefore they provide additional codes for categorizing a case that hasn't been properly captured by any other code – sometimes called "catch bins".

SNOMED assumes that the physician or health care provider may be in the process of documenting observations about a patient, and in this setting anything that has fidelity to the *clinical situation* may be stated and coded. It also assumes that the entire range of reproducible and useful meanings is available to be used to faithfully document the health and health care of the individual, and codes may be selected at any and all levels of specificity if desired. There is no single best code that must be selected, and using one code does not require the exclusion of another code with overlapping meaning. Finally, there are no "catch-bins", but there are multiple codes at a variety of levels of generality that may be used, and a rich set of *qualifiers* for refining the meaning of an existing code.

6.2.1.4.4.2.2 Phrases that make full statements or sentences

There are many phrases in classification systems that make statements. These phrases are in a borderline area for acceptance into *SNOMED CT*. In an ideal world, the information model would provide the mechanism for making these statements and there would be no pressure to precoordinate them into *SNOMED CT*. In the practical world, many users of existing systems are attempting to migrate from ICD-9-CM or similar coding systems toward *SNOMED CT*, and in the process they require maximum possible concurrence between the codes they are currently using and the codes to which they are migrating.

Nevertheless, *SNOMED CT* requires that when such *concepts* are included, they must have a fully specified name that omits all classification-style phrases and meets URU criteria. This may be difficult to achieve because of the idiosyncratic nature of some classification additions.

6.2.1.4.4.2.2.1 Example: episode of care and pregnancy complications

For example, ICD-9-CM adds a series of complicated "episode of care" phrases to several of the categories of disorders affecting pregnancy and delivery. Here is the full set of "fifth digit" modifiers for complications related to pregnancy:

The following fifth-digit subclassification is for use with categories 640-649 to denote the current episode of care:

- **0 unspecified as to episode of care or not applicable**
- **1 delivered, with or without mention of antepartum condition**
  - **Antepartum condition with delivery**
  - **Delivery NOS (with mention of antepartum complication during current episode of care)**
  - **Intrapartum obstetric condition (with mention of antepartum complication during current episode of care)**
  - **Pregnancy, delivered (with mention of antepartum complication during current episode of care)**
- **2 delivered, with mention of postpartum complication**
  - **Delivery with mention of puerperal complication during current episode of care**
- **3 antepartum condition or complication**
  - **Antepartum obstetric condition, not delivered during the current episode of care**
- **4 postpartum condition or complication**
  - **Postpartum or puerperal obstetric condition or complication following delivery that occurred:**
    - **during previous episode of care**
    - **outside hospital, with subsequent admission for observation or care**

The first task is to strip away all "mention of" and "unspecified" phrases, and then determine whether there is still a URU meaning. A fifth digit of "0" generates no special meaning for *SNOMED* and the phrase "unspecified as to episode of care" would be rejected as invalid for an FSN. Therefore *SNOMED CT* cannot incorporate any code that corresponds to the "0" fifth digit here.

A fifth digit of "1" means that the current episode of care involved delivery and the complication finding was antepartum. Some submitters might want to have a short phrase that says something like "finding X, delivered", where the finding occurred antepartum and the mother was delivered during the current episode of care. This constitutes two statements and our recommendation would be to place each statement in the patient record separately. However, given the request for a single *precoordinated* code that captures both meanings, it is possible to use the *SNOMED CT* context model, with two role groups, to capture the two statements.

A specific code, 641.91, would carry the ICD-9-CM phrase "Unspecified antepartum hemorrhage, delivered, with or without mention of antepartum condition".

Obviously there is a great deal of revision required to make this acceptable (even marginally) to *SNOMED*. The "unspecified" and "with or without mention of" phrases must be dropped. Clarification must be added to indicate whether it is the mother or fetus who is the subject of the record. The "episode of care" meaning is hard to capture in *SNOMED* and would be largely unrepresentable. This process might result in an FSN that says "history of antepartum hemorrhage, mother delivered (situation)", which could be defined as:

Situation, {associated-finding = antepartum hemorrhage, temporal-context = past},

{associated -finding = mother delivered, temporal-context = current or specified}

This *expression* does not completely capture all the subtle meanings associated with the ICD-9-CM code 641.91, but it is perhaps as close as we can come without major changes.

In general, *precoordination* of such codes is to be denigrated and discouraged because of the false sense of completeness it gives to those seeking to link *SNOMED* and ICD-9-CM (the meanings are ***not*** the same and ***cannot be***), and because of the added complexity such compositional *expressions* may present to decision support algorithms, particularly those that require automated processing of temporal *Relationships*. In the example given, a statement in a patient record using this code cannot tell us when the antepartum hemorrhage took place. It only tells us that it was sometime in the past – prior to delivery. The *expression* as given actually matches cases where the antepartum hemorrhage took place in a prior pregnancy. This temporal linkage problem would be avoided by using two statements with two time stamps, one coding the complication (and recording the time when it occurred) and the other one for the delivery.

6.2.1.4.4.2.3 Disjunctive aggregates

Frequently classifications employ disjunction (and/or) to group and aggregate related disorders or procedures. For example, a procedure classification might have a term for "total abdominal hysterectomy with unilateral or bilateral oophorectomy". From a procedure classification standpoint it may be appropriate to lump these all together; but from a patient standpoint, a unilateral oophorectomy leaves estrogen-producing capacity while a bilateral does not, and this is very important to the patient and to their health care.

The general rule is that a FSN should be capable of being stated without and/or. There are occasional exceptions. The first exception is where the *referent* is a single thing but there is no *name* for it. These occur in anatomy. For example, "head and neck" is really a single anatomical structure that can be defined as the body above the level of the shoulders. We don't have a name and so we use disjunction to name this body part. The second exception is where the term is an intentional navigational aggregate term. For example, we might want to group together disorders that relate to life up to the end of the neonatal period, and group them using a term such as "fetal or perinatal or neonatal disorders". But outside these broad navigational aggregate terms, it is advisable to reject disjunctive terms and instead create separate terms to be more specific about what a particular disorder is, or what particular procedure was performed.

It requires some judgment to identify the allowable exceptions, but the general rule is that FSNs should not contain disjunctions. An example of an allowable exception is | Structure of skin and/or skin-associated mucous membrane (body structure) |. The reason it is allowed is that it is useful for representing the general site of dermatological disorders, and this is useful, understandable and reproducible, from both a pathophysiological and therapeutic standpoint.

6.2.1.4.4.2.4 Excessive precoordination

It has not been possible to clearly define what "excessive" *precoordination* is. Instead, we rely on the rules for usefulness to avoid this.

6.2.1.4.4.3 Numeric ranges

Categories that depend on numeric ranges are almost always inappropriate for *precoordination*. For example, a finding of the number of lesions might be split up into ranges of 1, 2 to 5, and greater than 5. But in another context it might as easily be split into ranges of 1-2, 3 to 10, and greater than 10. It is obvious that there are literally infinite possibilities (in theory) and practically far too many possibilities to consider *precoordinations* of this type.

Rare exceptions may be made when a fixed standard uses numeric ranges and there are no reasonable alternatives. For example, some histologic scoring systems give a score of "1" when there are 0 to 5 mitoses per high power field, and a score of "2" when there are 6 to 10, etc. In these cases, where the range is really an explanation or definition of the score, it may be reasonable to make an exception.

On the other hand, it is important to avoid *precoordination* of knowledge that should reside external to *SNOMED CT* terms. For example, the serum sodium concentration is in the "normal range" when between 135 and 145 mEq/L, but for low sodium *SNOMED CT* should not use the phrase "serum sodium less than 135 mEq/L", but instead should use a phrase such as "serum sodium concentration low", and not attempt to include the definition of the lower limit of the reference range. The reason for this should be obvious – that is, sometimes reference ranges change, and sometimes systems of units change and it would cause unnecessary disruption if the *SNOMED CT* terms were dependent on those external factors. It should be forbidden to precoordinate these kinds of numeric ranges into *SNOMED CT* terms where it is not absolutely necessary.

6.2.1.4.4.4 Procedures categorized by complexity

Procedure *concepts* that include modifiers that represent procedure complexity based on the amount of effort required, or based on realm-specific definitions, are not to be added to the *international release*.

Examples of prohibited *concepts*:

Simple arthrodesis, simple repair, complex repair.

This policy does not proscribe the additions of procedures that use the words "simple" or "complex" which are defined by reproducible meanings based on what is done to or for the patient, rather than how much effort is expended in doing it.

Example of acceptable definition:

Simple mastectomy: Reproducibly defined as the removal of all breast tissue without removal of axillary contents. Differentiated from modified radical, radical, skin-sparing, and subcutaneous variants of mastectomy.

6.2.1.4.4.5 Counts of the number of procedures done

Many procedure classifications focus on the amount of resource required to carry out a procedure in order to support use cases involving reimbursement or tracking of resource expenditures. For this reason there may be a desire to precoordinate different codes for different counts of a procedure. For example, consider "placement of one stent" vs "placement of two stents".

The general advice is that the counts of number of procedures done should be handled by the information model of the patient record, and should not be *precoordinated* into *SNOMED CT* codes.

6.2.1.4.4.6 Acronyms

Acronyms are an abbreviation formed from the initial letters of other words and pronounced as a word (e.g. ASCII, AIDS). Here we reiterate the prohibition of acronyms in *fully specified names* (see *Acronyms in FSNs* on page 396. Acronyms can be misinterpreted because they are not fully spelled out. It is a mistake to assume that everyone will know what an acronym means. Therefore acronyms may not be used in *fully specified names* when the fully spelled out name is available. An exception may be where a sequence of letters started as an acronym but has now become a word in its own right, understood without expansion to its original full form. A common example would be "laser". Evidence that it is a word in its own right is that it is included in dictionaries in lower case, and the fully spelled-out meaning has become a trivia question. An example of an acronym that may *not* be included in an FSN is "CT" for "computed tomography". While those involved in imaging and radiology may regard "CT" as a word (pronounced "see tee"), it does not pass the test of being unambiguous, of appearing in a dictionary in lower case, or of its component words being a trivia question.

Acronyms are however allowed in synonyms or preferred terms when accompanied by the full expansion of the abbreviation. Expansions should be enclosed in parenthesis to reduce the technical implementation burden when indexing and searching e.g. CT (Computed tomography).

6.2.1.4.4.7 Eponyms and proprietary names

6.2.1.4.4.7.1 Eponyms

Eponyms are names that are derived from a proper name, usually the name of a person who discovered or described the thing originally. They are commonly found in a wide variety of names in health terminology, ranging across diverse areas such as anatomic structures, morphologic abnormalities, blood groups, diseases, findings, and procedures. Examples include Rutherford Morrison's pouch, vein of Galen, Aschoff body, Kell blood group, Down syndrome, Moro reflex, and Whipple procedure.

It is neither desirable nor indeed possible to completely avoid the use of eponyms in a health terminology. Nevertheless, FSNs should avoid including eponyms wherever possible in order to improve clarity of meaning and to facilitate translation to other languages. The full *Description* should be used as the FSN, and the eponymous term can be added as a synonym. For example, the FSN for "Moro reflex" should use the phrase "infant startle reflex."

Exceptions are allowed when the full *Description* is exceptionally long and unwieldy. An example of allowed exception is “Hemi-Fontan operation (procedure).” This operation is defined as a "bidirectional Glenn shunt with end-to-side anastomosis of proximal superior vena cava to right pulmonary artery with isolation from right atrium". The resulting FSN would be too long and unwieldy, so the eponym is allowed in the FSN in this case. Such exceptions require careful attention to the possibility that the eponym’s meaning may change over time.

Exceptions are also allowed for *concepts* where the eponym is the only precise clinically relevant name available, and where an artificially constructed non-eponymous name would necessarily be vague or subject to significant misinterpretation. Examples include “Hodgkin lymphoma” and “Burkitt lymphoma.”

It is permitted and encouraged to include eponyms as designations (non-FSN terms) whenever they are understandable, reproducible and useful in a given context. For example, the *preferred term* for “infant startle reflex” may be “Moro reflex.”

6.2.1.4.4.7.2 Proprietary names

Proprietary names are the proper names that have been assigned to products, usually drugs and devices, by their corporate producers. It is both necessary and useful to include proprietary names in a health terminology, subject to the following criteria:

1. **Proprietary names belong in national extensions**

   When needed in health terminology, the names and codes for proprietary products (drugs, devices, and other products including foods etc.) should be included in national extensions and not in the *international release*. This is not only because the same proprietary name may refer to an entirely different product in a different country, but also because there are differences in the process of production, including rules and regulations related to safety, packaging, labeling, and so forth, that make the meaning of proprietary product names dependent on the country or jurisdiction in which the product is approved for sale or distribution.

2. **Exception for brand names that have become eponyms**

   An exception may be made for brand names that have become eponyms. In this case, some brand names have come to stand for a category of product and not the particular brand itself. (Examples in US English include kleenex, band aid, coke, popsicle, jello, velcro, etc). These “proprietary eponyms” may be included in the *international release* as designations (non-FSN terms). Their FSNs should follow the rules for eponyms (above), and avoid the inclusion of the eponym in the FSN wherever possible. For example, the FSN for “jello” should use the phrase “fruit flavored gelatin”, and the FSN for “band aid” should use the phrase “plastic adhesive bandage strip”.

6.2.1.4.4.8 Hyphens and the word "of"

Hyphens should be avoided in FSNs, with rare exceptions. For example, the phrase “disability – all limbs” should be changed to say “disability of all limbs”. The rare exceptions occur in places such as the morphology hierarchy, where we need to distinguish categories from specific *subtypes* (see the editorial guidelines for the morphology hierarchy for explanation). In those circumstances, we may allow phrases such as “glioma – category” to differentiate the category term that includes all gliomas from a specific morphology of “glioma” as specified by ICD-O.

6.2.1.4.5 Process for adjudication (draft)

Assessment by submitters

Review by editorial staff

Rejections and deferrals appealed to Chief Terminologist, backup by Content Committee

Editorial staff will appeal both **Rejections** and **Deferrals** to the Chief Terminologist. For Rejections, there should be a way for the submitter to have a voice in the appeal process. Deferrals arise where the editorial staff needs clarity around modeling rules before proceeding (e.g. “in remission”). Resolution of deferrals may require an issue document, committee discussion, management board decision, etc. Simpler issues can hopefully be resolved more expeditiously – e.g. by a ruling of the Chief Terminologist, subject to being challenged by the Content Committee.

Principles, process and rules to be approved by Management Board

6.2.1.5 Pharmaceutical and Biologic Products Boundary and Scope for the International Release

6.2.1.5.1 Pharmacy Boundary and Scope Definition

The conclusion of the *IHTSDO* Management Board statement on medicines terminology reads as follows:

SNOMED CT will improve and support its terminology for medicines and pharmaceuticals in order to:

- *Be genuinely useful and used in delivering better health care to people*;
- *Support the representation of drugs, with a perspective close to the patient*.

The views and thinking articulated in this statement have been reflected in a number of projects. In particular, there was work that produced three documents outlining a proposed boundary for the "Pharmaceutical and Biological Products" Hierarchy in the *International Release* of *SNOMED CT* vs. National *Extension* Content. These documents were produced by the Pharmacy SIG.

Part 1 details the scope of the Pharmaceutical and Biological products hierarchy in the international release. Part 2 provides a data model, and Part 3 outlines editorial rules and style guidance.

6.2.1.5.2 Pharmaceuticals and Biologics Scope: Use Cases

In deciding where the boundary should be between the *International Release* of *SNOMED CT* and national extension, it is necessary to consider the purpose of *SNOMED CT* as a terminology.

The conclusion of the Management Board statement on medicines terminology reads as follows:

SNOMED CT will improve and support its terminology for medicines and pharmaceuticals in order to:

- *Be genuinely useful and used in delivering better health care to people*;
- *Support the representation of drugs, with a perspective close to the patient.*

To further clarify this:

The use cases for the International Release (in priority order):

- *being the foundation for the National releases (ease of linkage to the National releases)*
- *names, concept identifiers, and Relationships as a foundation to allow linking of the 3rd party Decision Support tools such as interaction checking to the International Release and therefore all National releases (definition still under discussion)*
- *International semantic interoperability*:
    - *Exchange of inferred ("generic") information to support the electronic health record across national boundaries (medication list or allergies list), (definition still under discussion)*

The use cases for the National release:

- *National release would support the point of care applications*
- *National semantic interoperability* :
    - *prescribing (e.g. the prescribing of a medication which includes prescription and drug chart)*
    - *dispensing (e.g. the supply for consumption)*
    - *administration (e.g. the use of the medicine)*
    - *recording as part of the electronic health record (medication list or allergies list)*

Note. It is recognized that National release centers may have additional use cases that require to be supported. It is however their responsibility to ensure that their National release is able to support these.

6.2.1.5.3 Pharmaceuticals and Biologics: Boundary assumptions

In defining the boundary for *SNOMED CT International Release*, we should achieve the following:

6.2.1.5.3.1 A "Risk managed" Terminology

SNOMED CT hierarchies are built using an *IS_A Relationship* so any *Relationships* stating that a *concept* has an *IS_A* to another *concept* must be true on an International arena.

The World Health Organization, British Pharmacopoeial Commission and other national regulatory authorities have been working toward standardizing the names used for medicinal substances. In order to be certain that an *International Release concept* as described has consistent meaning in the international arena *concepts* should seek to use INN where possible or other internationally recognized naming convention for the *fully specified name*.

Trademarks or brand names may be established by usage in the market place but more commonly nowadays are registered with the relevant registry or trademarks office. Since the legal right to usage is limited by the jurisdiction of the registering authority it is possible for a single brand name to be used in several nations to represent products which may or may not be similar.

For this reason the *International Release* should not contain any *concepts* that refer to a brand name whether as part of the moiety name or as the dose form.

6.2.1.5.3.2 A terminology that is Manageable / Maintainable

To achieve a terminology which is manageable and maintainable, it is necessary to consider to what level of *concept* granularity the *International Release* should go. While it is preferable to provide a level of granularity that allows maximal decision support to function, the overhead cost of maintenance needs to be considered. It is also important that combinatorial explosion is avoided where clinically appropriate.

6.2.1.5.3.3 A level of granularity at the lowest level that is useful for international semantic interoperability

The *SNOMED CT International Release* should contain *concepts* of a level of granularity to allow portability of a clinically useful medication record from one country to another including those where similar branded products may not be available. This is judged to be DRUG-STRENGTH-FORM and represented by the NPMP *concept* class. For further information on the *SNOMED CT International Release* please see the document Pharmacy Boundary and Scope Part 2.

6.2.1.5.3.4 A level of granularity at the achievable level that is useful for decision support

NOTE: for the purpose of the pharmaceutical and biologic product boundary and scope work, "decision support" means:

- Interaction checking
- Contraindication checking
- Dosage checking
- Label text for patients
- Allergy / intolerance checking
- Therapeutic duplication

For decision support to be active it is necessary to clearly identify the active moiety. For most decision support in relation to dosage and indications, route of administration is required. Route of administration is often not easy to determine at a terminology level as the licensed route of administration may vary by legal right, and usage is limited by the jurisdiction of the licensing authority. Additionally in some settings the clinicians may choose to use a non licensed route of administration. It is often more appropriate to use dose form in decision support, in particular to identify suitable dosing increments, and may also have some bearing on suitable routes for administration or dosage.

Pack size may impact on decision support such as compliance however its use is limited and the overhead in maintaining all the internationally available pack sizes may be large.

Decision support requires the identification of medicines to apply decision support to current and future therapeutic decisions to ensure safety and quality of care. A basic level of decision support is based on contraindications, potential side effects and drug-drug interactions. This can be operated using the drug, moiety or Medicinal Entity *concept*. However more sophisticated decision support requires knowledge of route of administration in order to provide guidance around dosage and in some cases contraindications and side effects. A *concept* of at least moiety + route would be necessary to provide this.

A greater level of decision support relating the previous or current medication history could be achieved by providing a *concept* that gave indication drug, strength and route. Populating this runs the risk of terminological explosion as all potential combinations of drug + dose + route would need to be populated. Although the *Relationship* between form and route is not an absolute many-to-one match, for most drug forms the route is implicit. The provision of a *concept* of drug + dose + form would provide a pragmatic solution since only those known to be available internationally would need to be populated. However this would also have the benefit of allowing sufficient granularity for effective decision support.

6.2.1.5.3.5 A point at which existing terminologies can link to SNOMED CT

It is anticipated that national terminologies would wish to link to *SNOMED CT International Release* at a number of potential *concept* levels within the pharmaceutical and biological product hierarchy.

We envisage that there may be two ways of achieving this objective by:

- using the *SNOMED CT International Release concepts* within national extensions
- mapping the relevant national code set to the *SNOMED CT International Release concepts* when required.

Examples of existing terminologies that may use *SNOMED CT* include:

- Australia's AMT will replace MP and MPUU which are equivalent to the moiety and the single unit dose form with *International Release concepts*. It should be noted that not all AMT MP and MPUU *concepts* will have equivalents in the *International Release* however those that are not equivalent will link using an *IS_A Relationship* to a core *concept*.
- UK's dm+d links VTM and VMP which are equivalent to the moiety and the single unit dose form which are *SNOMED CT International Release concepts*. It should be noted that not all dm+d VTM and VMP *concepts* will have equivalents in the *International Release* however those that are not equivalent will link using an *IS_A Relationship* to a core *concept*.
- US's RxNorm links semantic clinical drug (drug + strength + dose form). Other *concept* classes that could link to *International Release* are semantic clinical drug form (drug + dose form) and semantic clinical drug component (drug + strength) At present the editorial policy for products in *SNOMED CT International Release* does not support linking for the last two *concept* classes although there are some instances where a *concept* of the format drug + dose form have been created.
- Netherlands' G standard can link the 'substance' which is equivalent to the moiety and moiety + salts/water for hydration *SNOMED CT International Release concepts*. In addition Netherlands' G standard *concept* class of 'Generic Product Characteristics' (GPK) which equivalent to the proposed NPMP of the *SNOMED CT International Release concepts*.

6.2.1.5.3.6 Support for historical data in records

Medicines that are no longer used in current therapeutics still have a place in decision support, the recording of medical history, previous adverse reactions or the recording and notification of sensitivities. Retirement of *concepts* in *SNOMED CT* means removal from the hierarchy and removal of all *Relationships* such as HAS_ACTIVE_INGREDIENT. If this is actioned then in order for a *concept* to be used for decision support it would be necessary to utilize a *concept history mechanism* that contains all the required data elements. The current *SNOMED CT concept history mechanism* does not retain *Relationship* information (such as HAS_ACTIVE_INGREDIENT) and so would be inadequate. The additional issue of the need to identify *concept* history resulting from correction as opposed to history resulting from cessation of availability of a product would also need to be addressed. Therefore it is proposed that *concepts* describing medicines that are no longer current therapeutics should not be retired routinely provided they are, or were, valid *concepts*.

6.2.1.5.3.7 Timely population of agreed concepts

Timely population of *International Release concepts* is a significant issue for usefulness of the terminology. For other parts of the hierarchy a new *concept* is added when two or more *IHTSDO* member nations request this addition. This method of population is unlikely to be sufficiently responsive for the population of this part of the hierarchy. It is proposed that when a medicinal product is licensed within one of the member nations it can be proposed for inclusion within the *International Release*. Unlicensed medications and other edge cases would be included as defined in the paper describing the scope of products to be included in the hierarchy.

6.2.1.5.4 Pharmaceuticals and Biologics: Definitions and Boundaries

6.2.1.5.4.1 What is a Medicine?

Definition (as per EU directive)

"Any substance or combination of substances being presented as having properties for treating or preventing disease in human beings;

Any substance or combination of substances which may be used in or administered to human beings either with a view to restoring, correcting or modifying physiological functions by exerting a pharmacological, immunological or metabolic action, or to making a medical diagnosis."

Article 1 of Directive 2001/83/EC

6.2.1.5.4.2 Boundaries and Scope for Medicinal Products

Using the above definition taken from the EU as the starting point, the scope for inclusion in the *SNOMED CT International Release* of the Pharmaceutical and Biological products hierarchy should primarily aim to only include licensed medicinal products. However, it is noted that many boundary issues exist. Below is more information provided to further describe the boundaries to be applied to *concepts* for inclusion in the *International Release* in relation to product types.

- *Licensed* vs. *unlicensed*
- *Health supplements and OTC medicines*
- *Nutritional Products*
- *Homoeopathic Medicines*
- *Herbal Products*
- *Medical Devices*
- *Nanotechnology products*
- *Cosmetics and Toiletries*
- *Single component vs multi-component packs*

Additionally there are items that require more information on how to be handled or excluded *Drugs, Biologics and other products not yet ruled in or out of scope* on page 252:

- Human derived therapeutic products
- Dental products
- Monoclonal antibodies
- Ingredients for extemporaneously dispensed products.

6.2.1.5.4.2.1 Licensed Medicines

Licensed medicines and supporting data can be submitted by *IHTSDO Members* for inclusion.
A medicine would need to be licensed in at least one *IHTSDO Member* nation and include a definitive strength.

The medicines to be included as the highest priority are those that could be added to a patient medication profile *if* they have been prescribed, dispensed and/or administered.

It was noted that many specialist systems and processes that have medicines use information within them that may not be connected to a medication profile application and therefore may be a lower priority for inclusion ***but are in scope*** ; for example:

- Vaccination – this is often seen as a procedure, with the medicinal product used being recorded as part of the procedure e.g. a 67308009 l yellow fever vaccination (procedure) l actually involved the administration of the yellow fever vaccine product, but this latter does not get recorded.
- Anesthetics – these and other supporting medications given during surgery are often recorded in specialist systems and not (yet) "shared" with medication profile applications. Note that medications administered during day case surgery would also fall into this category.
- X-ray contrast media and other diagnostic agents – these often highly allergenic products are often used in isolation and again recorded in specialist systems (which may still be paper-based) and the information does not find its way to a medication profile.
- Orphan drug products: these are licensed products intended for use in conditions where the patient population is likely to be very small, such as for rare diseases, resulting in lack of commercial development of the drug product due to limited revenue potential for the manufacturer. In some realms, the licensing process is modified to make it easier to gain marketing approval for drugs with "orphan" status.

6.2.1.5.4.2.2 Unlicensed Medicines

Unlicensed medicines may be included in the *International Release* only where good data can be provided and is agreed upon by two *IHTSDO Member* nations. This must include a definitive strength. However, unlicensed medicinal products are not routinely included in the *International Release* but rather should be included in national extensions. This includes medicines which are part of a Clinical Trial which should not be included in the *International Release* but are a national extension issue.

6.2.1.5.4.3 Boundaries and scope for additional products

6.2.1.5.4.3.1 Health Supplements and OTC Medicines

Definition (as per EU directive)

OTC = "over the counter" medicines; those sold directly to patients or the public either from a pharmacy or in a non-healthcare environment without the requirement for a prescription.

For the purposes of this document, health supplement and OTC medicines are defined as being those products that are **ONLY** intended as preparations for sale directly to the public without the requirement for a prescription.

These are ruled out of scope for the *International Release* and if required by a release center should be added to a national extension.

6.2.1.5.4.3.2 Foods, Food Supplements, and Nutritional Products

Foods

Definition (as per EU directive)

"Food (or foodstuff) *means any substance or product, whether processed, partially processed or unprocessed, intended to be, or reasonably expected to be ingested by humans…*

Food shall not include…medicinal products within the meaning of Council Directive 65/65/EEC [now Directive 2001/83/EC.]"

Foods, generally, then are NOT medicinal products, and can be distinguished from medicinal products on the grounds of being identified as products which the average consumer would regard as something to be eaten, drunk or chewed as part of his/her diet.

It is anticipated that food *concepts* would not be included in patient's medication records as a matter of course and therefore any decision support available would be limited. N.B. Allergy recording could be supported by the use of the Substance hierarchy in *SNOMED CT*.

Scope for foods:

These are ruled out of scope for the *SNOMED CT International Release* pharmaceutical and biologics hierarchy unless they are licensed products and specific medicinal claims are made for their use. It is anticipated that allergies to such products would be recorded using *concepts* from the Substances hierarchy. Decision support is not generally utilized for foodstuffs. If they are required by a *National release* center the *concepts* should be added to the national extension.

Food supplement

Definition (as per EU directive)

"foodstuffs, the purpose of which is to supplement the normal diet and which are concentrated sources of nutrients or other substances with a nutritional or physiological effect, alone or in combination, marketed in dose form"

Food supplements are concentrated sources of nutrients or other substances with a nutritional or physiological effect whose purpose is to supplement the normal diet. They are marketed 'in dose' form i.e. as pills, tablets, capsules, liquids in measured doses etc.

Scope for food supplements:

These are ruled out of scope for the *SNOMED CT International Release* pharmaceutical and biologics hierarchy unless they are licensed products and specific medicinal claims are made for their use. It is anticipated that allergies to such products would be recorded using *concepts* from the Substances hierarchy. Decision support is not generally utilized for food supplements. If they are required by a *National release* center the *concepts* should be added to the national extension.

Dietary foods for special medical purposes

Definition (as per EU directive)

Foods used in patients with specific intolerance conditions (e.g. lactose free foods) or foods for patients with gluten sensitive enteropathies, such as coeliac disease ("gluten-free foods"), and low protein foods for patients suffering from inherited metabolic disorders, renal or liver failure requiring a low-protein diet.

Directive 1999/21/EC, implemented into UK law by the Medical Foods (England) Regulations 2000, the Foods for Special Medical Purposes (Scotland) Regulations 2000, the Medical Food Regulations (Northern Ireland) 2000, and the Medical Food (Wales) Regulations 2000.

In addition there are food products that seek to provide nutritional support to athletes and persons engaged in significant exercise (sports supplements) as well as slimming and dieting products. They fall within the definition of a medicinal product if they make medicinal claims or if they modify physiological functions by acting pharmacologically, immunologically or metabolically, or are marketed and used with a view to having such an effect.

Scope for dietary foods for special medical purposes:

These are ruled out of scope for the *SNOMED CT International Release* pharmaceutical and biologics hierarchy unless they are licensed products and specific medicinal claims are made for their use and there is sufficient data available to allow the *IHTSDO* support organization to fully model. It is anticipated that allergies to such products would be recorded using *concepts* from the Substances hierarchy. Decision support is not generally utilized for such foods. If they are required by a *National release* center the *concepts* should be added to the national extension.

6.2.1.5.4.3.3 Homeopathic Medicines

Definition (as per EU definition)

Homoeopathic medicinal product Definition

Any medicinal product prepared from substances called homoeopathic stocks in accordance with a homoeopathic manufacturing procedure described by the European Pharmacopoeia or, in the absence thereof, by the pharmacopoeias currently used officially in the Member States. A homoeopathic medicinal product may contain a number of principles."

Article 1(5) of Directive 2001/83/EC, as amended by 2004/27/EC

Scope for homeopathic medicines

These are ruled out of scope for the *SNOMED CT International Release.* If at some point in the future these were to be brought into scope it is anticipated that they would need to be modeled in a new section of *SNOMED CT* and would require a separate project to consider.

6.2.1.5.4.3.4 Herbal Products

Definition

"*Herbal Drug Preparations are obtained by subjecting herbal drugs to treatments such as extraction, distillation, expression, fractionation, purification, concentration or fermentation. These include comminuted or powdered herbal drugs, tinctures extracts, essential oils, expressed juices and processed exudates.*"

(European Pharmacopoeia)

Herbal products typically contain a mix of compounds and it is often difficult to identify which are the therapeutically relevant ones. In addition since these products are usually unlicensed it is often not possible to access information giving details of the compounds and the amounts present. Because of this lack of information decision support for these products is limited and the information required to represent them in the terminology may not be available.

Scope for herbal products

These are ruled out of scope for the *SNOMED CT International Release* unless they are licensed products and specific medicinal claims are made for their use, and there is sufficient data available to allow the *IHTSDO* support organization to fully model. If at some point in the future these were to be brought into scope it is anticipated that they would need a separate project to consider.

If these products are required to be represented in the terminology by a *National release* center the *concepts* should be added to the national extension.

It is anticipated that allergies to such products would be recorded using *concepts* from the Substances hierarchy. For this reason, substance *concepts* that would be used in herbal medicines should continue to be included in the *International Release* within the Substance hierarchy.

6.2.1.5.4.3.5 Medical Devices

A medical device

Definition (as per EU definition)

"any instrument, apparatus, appliance, material or other article, whether used alone or in combination, including the software necessary for its proper application intended by the manufacturer to be used for human beings for the purpose of:

- diagnosis, prevention, monitoring, treatment or alleviation of disease,

- diagnosis, monitoring, treatment, alleviation of or compensation for an injury or handicap,

- investigation, replacement or modification of the anatomy or of a physiological process,

- control of conception,

and which does not achieve its principal intended action in or on the human body by pharmacological, immunological or metabolic means, but which may be assisted in its function by such means"

In the case of a medical device, the principal intended action is typically fulfilled by physical means (including mechanical action, physical barrier, replacement of, or support to, organs or body functions). The action of a medicinal product is typically achieved by pharmacological, immunological or metabolic means; a substance administered for diagnostic purposes, even though it does not act in such ways, is also usually considered to be a medicinal product.

Scope of medical devices

These are out of scope of the pharmaceutical and biologics hierarchy and are to be modeled in a different section of *SNOMED CT*. They require a separate project to consider if changes are required. N.B. There is a separate proposal being developed from the *UK NHS* Connecting for Health with regard to a model for these items.

Products that Incorporate or Administer an "Ancillary" Medicine

Definition (as per EU definition)

Products that incorporate or are used to administer a drug, may be regulated as either medical devices or as medicinal products, depending on the principal intended function of the product and the method by which this action is achieved. There are three main types of medical device which incorporate or are used to administer a medicinal product:

1. *Devices which are used to administer medicinal products: for example, a syringe marketed empty, medicine spoons, droppers etc. This category also includes devices which can be refilled with further doses of medication contained within the same pack as the medicine. All of these products are covered by the Medical Devices Regulations. If they are included separately in a pack with the medicine they will still need to comply with the MDR, including labeling provisions.*
2. *Devices for administering medicinal products where the device and the medicinal product form a single integral product designed to be used exclusively in the given combination and which are not re-usable or re-fillable: For example a syringe marketed pre-filled. These products are covered by medicines legislation, although in addition to this, the relevant essential requirements in Annex 1 of the Medical Devices Directive 93/42/EEC apply with respect to safety and performance related features of the device (e.g. a syringe forming part of such a product).*
3. *Devices incorporating, as an integral part, a substance, which, if used separately, may be considered to be a medicinal product and which is such that the substance is liable to act upon the body with action ancillary to that of the device: For example a heparin coated catheter. These products are subject to the MDR*. In addition, the safety, quality and usefulness of the medicinal substance must be verified by analogy with the methods required in Directive 2001/83/EC concerning the testing of proprietary medicinal products. Under the classification rules set out in the Medical Devices Directive (see Bulletin Number 10); such a device would fall into class III under rule 13. The Notified Body carrying out relevant conformity assessment procedures in respect of such a device must consult a Member State competent authority for medicinal products or the EMA where appropriate on the medicinal aspects of the device. Note that in the MHRA's opinion 'integral' means a single component product (e.g. such as coated or incorporated within) rather than a pack containing the two components (i.e. a drug and a device).*

Scope of products that incorporate or administer an ancillary medicine

Items that are considered to be administered products are **within scope**. Products that fall into these categories include:

- Dialysis fluids including Peritoneal dialysis solutions, including those for CAPD, Hemodialysis solutions, Hemofiltration solutions and iontophoresis solutions
- Ophthalmic Products including
  - Contact lens care products
  - Unmedicated artificial tears
  - Fluorescein and Rose Bengal products
  - Ocular endotamponades and viscoelastic/viscosurgical products
  - Ophthalmic irrigation solutions
- Non-medicated dermatological creams
- Irrigation solutions
- Viscoelastic gels for joint lubrication
- IUDs with a hormone action, and spermicidal preparations – creams, pessaries and sponges/films where the primary purpose is as a drug delivery system are all medicinal products.

All other device items are considered **out of scope** of the pharmaceutical and biological product hierarchy and should be modeled as devices. There are devices that incorporate a drug substance but do not deliver a specific amount of that drug either in total or as a rate of release; these include

- stents incorporating anticoagulant or anti-thrombotic medication (gelatin, heparin, proteins)
- bone cements incorporating antibiotics
- devices with bacteriological coatings (chlorhexidine, benzalkonium chloride, silver, antibiotics)

Other items considered **out of scope** include:

- Medicated dressings. Most surgical dressings are devices, unless they are medicated and are making a “medicinal” claim; bandages are medicines only if the therapeutic effect of their medication is their primary purpose.
- Disinfectants that are designed for use on objects rather than on or in living subjects are not be considered medicines.
- Sutures etc. are medical devices, but some biological sealants may be classed as medicines, particularly if their mechanism of action is pharmacological (e.g. working on the clotting cascade)
- Resorbable bone plates and hard tissue scaffolds (hydoxyapatite and calcium phosphate – both with and without collagen – bioglas, coral and cartilage repair systems) (including scaffolds used in dentistry) are usually medical devices unless they contain bioactive materials.
- Artificial skin systems that do not contain material of human origin will be medical devices (probably).
- Devices used to deliver anesthetic products
- Medicinal gases are considered to be medicines, but the equipment used to administer them (or to make (e.g. oxygen concentrators or generators) or transport them (e.g. piping) is considered to be medical device.
- Contraceptive Products. Intrauterine devices (IUDs) without hormonal/pharmacological action, diaphragms and condoms (with or without spermicide) are medical devices.

* MDR – Medical Device Regulations

6.2.1.5.4.3.6 Nanotechnology Products

Definition

Nanotechnologies are the design, characterization, production and application of structures, devices and systems by controlling shape and size at nanometer scale.

(Royal Pharmaceutical Society of Great Britain definition, June 2004)

Nanomedicine is defined as the application of nanotechnology in view of making a medical diagnosis or treating or preventing diseases. It exploits the improved and often novel physical, chemical and biological properties of materials at nanometer scale (EMEA, June 2006).

The majority of current commercial applications of nanotechnology in medicine are geared toward drug delivery to enable new modes of action, as well as better targeting and bioavailability of existing medicinal substances. Novel applications of nanotechnology include nanostructure scaffolds for tissue replacement, nanostructures that allow transport across biological barriers, remote control of nanoprobes, integrated implantable sensory nanoelectronic systems and multifunctional chemical structures for drug delivery and targeting of disease.

At the moment there are no regulations specific to medicines or medical devices using nanotechnology, but it is a developing area and new products are likely in the coming years.

Scope of nanotechnology products

Products currently incorporating nanotechnologies are already available in member nations as licensed medicinal products. The use of these technologies can significantly affect the chemical and physical properties of the active ingredient and would impact on decision support for products containing them. These should be considered as in scope for inclusion in the *SNOMED CT International Release*.

6.2.1.5.4.3.7 Cosmetics and Toiletries

Cosmetics (products applied to the body, especially the face to improve its appearance – New Oxford English Dictionary) and toiletries (articles used in washing and taking care of one's body, such as soap, shampoo and toothpaste - New Oxford English Dictionary) are generally not defined as medicines.

However, if any of these contain a pharmacologically active substance or to make medicinal claim, then they become "subject to medicines control" (e.g. a toothpaste making claims to treat or prevent 'sensitive teeth' becomes "a medicinal product").

It is anticipated that cosmetics and toiletries *concepts* would not be included in patient's medical records as a matter of course and therefore the use case for application of decision support available would be limited. N.B. Allergy recording could be supported by the use of the Substance hierarchy in *SNOMED CT*.

Scope of cosmetics and toiletries

These are ruled out of scope for the *SNOMED CT International Release* unless they are licensed products or specific medicinal claims are made for their use e.g. shampoo for lice treatments.

It is anticipated that allergies to such products would be recorded using *concepts* from the Substances hierarchy. Decision support does not generally consider cosmetics and toiletries.

6.2.1.5.4.4 Drugs, Biologics and other products not yet ruled in or out of scope

Human derived therapeutic products including:

- Soft tissue fillers (collagen and silicone elastomer dispensions) are medical devices, but if human tissue derived, they may be seen as medicinal products or may be part of the new "human derived therapeutic products" class.
- Human tissue products such as dura (mater) grafts, skin fibroblasts and bone tissue are either considered medicines, or are currently not regulated as "products" as such.

Blood Products

Currently, blood and blood products are found in the pharmaceutical/biological product hierarchy under the parent *concept* 410652009 | blood product (product) |. This includes autologous blood products.

Note: this does not include products manufactured from blood, such as albumin solutions and clotting factors; these are found elsewhere in the pharmaceutical/biological product hierarchy under the parent *concept* 346348003 | blood derivative product (product) |.

Note: Artificial plasma volume expanders are considered to be medicinal products.

N.B. Blood products are included in the scope of the UK project on devices.

Dental Products

- Sealants for fissures and root canal pits
- Root canal dressings, which usually contain antibiotics and/or antiseptics
- Pulp capping material and materials for dry socket preparation
- Disclosing tablets and other in-vivo diagnostics
- Hemostatic agents and astringents
- Fluoride preparations – tablets, gels (toothpastes) and varnishes
- Periodontal antibacterials – gels, ointments and fibers
- Periodontal dressings
- Antibacterial mouthwashes and gels
- Medicated mouth ulcer preparations are medicines
- Desensitizing agents
- Artificial saliva products are devices

The above are potentially a mixture of medicines and devices; input from dental professionals as to the use cases for dental product *concepts* would be required before decisions would be taken on how best to handle this area.

Monoclonal antibodies

Where there are available licensed medicinal products using this technology, those products are to be considered as in scope for the Pharmaceutical and Biological product hierarchy for the *SNOMED CT International Release*. Monoclonal antibodies may also be used as in-vitro diagnostics or as immunotoxins; currently these are not considered to be "products".

Extemporaneously Dispensed Products

Since these are in theory potentially infinite in their combinations, strengths and presentations they are not considered appropriate for addition to the *SNOMED CT International Release*. Where there are standard pharmacopeial preparations or those endorsed by appropriate specialist organizations (for example Royal colleges) a *National release* Center may decide to add these to their extension.

Ingredients for Extemporaneously Dispensed Products

Ingredients for extemporaneously dispensed medicines are likely to be "products" in their own right (e.g. raspberry syrup, chloroform spirit). Products such as these are currently located as children of 43747001 | drug excipient (product) | in the pharmaceutical/biological product hierarchy. This may not be strictly appropriate, and this should be reviewed.

There may be some cases where there is a requirement to use a "substance" *concept* to describe an ingredient within an extemporaneous preparation.

Leeches and Maggots

Those that are supplied commercially are generally accepted as being medicinal products when intended for medicinal use however at present are not subject to licensing as medicinal products. Without the licensing requirements it may be difficult to access sufficient information to allow the support organization to add these *concepts*. In addition it would appear that the decision support available to guide the use of these products is limited at this time.

While the *concept* 410969008 | sterile maggots (product) | is present in the pharmaceutical/biological product hierarchy and there is the *concept* 8181006 | maggot (organism) | in the organism hierarchy, currently leeches are *concepts* in the organism hierarchy only.

It is recommended that a decision as to whether these and other similar biological entities are considered for inclusion in this hierarchy is pended until further information is available and may require an additional project to define editorial rules and an appropriate model for such *concepts*.

6.2.1.5.5 Pharmaceuticals and Biologics: International Release Model

6.2.1.5.5.1 Model Overview

The data representation is based on *SNOMED CT*. As noted previously, *SNOMED CT* is a comprehensive and precise clinical *reference terminology*. It provides a comprehensive list of clinical terms and identifiers that allows complex clinical *concepts* to be described in a way that computers can interpret.

Building the terminology in this manner provides:

Descriptions: Defined using the standard *SNOMED CT Description* types - that is ' *Fully Specified Name* ' and a *preferred term* [1] will be added for the *International Release*

Relationships: Each *concept* will have a *SNOMED CT* defined *Relationship* to an appropriate super type *concept*

Inherited defining *Relationships* (where appropriate)

Specific defining *Relationships* and *Relationships* to other defined *concepts*;

Historical relationships.

[1] RF2 does not have a *Description* type value " *Preferred Term*", only types of " *Fully specified name* " and "Synonym", where the latter may be refined either to a "*Preferred term* " or to a "Synonym" within a language *reference set*. As a result of this change, in RF2 the preference for particular *Descriptions* in a language or *dialect* will be represented in the language *reference set*, and not in the *description file*.

6.2.1.5.5.2 Model Assumptions

- The *National releases* will handle multi-component packs and they are therefore excluded from the scope of the *International Release*. However, it is noted that the *International Release* must represent the individual active components of these packs. (Inert components of a multicomponent pack are expected to be the responsibility of the *National release* – this includes diluents and solvents.) For the purposes of this document a multi-component pack is defined as a pack that contains two or more separate components each of which is a virtual medicinal product or appliance *concept* (in the case of applicators) in its own right although it may not be available or prescribable alone.

  N.B. investigation of how multi-component packs are handled by decision support knowledge vendors is required to ensure that this is a valid assumption to make.
- Dual representation of strength will be managed by terms included in the *National release*.
- The existing top hierarchy will be required to be removed in a manner that is consistent with the *IS_A* overload project undertaken by the allergies redesign working group. See Revision of upper levels on page 254.
- Modified preparations. Since no phamacopeial standards exist for modified release dose forms, the intended duration of action should not be identified for *concepts* in the *International Release*. Should a member nation wish to create *concepts* indicating duration (for example, Morphine 10mg 12-hour modified release capsules), these would be managed by the national extension. This means that in the *SNOMED CT International Release*, modified release preparations (for example, the NPMP 'Morphine 10mg modified release capsules') would be identified with no reference to duration of action.
- It is recognized that implementations may have a requirement for a term that has fewer characters than the *Fully Specified Name*. However since terms containing abbreviations should not be part of the *International Release* these synonyms will be the responsibility of the *National release* Center and contained in their national extension.
- For countries without a *National Release* Center it is anticipated that they will utilize another country's National *Extension* as their drug dictionary since the *International Release* is unlikely to provide all the *concepts* and terms required to support prescribing.
- There is a requirement for further specification of issues relating to Dose Forms including:
  - How Form and Strength are represented e.g. where units are used to describe strength, does it need to be explicitly stated what flavor of units is being used?
  - How to identify and work with appropriate standards to harmonize content, for example HL7 and the European Directorate for the Quality of Medicines (EDQM), ISO and others.
  - How to define new therapeutic forms as required.
  - How to represent dual chamber medicinal products e.g. Coverlet
- For a Non-proprietary Medicinal Preparation *concept* to be eligible for inclusion in the *SNOMED CT International Release*, it should be licensed in a member nation. Monitoring lists of newly licensed entities can facilitate a more proactive population of the Pharmaceutical and Biological product hierarchy.
- Valid *Concepts* relating to medicinal products that are no longer available internationally should still remain within the hierarchy. This is because the *concept* may be part of a medical record as previous medical history and would still have value provided its existing *Relationships* and attributes are preserved.

6.2.1.5.5.3 Revision of upper levels

IS_A overloading in the Pharmaceutical and Biological Product hierarchy causes problems with inappropriate inheritance of information. The top levels of this hierarchy – all those above the Medicinal Entity class - should be removed to eliminate inappropriate inheritance in *SNOMED CT*.

The multiple *Relationships* at the upper levels of this hierarchy cause problems for implementers particularly where therapeutic agents have more than one potential use. Timolol, Aspirin, Glycerin, Prednisolone products are some of the *concepts* affected by this problem of multiple indications.

For example it is currently possible by navigating the *IS_A Relationships* to infer that 407802008 | timolol 0.1% liquid eye gel (product) | *IS_A* 1182007 | hypotensive agent (product) |.

This occurs because the parent *concept* 422023000 | ophthalmic form timolol (product) | has three parent *concepts*;-

404635006 | beta blocker glaucoma preparation (product) |

440130005 | ocular dosage form product (product) |

85591001 | timolol (product) |

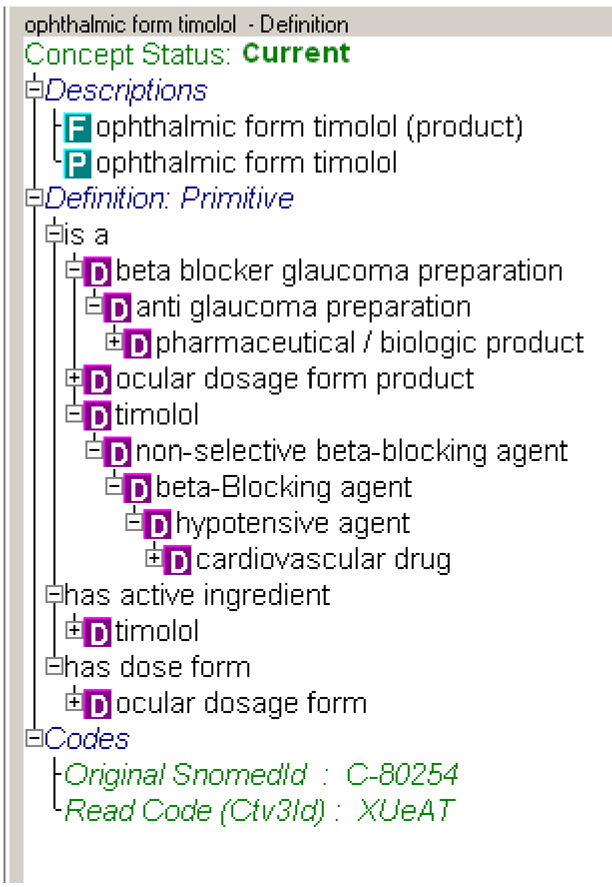


Figure 57: Timolol Supertypes

All of these are reasonable for the *concept* of 85591001 | timolol (product) | but this *IS_A Relationship* is not always true as the *Relationships* are inherited further down the hierarchy, as | timolol 0.1% liquid eye gel | inherits the | cardiovascular drug | *concept* through its | timolol (product) | parent.

The plan is to eliminate this type of error by removing / relocating the upper levels of the hierarchy (any *concepts* above the Medicinal Entity or Medicinal Entity with Modifier level).

It is estimated that this will involve the review and relocation / retirement of approximately 6,000 *concepts*.

Although it is not anticipated that these upper level “therapeutic classification” type *concepts* would be used directly for prescribing, they do have a variety of uses, which include:

- formulary management and pharmaco-epidiemiology
- medication statement – for example “patient taking NSAID” (exact medicine unknown); used in pharmacovigilance and in emergency care

- various types of querying – for example:
  - decision support – “contra-indicated in patient using a beta-blocker”
- audit – “find all patients using beta-agonists”
- navigation of medicinal product information

These removed / relocated *concepts* would be placed in a separate part of the pharmaceutical and biological product hierarchy and could then be associated with the main pharmaceutical and biological product hierarchy *concepts* by a role based *Relationship* rather than an *IS_A Relationship*.

A separate part of the pharmaceutical and biological product hierarchy is to be created under a parent *concept* of Therapeutic product group (product).

This will be a flatter hierarchy and will contain *concepts* denoting mode of action or chemical structure without specific indications for use.

Examples of such *concepts* are:

33252009 | beta-Blocking agent (product) |

83522001 | non-selective beta-blocking agent (product) |

427804001 | chemokine receptor antagonist (product) |

32249005 | antiviral agent (product) |

New *concept* - Osmotic agent (product) – a replacement for the *concept* of 317573002 | osmotic laxatives | that would be retired.

These *concepts* will be associated with the Medicinal Entity or Medicinal Entity with Modifier *concept* class by a new *Relationship* of 'plays therapeutic role'. One Medicinal Entity or Medicinal Entity with Modifier *concept* may have none to many 'plays therapeutic role'*Relationships* dependent upon whether it has multiple modes of action or if the mode of action is not known.

Concepts from this top level of the pharmaceutical and biological product hierarchy that do not fit with the new editorial guidance, and which therefore have no *Relationship* to any ME or ME with modifier will be retired.

Examples of *concepts* that it is anticipated will be retired are:

Those that indicate a specific indication, e.g. :

413969007 | dental disclosing preparation (product) |

404635006 | beta blocker glaucoma preparation (product) |

61621000 | laxative (product) |

Those that provide general groupings too non-specific to be useful:

350074002 | chelating agents and antidotes (product) |

105924008 | laxative, cathartic AND/OR purgative (product) |

Those denoting the body system that the medicinal entity acts upon:

350060009 | drug groups primarily affecting the musculoskeletal system (product) |

14833006 | cardiovascular drug (product) |

39741008 | gastrointestinal drug (product) |

It is important to note that the removal of these therapeutic “grouper” *concepts* from the hierarchy would result in there being a large number of *concepts* as direct children of the *concept* 373873005 | pharmaceutical / biologic product (product) |. It is expected this will be in the region of 5,000-10,000 *concepts*. This may make the operation of some browsers difficult but is seen as essential to remove inappropriate inheritance. It may be that some implementations might choose to use the new grouper *concept* structure as an aid to navigation and display of this hierarchy in browsers, which would overcome this problem. However, it should be noted that *in normal clinical practice* the medicinal *concepts* themselves (ME, ME with modifier and NPMP) are usually accessed by direct alpha-numeric search, not through browsing a hierarchy, so the removal of large numbers of grouper *concepts* is not expected to cause issues to most non-browser-based implementations.

6.2.1.5.6 Pharmaceuticals and Biologics Boundary and Scope: Acronyms

Acronym	Explanation
AMT	Australian Medicines Terminology
dm+d	Dictionary of Medicines and Devices (UK)
EMA	European Medicines Agency
FSN	*Fully Specified Name*
IHTSDO	International Health Terminology Standards Development Organization
INN	International Non-proprietary Name
ME	Medicinal Entity
MHRA	Medicines and Healthcare Products Regulatory Authority (UK)
MP	Medicinal Product
MPUU	Medicinal Product Unit of Use
NHS	*National Health Service* (UK)
NPMP	Non-Proprietary Medicinal Preparation
OTC	Over the Counter
RF2	*SNOMED CT Release Format 2*
SNOMED CT®	*SNOMED Clinical Terms*
VTM	Virtual Therapeutic Moiety
VMP	Virtual Medicinal Product

6.2.2 Attributes Used in SNOMED CT

This part of the guide provides an overview of the defining attributes used by the *SNOMED CT Concept Model*. Further details are provided in the chapters dedicated to each hierarchy.

6.2.2.1 Summary of attribute domains and ranges

6.2.2.1.1 Defining Attributes by Hierarchy and Domain

The following table lists the top-level hierarchies for which there are defining attributes. Not all hierarchies in *SNOMED CT* have defining attributes. They were developed in the priority areas first. The highest priority was to develop defining attributes that would be useful for aggregated analysis of outcomes, decision support, knowledge-based practice guidelines, etc. in a clinical setting. Therefore, defining attributes in *SNOMED CT* were first assigned to those hierarchies where retrieval of clinical data is most useful and relevant: procedure, finding, and *situation with explicit context*. *Concepts* in other hierarchies can be *primitives* and still serve as the values of attributes for the *concept* definitions of the main hierarchies.

Some domains are not top-level hierarchies, and some top-level hierarchies are not domains. Each item in the *Hierarchy* column refers to the top-level *hierarchy* where the attribute applies. Some of these do not actually apply at the very top of the *hierarchy*, but are restricted to a domain defined lower down. Some of them apply to more than one top-level *hierarchy*. Each item in the *Attribute* column refers to a single attribute that resides in the *Attribute hierarchy*.

Table 101: Defining Attributes by Top-Level Hierarchy

HIERARCHY	ATTRIBUTE
Body structure	| Laterality |

HIERARCHY	**ATTRIBUTE**
Clinical finding	| After |
	| Associated morphology |
	| Associated with |
	| Causative agent |
	| Clinical course |
	| Due to |
	| Episodicity |
	| Finding informer |
	| Finding method |
	| Finding site |
	| Has definitional manifestation |
	| Has interpretation |
	| Interprets |
	| Occurrence |
	| Pathological process |
	| Severity |
Situation with explicit context	| Associated finding |
	| Associated procedure |
	| Finding context |
	| Procedure context |
	| Subject relationship context |
	| Temporal context |

HIERARCHY	ATTRIBUTE
Event	I After I
	I Associated with I
	I Causative agent I
	I Due to I
	I Occurrence I
Pharmaceutical / biologic product	I Has active ingredient I
	I Has dose form I
Physical object	I Has active ingredient I
	I Has dose form I

HIERARCHY	ATTRIBUTE
Procedure	I Access I
	I Component I
	I Direct device I
	I Direct morphology I
	I Direct substance I
	I Has focus I
	I Has intent I
	I Has specimen I
	I Indirect device I
	I Indirect morphology I
	I Measurement method I
	I Method I
	I Priority I
	I Procedure device I
	I Procedure morphology I
	I Procedure site I
	I Procedure site - Direct I
	I Procedure site - Indirect I
	I Property I
	I Recipient category I
	I Revision status I
	I Route of administration I
	I Scale type I

HIERARCHY	ATTRIBUTE
	l Surgical Approach l
	l Time aspect l
	l Using device l
	l Using access device l
	l Using energy l
	l Using substance l
Specimen	l Specimen procedure l
	l Specimen source identity l
	l Specimen source morphology l
	l Specimen source topography l
	l Specimen substance l

Table 102: Top-Level Hierarchies That Have No Defining Attributes

HIERARCHY	ATTRIBUTE
Attribute	none
Environments and geographical locations	none
Observable entity	see draft of new model
Organism	see draft of new model
Physical force	none
Qualifier value	none
Social context	none
Special *concept*	none
Staging and scales	none
Substance	see draft of new model

Each attribute has a specific domain to which it applies; in many cases these domains are simply the same as the top-level hierarchies, as listed in parentheses after the domain name. In other cases, there is a more restrictive domain. In the following table, the left-hand column names the specific domain, and the right hand column the defining attributes.

Table 103: Allowed Attributes by Domain

DOMAIN (HIERARCHY)	ATTRIBUTE
Administration of substance via specific route (procedure)	| Route of administration |
Anatomical structure (body structure)	| Laterality |
	| Part of |

DOMAIN (HIERARCHY)	ATTRIBUTE
Clinical finding (finding)	| After |
	| Associated morphology |
	| Associated with |
	| Causative agent |
	| Clinical course |
	| Due to |
	| Episodicity |
	| Finding informer |
	| Finding method |
	| Finding site |
	| Has interpretation |
	| Interprets |
	| Laterality | [10]
	| Occurrence |
	| Pathological process |
	| Severity |
Disorder (finding)	| Has definitional manifestation |
Drug delivery device (physical object)	| Has active ingredient |
	| Has dose form |

[10] This attribute is allowed only in close-to-user form. It is used for user-level composition, and is not applied directly as defining attributes in the distributed form (or in normal forms).

DOMAIN (HIERARCHY)	ATTRIBUTE
Evaluation procedure (procedure)	l Component l
	l Has specimen l
	l Measurement method l
	l Property l
	l Scale type l
	l Time aspect l
Event (event)	l After l
	l Associated with l
	l Causative agent l
	l Due to l
	l Occurrence l
Pharmaceutical / biologic product (product)	l Has active ingredient l
	l Has dose form l

DOMAIN (HIERARCHY)	ATTRIBUTE
Procedure (procedure)	| Access |
	| Direct device |
	| Direct morphology |
	| Direct substance |
	| Has focus |
	| Has intent |
	| Indirect device |
	| Indirect morphology |
	| Method |
	| Priority |
	| Procedure device |
	| Procedure morphology |
	| Procedure site |
	| Procedure site - Direct |
	| Procedure site - Indirect |
	| Recipient category |
	| Revision status |
	| Using device |
	| Using access device |
	| Using energy |
	| Using substance |
Situation with explicit context (situation)	| Subject relationship context |
	| Temporal context |

DOMAIN (HIERARCHY)	**ATTRIBUTE**
Finding with *explicit context* (situation) - *descendants* only	| Associated finding |
Finding with *explicit context* (situation) - self and *descendants*	| Finding context |
Procedure with *explicit context* (situation) - *descendants* only	| Associated procedure |
Procedure with *explicit context* (situation) - self and *descendants*	| Procedure context |
Specimen (specimen)	| Specimen procedure |
	| Specimen source identity |
	| Specimen source morphology |
	| Specimen source topography |
	| Specimen substance |
Surgical procedure (procedure)	| Surgical Approach |

Table 104: Historical Relationships by Domain

DOMAIN	**HISTORICAL REALATIONSHIP**
Ambiguous *Concept*	MAYBE A
Duplicate *Concept*	SAME AS
Erroneous *Concept*	REPLACED BY
	WAS A
Inactive reason Not Stated *Concept*	REPLACED BY
	WAS A
Limited *Status Concept*	WAS A
Moved From Elsewhere *Concept*	MOVED FROM
Moved To Elsewhere *Concept*	MOVED TO

DOMAIN	HISTORICAL REALATIONSHIP
Outdated *Concept*	REPLACED BY
	WAS A
Pending Move Concept	MOVED TO

6.2.2.1.2 Allowable Ranges

The following table contains the allowable Values (*Ranges*) that can be applied to each *Attribute*. Note that each item in the *Attribute* column refers to a single *concept* that resides in the *Attribute hierarchy*.

Table 105: Allowable Ranges for Concept Model Attributes

ATTRIBUTE	RANGE
I ACCESS I	I Surgical access values I 309795001 (<=)(< Q)
I AFTER I	I Clinical Finding I 404684003 (<<) I Procedure I 71388002 (<<)
I ASSOCIATED FINDING I	I Clinical finding I 404684003 (<=)(< Q) I Event I 272379006 (<=)(< Q) I Observable entity I 363787002 (< Q only) I Link assertion I 416698001 (< Q only) I Procedure I 71388002 (< Q only)
I ASSOCIATED MORPHOLOGY I	I Morphologically abnormal structure I 49755003 (<<)
I ASSOCIATED PROCEDURE I	I Procedure I 71388002 (<=)(< Q) I Observable entity I 363787002 (< Q only)
I ASSOCIATED WITH I	I Clinical Finding I 404684003 (<<) I Procedure I 71388002 (<<) I Event I 272379006 (<<) I Organism I 410607006 (<<) I Substance I 105590001 (<<) I Physical object I 260787004 (<<) I Physical force I 78621006 (<<) I Pharmaceutical / biologic product I 373873005 (<< Q only) I SNOMED CT Concept I 138875005 (==)

ATTRIBUTE	RANGE
| CAUSATIVE AGENT |	| Organism | 410607006 (<<) | Substance | 105590001 (<<) | Physical object | 260787004 (<<) | Physical force | 78621006 (<<) | Pharmaceutical / biologic product | 373873005 (<< Q only) | SNOMED CT Concept | 138875005 (==)
| COMPONENT |	| Substance | 105590001 (<=)(< Q) | Observable entity | 363787002 (<=)(< Q) | Cell structure | 4421005 (<=)(< Q) | Organism | 410607006 (<=)(< Q)
| CLINICAL COURSE |	| Courses | 288524001 (<=)(< Q)
| DIRECT DEVICE |	| Device | 49062001 (<<)
| DIRECT MORPHOLOGY |	| Morphologically abnormal structure | 49755003 (<<)
| DIRECT SUBSTANCE |	| Substance | 105590001 (<<) | Pharmaceutical / biologic product | 373873005 (<<)
| DUE TO |	| Clinical Finding | 404684003 (<=) | Event | 272379006 (<=)
| EPISODICITY |	| Episodicities | 288526004 (<=)(< Q)
| FINDING CONTEXT |	| Finding context value | 410514004 (<=)(< Q)
| FINDING INFORMER |	| Performer of method | 420158005 (<<) | Subject of record or other provider of history | 419358007 (<<)
| FINDING METHOD |	| Procedure | 71388002 (<=)
| FINDING SITE |	| Anatomical or acquired body structure | 442083009 (<<)
| HAS ACTIVE INGREDIENT |	| Substance | 105590001 (<<)
| HAS DEFINITIONAL MANIFESTATION |	| Clinical finding | 404684003 (<<)
| HAS DOSE FORM |	| Type of drug preparation | 105904009 (<<)

ATTRIBUTE	RANGE
I HAS FOCUS I	I Clinical finding I 404684003 (<<) I Procedure I 71388002 (<<)
I HAS INTENT I	I Intents (nature of procedure values) I 363675004 (<=)
I HAS INTERPRETATION I	I Findings values I 260245000 (<<)
I HAS SPECIMEN I	I Specimen I 123038009 (<=)(< Q)
I INDIRECT DEVICE I	I Device I 49062001 (<<)
I INDIRECT MORPHOLOGY I	I Morphologically abnormal structure I 49755003 (<<)
I INTERPRETS I	I Observable entity I 363787002 (<<) I Laboratory procedure I 108252007 (<<) I Evaluation procedure I 386053000 (<<)
I LATERALITY I	I Side I 182353008 (<=)
I MEASUREMENT METHOD I	I Laboratory procedure categorized by method I 127789004(<=)
I METHOD I	I Action I 129264002 (<<)
I OCCURRENCE I	I Periods of life I 282032007 (<)
I PATHOLOGICAL PROCESS I	I Autoimmune I 263680009 (==) I Infectious process I 441862004 (<<) I Hypersensitivity process I 472963003 (< <)
I PRIORITY I	I Priorities I 272125009 (<=)(< Q)
I PROCEDURE CONTEXT I	I Context values for actions I 288532009 (<=)(< Q)
I PROCEDURE DEVICE I	I Device I 49062001 (<<)
I PROCEDURE MORPHOLOGY I	I Morphologically abnormal structure I 49755003 (<<)
I Direct morphology I	I Morphologically abnormal structure I 49755003 (<<)
I Indirect morphology I	I Morphologically abnormal structure I 49755003 (<<)
I PROCEDURE SITE I	I Anatomical or acquired body structure I 442083009 (<<)

ATTRIBUTE	**RANGE**
I Procedure site - Direct I	I Anatomical or acquired body structure I 442083009 (<<)
I Procedure site - Indirect I	I Anatomical or acquired body structure I 442083009 (<<)
I PROPERTY I	I Property of measurement I 118598001 (<=)(< Q)
I RECIPIENT CATEGORY I	I Person I 125676002 (<<) I Family I 35359004 (<<) I Community I 133928008 (<<) I Donor for medical or surgical procedure I 105455006 (<<) I Group I 389109008 (<<)
I REVISION STATUS I	I Primary operation I 261424001 (<<) I Revision - value I 255231005 (<<) I Part of multistage procedure I 257958009 (<<)
I ROUTE OF ADMINISTRATION I	I Route of administration value I 284009009 (<<)
I SCALE TYPE I	I Quantitative I 30766002 (<<) I Qualitative I 26716007 (<<) I Ordinal value I 117363000 (<<) I Ordinal or quantitative value I 117365007 (<<) I Nominal value I 117362005 (<<) I Narrative value I 117364006 (<<) I Text value I 117444000 (<<)
I SEVERITY I	I Severities I 272141005 (<=)(< Q)
I SPECIMEN PROCEDURE I	I Procedure I 71388002 (<)
I SPECIMEN SOURCE IDENTITY I	I Person I 125676002 (<<) I Family I 35359004 (<<) I Community I 133928008 (<<) I Device I 49062001 (<<) I Environment I 276339004 (<<)
I SPECIMEN SOURCE MORPHOLOGY I	I Morphologically abnormal structure I 49755003 (<<)

ATTRIBUTE	RANGE
I SPECIMEN SOURCE TOPOGRAPHY I	I Anatomical or acquired body structure I 442083009 (<<)
I SPECIMEN SUBSTANCE I	I Substance I 105590001 (<<)
I SUBJECT RELATIONSHIP CONTEXT I	I Person I 125676002 (<=)(< Q)
I SURGICAL APPROACH I	I Procedural approach I 103379005 (<=)(< Q)
I TEMPORAL CONTEXT I	I Temporal context value I 410510008 (<=)(< Q)
I TIME ASPECT I	I Time frame I 7389001 (<=)(< Q)
I USING ACCESS DEVICE I	I Device I 49062001 (<<)
I USING DEVICE I	I Device I 49062001 (<<)
I USING ENERGY I	I Physical force I 78621006 (<<)
I USING SUBSTANCE I	I Substance I 105590001 (<<)

Meaning of Allowable Values (*Range*) notations:

(<<) this code and *descendants*,
(<) *descendants* only,
(<=) *descendants* only (stated) except for supercategory groupers,
(==) this code only,
(< Q) *descendants* only when in a qualifying *Relationship*,
(< Q only) *descendants* only, and only allowed in a qualifying *Relationship*.

A supercategory grouper is sufficiently defined by reference to a value that is at the top of the value hierarchy, resulting in a very general meaning, such that the code is less useful (or sometimes useless) for record entry, but is useful as an organizer of the hierarchy.

6.2.2.2 Attribute Hierarchies in SNOMED CT

Selected *SNOMED CT* attributes have a hierarchical *relationship* to one another known as "attribute hierarchies". In an attribute *hierarchy*, one general attribute is the parent of one or more specific *subtypes* of that attribute. *Concepts* defined using the more general attribute can inherit *concepts* modeled with the more specific *subtypes* of that attribute.

6.2.2.2.1 Attribute hierarchies used in modeling Procedures

Three groups of attributes are organized as a simple two-level *hierarchy*. The three top level attributes are I PROCEDURE SITE I, I PROCEDURE DEVICE I, and I PROCEDURE MORPHOLOGY I. Each has a sub-attribute to represent the direct object, and another to represent the indirect object. In addition, I PROCEDURE DEVICE I can be specialized by the attributes I USING DEVICE I and I USING ACCESS DEVICE I.

I PROCEDURE DEVICE I attribute *hierarchy*:

- I PROCEDURE DEVICE I

  - I DIRECT DEVICE I
  - I INDIRECT DEVICE I
  - I USING DEVICE I
  - I USING ACCESS DEVICE I

I PROCEDURE MORPHOLOGY I attribute *hierarchy*:

- I PROCEDURE MORPHOLOGY I
  - I DIRECT MORPHOLOGY I
  - I INDIRECT MORPHOLOGY I

I PROCEDURE SITE I attribute *hierarchy*:

- I PROCEDURE SITE I
  - I PROCEDURE SITE - DIRECT I
  - I PROCEDURE SITE - INDIRECT I

6.2.2.2.2 Attribute hierarchy used in modeling Clinical Findings

I ASSOCIATED WITH I attribute *hierarchy*:

- I ASSOCIATED WITH I
  - I AFTER I
  - I DUE TO I
  - I CAUSATIVE AGENT I

6.2.2.3 Relationship groups in SNOMED CT

Multiple attributes and their values can be grouped together into "*Relationship groups*" to add clarity to *concept* definitions. A *Relationship group* combines an *attribute-value pair* with one or more other *attribute-value pairs*. *Relationship groups* originated to add clarity to I Clinical finding I *concepts* which require multiple I ASSOCIATED MORPHOLOGY I attributes and multiple I FINDING SITE I attributes and to I Procedure I which require multiple I METHOD I attributes and multiple I PROCEDURE SITE I attributes. However, *Relationship groups* are not limited to I Clinical finding I and I Procedure I *concepts*.

In the case of I Procedure I, *Relationship groups* generally associate the correct method with the correct site. In the example below, the *Relationship groups* clarify that there is exploration of the bile duct, and excision of the gall bladder. Without *Relationship groups*, the four attributes would be ungrouped and it would be unclear whether the excision was of the bile duct or of the gall bladder.

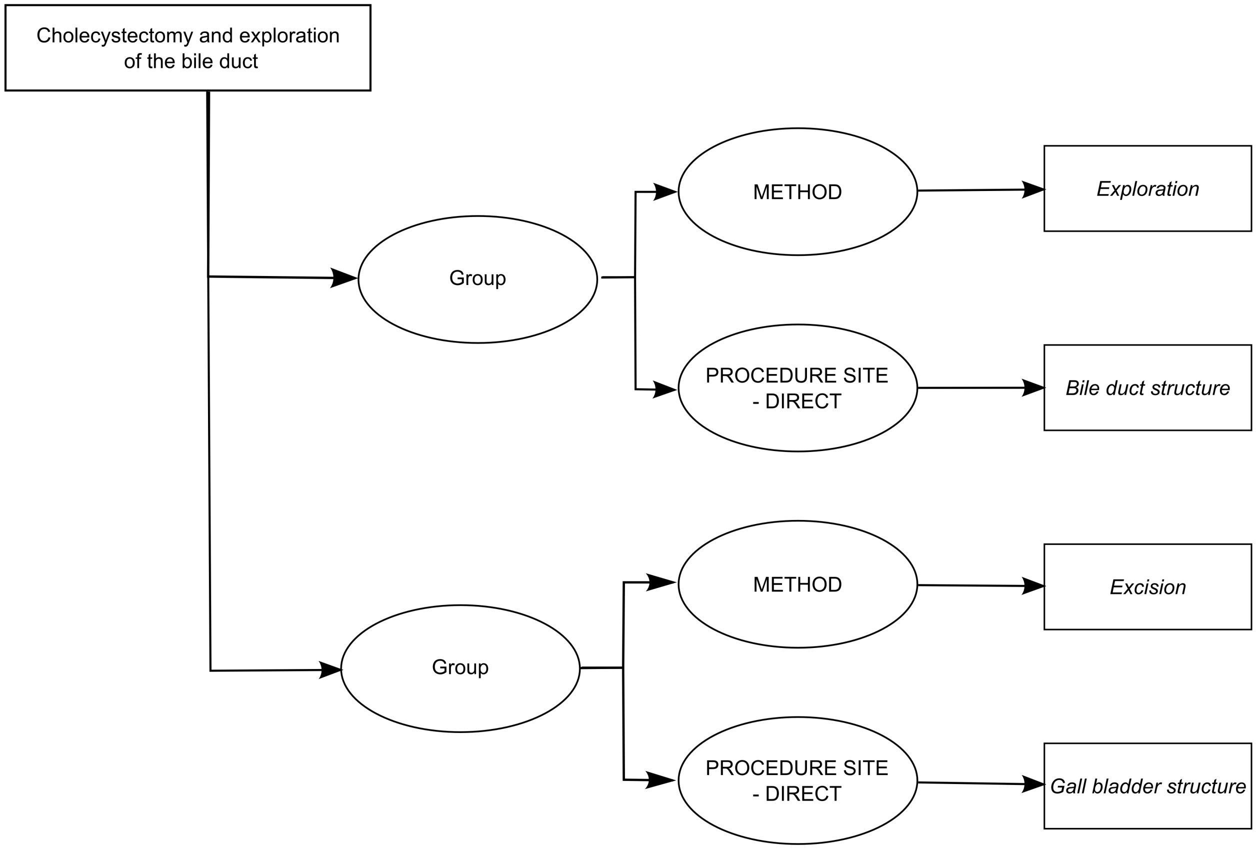


Figure 58: Example Cholecystectomy and exploration of bile duct

6.2.3 Hierarchies

SNOMED CT concepts are organized into hierarchies. There is one special *concept* referred to as the | Root Concept Code | . It represents the "root" of the hierarchy that contains all *Concepts* in *SNOMED CT*. The root named "*SNOMED CT Concept*" subsumes (is the supertype of) the top-level *concepts* (hierarchies parents) and all the *concepts* beneath them (their *subtypes*). As the hierarchies are descended, the *concepts* within them become increasingly specific (or granular). A brief description of the content in each *hierarchy* is given below.

Subtype (or "*child*") *concepts* are the *descendant concepts* of Supertype (or "parent") *concepts*.

Example: | Streptococcal arthritis (disorder) | is a *subtype* of | Bacterial arthritis (disorder) |.

Supertype *concepts* are the *ancestor concepts* of *Subtype concepts*.

Example: | Bacterial arthritis (disorder) | is a supertype of | Streptococcal arthritis (disorder) |.

6.2.3.1 Summary of Top Level Hierarchies

6.2.3.1.1 Top Level Concepts

Table 106: Top Level Concepts

• *| Clinical finding |* • *| Procedure |* • *| Observable entity |* • *| Body structure |* • *| Organism |* • *| Substance |* • *| Pharmaceutical / biologic product |* • *| Specimen |* • *| Special concept |* • *| SNOMED CT Model Component |*	• *| Physical force |* • *| Event |* • *| Environment or geographical location |* • *| Social context |* • *| Situation with explicit context |* • *| Staging and scales |* • *| Physical object |* • *| Qualifier value |* • *| Record artifact |*

6.2.3.1.2 Top Level Metadata

Table 107: Top Level Metadata

• | Core metadata concept | • | Foundation metadata concept | • | Linkage concept | • | Namespace concept |

6.2.3.2 Clinical finding

Concepts in this *hierarchy* represent the result of a clinical observation, assessment or judgment , and include both normal and abnormal clinical states.

Examples of Clinical finding concepts:

- | Clear sputum (finding) | ;
- | Normal breath sounds (finding) | ;
- | Poor posture (finding) | .

The | Clinical finding | *hierarchy* contains the *sub-hierarchy* of | Disease |. *Concepts* that are *descendants* of | Disease | (or disorders) are always and necessarily abnormal clinical states. The *subtype polyhierarchy* allows diseases to be *subtypes* of other disorders as well as *subtypes* of findings.

Examples of Disease concepts:

- | Tuberculosis (disorder) | ;
- | non-Hodgkin's lymphoma (disorder) | .

Note: See also *Attributes used to define Clinical Finding concepts*.

6.2.3.2.1 Clinical Finding - definition

Clinical findings have been defined as observations, judgments or assessments about patients. The problem with the terms *finding* and *observation* is that they seem to refer to the judgment of the observer rather than to the actual state of the body. Organism state has been suggested as a more neutral name, but it would need to be delimited from a course of disease. Examples of clinical findings include: difficulty swallowing, nose bleed, diabetes, headache, and so forth. More precise and reproducible definitions of clinical findings, and the precise boundaries between findings and events, between findings and observables, between findings and situations, and the distinction between finding and disorder, remain ongoing challenges at the margins. The distinction between a disorder and an observation has proven to be difficult to define in a reproducible manner across the tens of thousands of *concepts* included under clinical findings. Nevertheless, there are several reliable characteristics of each sub-category (disorders and findings):

6.2.3.2.1.1 Disorders

1. Disorders necessarily are abnormal.
2. They have temporal persistence, with the (at least theoretical) possibility of their manifestations being treated, in remission, or quiescent even though the disorder itself still present.
3. They necessarily have an underlying pathological process.

6.2.3.2.1.2 Findings

1. Findings may be normal (but not necessarily); no disorders may.
2. Some findings may exist only at a single point in time (e.g. a serum sodium level); no disorders may.
3. Findings cannot be *temporally* separate from the observing of them (you can't observe them and say they are absent, nor can you have the finding present when it is not capable of being observed).
4. They cannot be defined in terms of an underlying pathological process that is present even when the observation itself is not present.

Disorders may be present as a propensity for certain abnormal states to occur, even when treatment mitigates or resolves those abnormal states. In some cases the disease process is irrefutable, e.g. meningococcal meningitis. In others an underlying disease process is assumed based on the temporal and causal association of the disorder and its manifestation, e.g. nystagmus disorder is different from the finding/observation of nystagmus, which can be a normal physiological response to rotation of the head. If you spin around and around and then have nystagmus (the finding) you still do not have nystagmus disorder. And someone can have a nystagmus disorder without currently manifesting nystagmus. Similarly, deafness disorder is different from the symptom (observation) of reduced hearing, which can be due to a number of temporary causes such as excessive ear wax.

6.2.3.2.2 Attributes used to define Clinical Finding concepts

Table 108: Approved Clinical Finding attributes summary

<table>
<tr><th>Defining Attribute</th><th>Subsumed Attribute</th><th>Allowable Values</th></tr>
<tr><td colspan="2">| FINDING SITE |</td><td>| Anatomical or acquired body structure | 442083009 (<<)</td></tr>
<tr><td colspan="2">| ASSOCIATED MORPHOLOGY |</td><td>| Morphologically abnormal structure | 49755003 (<<)</td></tr>
</table>

Defining Attribute	**Subsumed Attribute**	**Allowable Values**
I ASSOCIATED WITH I		I Clinical Finding I 404684003 (<<) I Procedure I 71388002 (<<) I Event I 272379006 (<<) I Organism I 410607006 (<<) I Substance I 105590001 (<<) I Physical object I 260787004 (<<) I Physical force I 78621006 (<<) I Pharmaceutical / biologic product I 373873005 (<< Q only) I SNOMED CT Concept I 138875005 (==)
	I CAUSATIVE AGENT I	I Organism I 410607006 (<<) I Substance I 105590001 (<<) I Physical object I 260787004 (<<) I Physical force I 78621006 (<<) I Pharmaceutical / biologic product I 373873005 (<< Q only) I SNOMED CT Concept I 138875005 (==)
	I DUE TO I	I Clinical Finding I 404684003 (<=) I Event I 272379006 (<=)
	I AFTER I	I Clinical Finding I 404684003 (<<) I Procedure I 71388002 (<<)
I SEVERITY I		I Severities I 272141005 (<=)(< Q)
I CLINICAL COURSE I		I Courses I 288524001 (<=)(< Q)
I EPISODICITY I		I Episodicities I 288526004 (<=)(< Q)
I INTERPRETS I		I Observable entity I 363787002 (<<) I Laboratory procedure I 108252007 (<<) I Evaluation procedure I 386053000 (<<)
I HAS INTERPRETATION I		I Findings values I 260245000 (<<)
I PATHOLOGICAL PROCESS I		I Autoimmune I 263680009 (==) I Infectious process I 441862004 (<<) I Hypersensitivity process I 472963003 (< <)

Defining Attribute	Subsumed Attribute	Allowable Values
I HAS DEFINITIONAL MANIFESTATION I		I Clinical finding I 404684003 (<<)
I OCCURRENCE I		I Periods of life I 282032007 (<)
I FINDING METHOD I		I Procedure I 71388002 (<=)
I FINDING INFORMER I		I Performer of method I 420158005 (<<) I Subject of record or other provider of history I 419358007 (<<)

Note:

Meaning of Allowable Values (*Range*) notations:

(<<) this code and *descendants*,
(<) *descendants* only,
(<=) *descendants* only (stated) except for supercategory groupers,
(==) this code only,
(< Q) *descendants* only when in a qualifying *Relationship*,
(< Q only) *descendants* only, and only allowed in a qualifying *Relationship*.

Note: See also *Clinical finding*.

6.2.3.2.2.1 FINDING SITE

This attribute specifies the body site affected by a condition.

Table 109: Permissible values for FINDING SITE

Attribute Values	**Examples**
I Anatomical or acquired body structure I 442083009 (<<)	I Kidney disease (disorder) I • I FINDING SITE I I Kidney structure (body structure) I
	I Appendicitis (disorder) I • I FINDING SITE I I Appendix structure (body structure) I

6.2.3.2.2.2 ASSOCIATED MORPHOLOGY

This attribute specifies the morphologic changes seen at the tissue or cellular level that are characteristic features of a disease.

Table 110: Permissible values for ASSOCIATED MORPHOLOGY

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td rowspan="2">| Morphologically abnormal structure | 49755003 (<<)</td><td>| Bone marrow hyperplasia (disorder) |
• | ASSOCIATED MORPHOLOGY | | Hyperplasia (morphologic abnormality) |</td></tr>
<tr><td>| Pancreatitis (disorder) |
• | ASSOCIATED MORPHOLOGY | | Inflammation (morphologic abnormality) |</td></tr>
</table>

6.2.3.2.2.3 ASSOCIATED WITH

This attribute asserts an interaction between two *concepts* beyond simple co-occurrence in the patient. | ASSOCIATED WITH | represents a clinically relevant association between *concepts* without either asserting or excluding a causal or sequential *relationship* between the two.

Table 111: Permissible values for ASSOCIATED WITH

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td>| Clinical Finding | 404684003 (<<)
| Procedure | 71388002 (<<)
| Event | 272379006 (<<)
| Organism | 410607006 (<<)
| Substance | 105590001 (<<)
| Physical object | 260787004 (<<)
| Physical force | 78621006 (<<)
| Pharmaceutical / biologic product | 373873005 (<< Q only)
| SNOMED CT Concept | 138875005 (==)</td><td></td></tr>
</table>

| ASSOCIATED WITH | subsumes the following, more specific, attributes in what is called an attribute *hierarchy* (explained in *Attribute Hierarchies in SNOMED CT* on page 272):

- | AFTER |
- | DUE TO |
- | CAUSATIVE AGENT |

6.2.3.2.2.4 AFTER

This attribute is used to model *concepts* in which a clinical finding occurs after another clinical finding or procedure. Neither asserting nor excluding a causal *relationship*, it instead emphasizes a sequence of events.

Table 112: Permissible values for AFTER

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td>| Clinical Finding | 404684003 (<<)
| Procedure | 71388002 (<<)</td><td>| Post-viral disorder (disorder) |
• | AFTER | | Viral disease (disorder) |</td></tr>
</table>

This example can be paraphrased as: “every post-viral disorder occurs after some viral disease”.

6.2.3.2.2.5 DUE TO

This attribute is used to relate a | Clinical finding | directly to its cause. If a clinical finding merely predisposes to or worsens another disorder, rather than causing it directly, then the more general attribute | ASSOCIATED WITH | is used instead.

Table 113: Permissible values for DUE TO

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td>| Clinical Finding | 404684003 (<=)
| Event | 272379006 (<=)</td><td>| Cheilitis due to atopic dermatitis (disorder) |
• | IS A | | Cheilitis (disorder) |
• | DUE TO | | Atopic dermatitis (disorder) |</td></tr>
</table>

6.2.3.2.2.6 CAUSATIVE AGENT

This attribute identifies the direct causative agent of a disease. It does not include vectors, e.g. a mosquito that transmits malaria.

Table 114: Permissible values for CAUSATIVE AGENT

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td rowspan="3">| Organism | 410607006 (<<)
| Substance | 105590001 (<<)
| Physical object | 260787004 (<<)
| Physical force | 78621006 (<<)
| Pharmaceutical / biologic product | 373873005 (<< Q only)
| SNOMED CT Concept | 138875005 (==)</td><td>| Bacterial endocarditis (disorder) |
• | CAUSATIVE AGENT | | Superkingdom Bacteria (organism) |</td></tr>
<tr><td>| Fentanyl allergy (disorder) |
• | CAUSATIVE AGENT | | Fentanyl (substance) |</td></tr>
<tr><td>| Electrical burn of skin (disorder) |
• | CAUSATIVE AGENT | | Electricity (physical force) |</td></tr>
</table>

6.2.3.2.2.7 SEVERITY

This attribute is used to subclass a | Clinical finding | *concept* according to its severity; however, caution is encouraged because this use is said to be *relative*. By relative, it is meant that it is incorrect to assume that the same degree of disease intensity or hazard is implied for all | Clinical finding | to which this attribute is applied. There are three reasons.

First, “severe” could be interpreted differently depending on what other values are available to choose for severity. Thus severity is relative to the other values in the *value set* presented to users. Consider the different meaning of severity in each of the following three sets of values:

- mild / moderate / severe
- minimal / mild / moderate / severe / very severe
- mild / mild to moderate / moderate / moderate to severe / severe / life threatening / fatal

Second, the severity is defined relative to the expected degree of intensity or hazard of the | Clinical finding | that is being qualified. A common cold has a baseline intensity or hazard much less than that of a more serious disease like lupus erythematosus or pneumonia; thus a severe cold might be considered less intense or hazardous than a mild pneumonia.

Third, some disorders that are life-threatening do not ordinarily have a severity assigned to them. Cancer, for example, is generally not subclassed according to mild, moderate and severe types, but rather is subclassed according to stage or grade.

For these reasons, the | SEVERITY | attribute cannot be relied on to retrieve all *Clinical findings* with serious or life-threatening import. Nevertheless, it is still useful for subclassing certain *concepts* and differentiating between different severities of a single disorder. SEVERITY is not used to model any *concepts precoordinated* in the *International Release* but it can still be used in *postcoordination* as a *qualifier*.

Table 115: Permissible values for SEVERITY

Attribute Values	**Examples**
| Severities | 272141005 (<=)(< Q)	

6.2.3.2.2.8 CLINICAL COURSE

This attribute is used to represent both the course and onset of a disease. Many conditions with an acute (sudden) onset also have an acute (short duration) course. Few diseases with a chronic (long -*term*) course would need to have their onset sub-divided into rapid or gradual *subtypes*, and thus there is no clear need for separating the rapidity of onset from the duration of a disease; based on testing by implementers and *modelers*, a single attribute with values that combine these meanings has clearly been more reproducible and useful than two attributes that attempt to separate the meanings.

Table 116: Permissible values for CLINICAL COURSE

Attribute Values	**Examples**
| Courses | 288524001 (<=)(< Q)	| Acute amebic dysentery (disorder) | • | CLINICAL COURSE | | Sudden onset AND/OR short duration (qualifier value) |
	| Chronic fibrosing pancreatitis (disorder) | • | CLINICAL COURSE | | Chronic (qualifier value) |

The word acute has more than one meaning, and the meanings are often overlapping or unclear. The word acute may imply rapid onset, short duration, or high severity; in some circumstances it might be used to mean all of these. For morphological *terms* it may also imply the kind of morphology associated with the speed of onset. | Acute inflammation (morphologic abnormality) | does not necessarily have CLINICAL COURSE | Sudden onset AND/OR short duration |, but rather implies polymorphonuclear infiltration; likewise | Chronic inflammation (morphologic abnormality) | implies mononuclear cell infiltration, not necessarily a chronic course, although inflammation with a chronic course is highly correlated with a lymphocytic infiltration.

6.2.3.2.2.9 EPISODICITY

| EPISODICITY | is used to represent episodes of care provided by a physician or other care provider, typically a general practitioner, *not* episodes of disease experienced by the patient. See *EPISODICITY no longer modeled in active content* on page 293, regarding the origin of the attribute. For example, asthma with | EPISODICITY |=| first episode | represents the first *time* the patient presents to their health care provider with asthma. EPISODICITY is not used to model any *concepts precoordinated* in the *International Release* but it can still be used in *postcoordination* as a *qualifier*.

Table 117: Permissible values for EPISODICITY

Attribute Values	**Examples**
| Episodicities | 288526004 (<=)(< Q)	

6.2.3.2.2.10 INTERPRETS

This attribute refers to the entity being evaluated or interpreted, when an evaluation, interpretation or " judgment" is intrinsic to the meaning of a *concept*. This attribute is usually grouped with the | HAS INTERPRETATION | attribute.

Table 118: Permissible values for INTERPRETS

Attribute Values	**Examples**
| Observable entity | 363787002 (<<) | Laboratory procedure | 108252007 (<<) | Evaluation procedure | 386053000 (<<)	| Decreased muscle tone (finding) | • | INTERPRETS | | muscle tone (observable entity) | • | HAS INTERPRETATION | | Decreased (qualifier value) |
	| Abnormal glucose level (finding) | • | INTERPRETS | | Glucose measurement (procedure) | • | HAS INTERPRETATION | | Outside reference range (qualifier value) |

Note: For *concepts* in the Measurement finding subhierarchy, the value for | INTERPRETS |should be an Evaluation procedure or a Laboratory procedure rather than an Observable entity.

6.2.3.2.2.11 HAS INTERPRETATION

This attribute is grouped with the attribute | INTERPRETS |, and designates the judgment aspect being evaluated or interpreted for a *concept* (e.g., presence, absence, degree, normality, abnormality, etc.).

Table 119: Permissible values for HAS INTERPRETATION

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td rowspan="2">| Findings values | 260245000 (<<)</td><td>| Decreased muscle tone (finding) |
• | INTERPRETS | | Muscle tone (observable entity) |
• | HAS INTERPRETATION | | Decreased (qualifier value) |</td></tr>
<tr><td>| Abnormal glucose level (finding) |
• | INTERPRETS | | Glucose measurement (procedure) |
• | HAS INTERPRETATION | | Outside reference range (qualifier value) |</td></tr>
</table>

6.2.3.2.2.12 PATHOLOGICAL PROCESS

This attribute provides information about the underlying pathological process for a disorder, but only when the results of that process are not structural and cannot be represented by the | ASSOCIATED MORPHOLOGY | attribute.

The values | Infectious process (qualifier value) | and its *subtype* | Parasitic process (qualifier value) | are included in the range for | PATHOLOGICAL PROCESS |. These were added to accommodate the change in the modeling of *concepts* in the | Infectious disease (disorder) | subhierarchy where the infectious aspect of the disease is represented using | PATHOLOGICAL PROCESS |.

Table 120: Permissible values for PATHOLOGICAL PROCESS

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td rowspan="2">| Autoimmune | 263680009 (==)
| Infectious process | 441862004 (<<)
| Hypersensitivity process | 472963003 (< <)</td><td>| Autoimmune parathyroiditis (disorder) |
• | PATHOLOGICAL PROCESS | | Autoimmune (qualifier value) |</td></tr>
<tr><td>| Disease caused by parasite (disorder) |
• | PATHOLOGICAL PROCESS | | Parasitic process (qualifier value) |</td></tr>
</table>

Pathological process must not be used for values that could overlap with | ASSOCIATED MORPHOLOGY |. Inflammatory processes result in inflammation (by definition), but these disorders should be defined using their morphology.

6.2.3.2.2.13 HAS DEFINITIONAL MANIFESTATION

This attribute links disorders to the manifestations (observations) that define them. It can only be applied to disorders.

Table 121: Permissible values for HAS DEFINITIONAL MANIFESTATION

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td rowspan="2">| Clinical finding | 404684003 (<<)</td><td>| Seizure disorder (disorder) |
• | HAS DEFINITIONAL MANIFESTATION | | Seizure (finding) |</td></tr>
<tr><td>| Hypertensive disorder, systemic arterial (disorder) |
• | HAS DEFINITIONAL MANIFESTATION | | Finding of increased blood pressure (finding) |</td></tr>
</table>

6.2.3.2.2.14 OCCURRENCE

This attribute refers to the specific period of life during which a condition first presents. Multiple values of | OCCURRENCE | for a single *concept* are not desirable, and these will be addressed in a future release. This does not mean the condition cannot persist beyond the period of life in which it first presents.

Table 122: Permissible values for OCCURRENCE

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td>| Periods of life | 282032007 (<)</td><td>| Childhood phobic anxiety disorder (disorder) |
• | OCCURRENCE | | Childhood (qualifier value) |</td></tr>
</table>

6.2.3.2.2.15 FINDING METHOD

This attribute specifies the means by which a clinical finding was determined. This attribute is frequently used in conjunction with | FINDING INFORMER |. Findings that specify that they were determined by examination of the patient (e.g. | On examination - ankle clonus (finding) |) should have a value for both | FINDING METHOD | and | FINDING INFORMER |.

Table 123: Permissible values for FINDING METHOD

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td>| Procedure | 71388002 (<=)</td><td>| Finding by palpation (finding) |
• | FINDING METHOD | | Palpation (procedure) |</td></tr>
</table>

6.2.3.2.2.16 FINDING INFORMER

This attribute specifies the person or other entity from which the clinical finding information was obtained. This attribute is frequently used in conjunction with | FINDING METHOD |.

Table 124: Permissible values for FINDING INFORMER

Attribute Values	**Examples**
| Performer of method | 420158005 (<<) | Subject of record or other provider of history | 419358007 (<<)	| Complaining of a headache (finding) | • | FINDING INFORMER | | Subject of record or other provider of history (person) |
	| On examination - ankle clonus (finding) | • | FINDING INFORMER | | Performer of method (person) |

It is accepted that an information model should permit identification of a particular individual who provides information; | FINDING INFORMER | is not about the particular individual. It is about the *category or type* of informer, which is used to differentiate self-reported symptoms from provider-observed signs. Granted, this permits inclusion of epistemology-loaded *terms* (cf. Bodenreider el al., FOIS 2004), but health care is full of such *terms*, and they are (or at least can be) understandable, reproducible and useful.

6.2.3.2.3 Specific disorder types

6.2.3.2.3.1 Ischemic disorders

Ischemic disorders are defined by a morphology of ischemic structural change.
This need not be permanent structural change, but it is assumed that all ischemia results in some structural alterations at the molecular level, at least, even if reversible.

6.2.3.2.3.1.1 Ischemic heart disease

Ischemic heart disease includes myocardial infarction, myocardial ischemia (without infarction), angina, and other disorders of the heart that have ischemic structural change (reversible or non-reversible) as a *defining characteristic*.

Coronary arteriosclerosis can of course be present without causing ischemia, so | coronary arteriosclerosis | is not a *subtype* of | ischemic heart disease |.

Likewise there are causes of myocardial ischemia and infarction other than coronary arteriosclerosis, so | ischemic heart disease | is not a *subtype* of | coronary arteriosclerosis |.

6.2.3.2.3.2 Cardiac Arrhythmia

Cardiologists noted that there had been confusion in the placement of | conduction disorder of the heart | as a broad grouper that subsumed arrhythmias and heart blocks, whereas the common understanding of "arrhythmia" is that this term refers to a broad set of conditions that include conduction disorders, under which are heart blocks. Consequently, a new concept | Cardiac arrhythmia (disorder) | was created as a parent of | Conduction disorder of the heart (disorder) |, and as the active referent of the inactive codes named "dysrhythmia" or "arrhythmia".

Conduction disorders include heart block, AV block, bundle branch block, conduction delay, and conduction defect. Other arrhythmias were moved out from under | Conduction disorder of the heart (disorder) | and placed under | Cardiac arrhythmia (disorder) |.

6.2.3.2.3.3 Lesion

The word lesion can be used to refer to both structural and functional abnormalities. If a disorder or procedure refers to a lesion in a way that makes it clear that it is a generic term for a **structural** abnormality, then the correct modeling approach is to use | Associated morphology | = *morphologically abnormal structure* (for disorders) or | Procedure morphology | (for procedures).

Functional-only lesions obviously should not be modeled using values from the | morphologically abnormal structure | hierarchy.

6.2.3.2.3.4 Trauma, injury, damage

The word “trauma” has multiple senses. The first distinction is physical damage to the body versus psychic trauma. We assume “trauma” means physical damage unless accompanied by words that make clear it is psychic.

Traumatic injury (disorder) is defined as any disorder with a morphology of "traumatic abnormality". See known issues (below) for a discussion of the known problems with traumatic morphologies.

There is a problem that occurs if we attempt to require “injury” to be synonymous with “trauma” which can be best illustrated by the example of the very common usage of the word “injury” when referring to damage to the brain. An internet search for the phrase "non-traumatic brain injury" will show that this refers to brain damage that is the result of asphyxiation, stroke, drowning, toxic injury, etc., and not due to direct physical impact to the skull (the traumatic brain injuries). We needed a broad category that would allow us to categorize injuries broadly including non-traumatic ones. The *concept* created for this purpose is *traumatic and/or non-traumatic injury (disorder)*.

6.2.3.2.3.4.1 Laceration, incised wound, rupture, traumatic rupture, spontaneous rupture:

The word “lacerated” has two meanings, which can be succinctly summarized as “torn” vs “cut”. Common clinical usage equates “laceration” with “incised wound”. For example, a common emergency room problem is accidental cuts of fingers with kitchen knives. These are routinely called “lacerations”. On the other hand, most dictionaries insist that “laceration” implies a wound with ragged edges as a result of tearing. Obstetrical lacerations carry this latter meaning. When structures are torn or ruptured, the edges are usually irregular.

There are two morphologies with a synonym of “laceration”: "incised wound", and “traumatic rupture”. Modelers must choose which of these two meanings is intended when the word “laceration” or “lacerated” appears in a *concept Description* from the “injury” hierarchy.

More generally, ruptures can occur either as a result of injury or spontaneously. The word “rupture”, when applied to muscles and tendons, implies a traumatic injury (e.g. "rupture of collateral ligament of the knee"). But “rupture” when applied to an internal viscus may be either traumatic or spontaneous (e.g. rupture of aorta, rupture of ovary, etc).

“Rupture” has *subtype* morphologies “traumatic rupture” and “nontraumatic rupture”. It is important to make this distinction, at a minimum, in order to support queries related to the effects of trauma. Modelers should choose “traumatic rupture” as the value of Associated-morphology for *concepts* using the word “rupture” with anatomical sites (such as muscles and tendons) where rupture requires trauma, in the absence of a specific lesion. Modelers should choose “rupture” as the value of Associated-morphology for *concepts* using the word “rupture” with sites (such as internal organs) where both traumatic and spontaneous rupture are seen. Nontraumatic rupture is usually stated to be so, but may also be inferred if the thing rupturing is a lesion which ordinarily leads to spontaneous rupture in the absence of trauma (e.g. rupture of inflamed appendix).

6.2.3.2.3.4.2 Friction injury

An injury due to friction can be represented using a morphology of I Friction injury (morphologic abnormality) I, in which case it will not classify as a kind of wound. Examples of friction injuries that are not wounds would include "abrasion of tooth" and "mechanical irritation". However, most disorders that are named "abrasion" imply that skin or other body structure has been abraded away, and thus they should also be considered wounds. They will correctly classify as wounds after assigning the correct morphology for these skin abrasions, I Abrasion (morphologic abnormality) I.

6.2.3.2.3.5 Death

“Death” is an event, not a disorder. *Concepts* like “relatives died,” "death of companion" go under "life events - finding" which is under I social and personal history finding I.

6.2.3.2.3.5.1 Sudden Cardiac Death

"Sudden cardiac death" is a term used in clinical practice to refer to an arrhythmia that results in sudden loss of cardiac function which, if not quickly reversed, will lead to *actual* death (as opposed to a high risk of imminent death). This *concept* needs an FSN that indicates it is not a kind of death, and it should not be classified under “death” because individuals to whom this label is applied have not necessarily been officially declared dead, and are frequently revived. It is regarded as a *subtype* of "cardiac dysrhythmia."

6.2.3.2.3.6 Tumor vs. Neoplasm

The word "tumor" has two main meanings:

1. a mass, regardless of whether it is neoplastic or not, or
2. a neoplastic mass

Neoplasm is preferred since it is less ambiguous than tumor.

6.2.3.2.3.7 Primary vs secondary neoplastic disorders

SNOMED follows the usage in ICD-O, ICD-9 and *ICD-10*, where secondary malignant neoplasm of (site x) is uniformly interpreted to mean that metastasis has occurred *to* site x. The alternative reading (*from* site x) is not what is intended. If you want to code a metastasis from a lung tumor, then *SNOMED* also has codes that explicitly use the word from, such as 315006004 metastasis from malignant tumor of lung.

Detailed information about metastases (primary at site x, metastatic to site y) could possibly be recorded using one of two different styles: either a style using *two expressions* placed in two statements in the clinical record one statement for the primary and one statement for the secondary; or a style using only *one expression*. The two-*expression* style would be required to code the case with *ICD9* or *ICD10* codes, and it would be valid to use this style with *SNOMED expressions* also.

The one-statement style would have to use one *SNOMED expression* with two role groups each with a morphology and site that is appropriate to the level of detail required.

- 64572001 | disease |:
- { 116676008 | ASSOCIATED MORPHOLOGY |=86049000 | neoplasm, malignant (primary) |
- , 363698007 | FINDING SITE |=76752008 | breast structure |},
- { 116676008 | ASSOCIATED MORPHOLOGY |=14799000 | neoplasm, metastatic |
- , 363698007 | FINDING SITE |=59441001 | lymph node structure |}

Figure 59: An example of the one-statement style: primary malignant neoplasm of breast metastatic to lymph node

The morphology code in the role group differentiates the primary from the secondary site.

This style of modeling, using a single statement/*expression*, does not cleanly permit one to differentiate between *just* a metastasis from the lung, versus *both* a primary lung tumor and a metastasis, at a particular instance/point in time. The recommended solution to this problem is the two-statement style, with a third statement that links them, using the information model to accomplish the linkage, rather than trying to do it all in the terminology. This recommended style permits users to attach time-stamps and other instance identifiers to the primary neoplasm, and separate time-stamps and other instance identifiers to the metastasis. This appears more flexible and semantically robust.

On the other hand, if a user just wants a broad category expressing the site of the metastasis and the site of the primary, in the same statement, the one-*expression* style does allow that, as accommodated by some *precoordinated* codes that are modeled with two role groups as described above.

6.2.3.2.3.8 Neoplasm vs hamartoma

A neoplasm is defined as an abnormal growth of tissue no longer under normal control. A hamartoma is defined as a benign self-limited growth of disorganized mature cells normally found in the region, representing faulty development. Since the cells in hamartomas are mature cells whose growth is under normal control, a hamartoma is not a neoplasm.

SNOMED attempts to sort out the disorder *concepts* that get confused in the area of tumors, neoplasms and hamartomas by making a neoplasm and/or hamartoma disorder category, with five *subtypes*:

1. hamartoma

2. neoplastic disease
3. hemangioma
4. lymphangioma
5. melanocytic nevus

Likewise in the morphologic abnormality hierarchy, we have neoplasm and/or hamartoma with five *subtypes*:

1. hamartoma
2. neoplasm
3. hemangioma - category
4. lymphatic vessel tumor
5. melanocytic nevus - category

Hemangiomas, lymphangiomas and melanocytic nevi can be either hamartomas (these are usually present at birth) or neoplasms (these usually develop later in life). All of the *subtypes* of hemangioma, lymphangioma or melanocytic nevus can thus be aggregated under these upper-level generalizations, immediately under neoplasm and/or hamartoma, without necessarily having to first attempt to (incorrectly) categorize them as either neoplasms or hamartomas.

Nevus

The word "nevus" has many different meanings. The differences generally hinge on answers to the following questions:

1. is it necessarily on the skin? Or can it be located in mucosal sites or other sites?
2. Is it necessarily visible? Or can it be in internal locations such as gastric mucosa, etc?
3. Is it necessarily present at birth? Or can it make its appearance later in life?
4. Is it necessarily dark and made of melanocytes? Or can it be non-pigmented, or made of other types of cells?
5. Is it necessarily made of tissue that is normally present at the site? Or can it be ectopic?
6. Does it exclude benign neoplasms?

Here are some common meanings of nevus based on some combinations of answers to these questions:

1. A birthmark, that is, any visible spot on the skin or oral mucosa present since birth, regardless of tissue of origin, excluding benign neoplasms.
2. Any benign cluster of melanocytes, regardless of location, and regardless of pigmentation, whether present since birth or appearing later.
3. Any cutaneous hamartoma. This excludes non-cutaneous sites, and excludes neoplasms and ectopic tissue such as choristomas.

As a result of this wide variation in meaning, any *SNOMED* FSN containing the word nevus is prone to being ambiguous. For example, consider "vascular nevus". This term might mean:

1. congenital blood vessel tumors in the skin,
2. congenital blood vessel hamartomas or neoplasms that are visible somewhere (not just in the skin, but also including mucosa, whether visible externally or not),
3. congenital blood or lymphatic vessel tumors in the skin,
4. congenital blood or lymphatic vessel hamartomas or neoplasms that are visible somewhere,
5. any of the above but not necessarily congenital

A better FSN for vascular nevus (morphologic abnormality) would be vascular hamartoma (morphologic abnormality). Likewise a better FSN for congenital vascular nevus (disorder) would be congenital vascular hamartoma (disorder). In those cases where common clinical usage of a term containing nevus is unambiguous, there is no call for the term (or *concept* described using the term) to be retired.

6.2.3.2.3.10 Infectious disease vs. Inflammatory disorder

Infectious disease and inflammatory disorder are siblings. It might seem more intuitive for infectious disease to be a child of inflammatory disorder. However, not all infectious disorders are inflammatory. These *concepts* will remain siblings. Infectious disease and its *subtypes* have a CAUSATIVE AGENT *Relationship* to the organism that is infecting. Inflammatory disorder has an | ASSOCIATED MORPHOLOGY | *Relationship* to | inflammation (morphologic abnormality) | or one of its *subtypes*.

6.2.3.2.3.11 Post-infectious disorders

Post-infectious disorders are not *subtypes* of infectious disorders. The AFTER attribute is used for linking post-infectious disorders with their associated infections.

6.2.3.2.3.12 Congenital, hereditary, familial, developmental, genetic

6.2.3.2.3.12.1 Congenital

The *attribute-value pair* | occurrence |=| congenital | is applied to those disorders that are present at birth. Although the word congenital is often applied to genetic disorders, the term *genetic* is preferred for those disorders that arise from abnormalities of the genes.

The preferred modeling pattern for congenital disorders requires consideration of the possible use of | associated morphology | with values that are congenital morphologic abnormalities, and the | occurrence | attribute with a value of "congenital". The accepted pattern is to use | occurrence |=| congenital |, and to discontinue the use of congenital morphologic abnormality concepts as values of | associated morphology |, replacing them with their non-congenital morphologic supertype. Once the congenital morphologies are no longer in use, they will be retired.

| Occurrence | should be in the same role group as | associated morphology | and | finding site |, because the morphology is located at the site, and the occurrence applies to the combined morphology / site pair.

For general congenital anomaly disorder grouper concepts such as | Congenital anomaly of cardiovascular system (disorder) |, the preferred value for | associated morphology | is | Developmental anomaly (morphologic abnormality) |.

When modeling congenital disorders, the following guidelines should be followed:

1. Disorders with the word "congenital" in their FSN should be classified under | Congenital disease (disorder) |.
2. Do not make a direct stated assertion that the parent is | Congenital disease (disorder) |. Instead allow the classifier to infer this relationship.
3. All concepts to be classified under | Congenital disease (disorder) | should have a stated relationship | occurrence |=| congenital |.
4. | Associated morphology | relationships will **not** be given values from under | Congenital anomaly (morphologic abnormality) |.

6.2.3.2.3.12.2 Congenital vs. acquired

The general rule is that disorders in general may be either congenital or acquired, and congenital disorders are specifically modeled using OCCURRENCE = congenital, but there is no attribute for modeling the fact that a disorder is acquired. If the FSN does not mention either congenital or acquired, then we do not model the *concept* as being under congenital disorder, and there is nothing in the *concept model* to specifically indicate that it is necessarily acquired. There are a few *concepts* that have acquired in their FSN, but they remain *primitive*.

6.2.3.2.3.12.2.1 Congenital vs acquired syphilis

As an exception to the general rule that acquired is never assumed, syphilis is a disorder in which an FSN that does not mention congenital might be assumed to imply acquired. ICD seems to follow such a rule, but there is no rule to that effect in *SNOMED*. In the absence of such a rule, the precise meaning of the FSN should be followed. If acquired is not stated in the FSN, the *concept* means the general category that subsumes both congenital and acquired forms.

6.2.3.2.3.12.3 Hereditary

It is difficult to cleanly define *hereditary* because it either may or may not include random mutations; the offspring of genetically normal parents may have a genetic disease, but there may be confusion about whether this is to be classified as a hereditary disease. It may be hereditary to the proband's offspring, but was not inherited from the proband's parents. Because of this ambiguity, *hereditary* requires case-by-case definition and is not a globally reproducible label for categories. Nevertheless, the names by which many diseases are known include the term, and is permitted as long as the usage does not introduce ambiguity.

6.2.3.2.3.12.4 Familial

The term *familial* is also somewhat ambiguous when used for broad categories. It may be interpreted as meaning that the disorder is found in higher proportions in the immediate or extended family than in other groups, or it may be intended as an indicator of possible heritable disease. In any case, it should not be used as a synonym for *genetic*.

6.2.3.2.3.12.5 Developmental

Developmental is a useful label for disorders that occur during development (both before and after birth) and that affect structures or functions that are in the process of developing. Some of these may be present at birth, and others may only manifest themselves post-natally.

6.2.3.2.3.12.6 Genetic, developmental, congenital and physical

The following diagram lays out the general logical structure of genetic, developmental and congenital categories, along with non-genetic, non-developmental and post-natal categories. A final dimension, called extrinsic physical force, is necessary to distinguish deformations from malformations. The various boxes in the diagram represent categories formed from the combination of the four dimensions each of which represents the answer to one of the following four questions:

1. is it genetic or not?
2. is it developmental or not?
3. is it present at birth or not?
4. is it due to an extrinsic physical force or not?

The diagonal hashed lines represent combination categories that do not occur. For example, there are no genetic disorders that are due to an extrinsic physical force. Likewise, there are no congenital disorders that are considered non-developmental. The blue lines represent congenital malformations; they may be either genetic or non-genetic in origin. The red circle represents those that are genetic in origin. Finally the solid colored area represents the meaning of acquired, i.e. any disorder that is non-genetic and not present at birth.

Arrows leading from each of the non-hashed boxes in the central diagram point to examples of disorders that typify that category. For example, Huntingtons disease is the typical example of a genetic disease that is neither congenital nor developmental. Although the gene defect is present at birth, the disease effects do not become manifest until adult life. Vitamin D deficiency rickets is a typical example of a non-genetic, non-congenital developmental malformation.

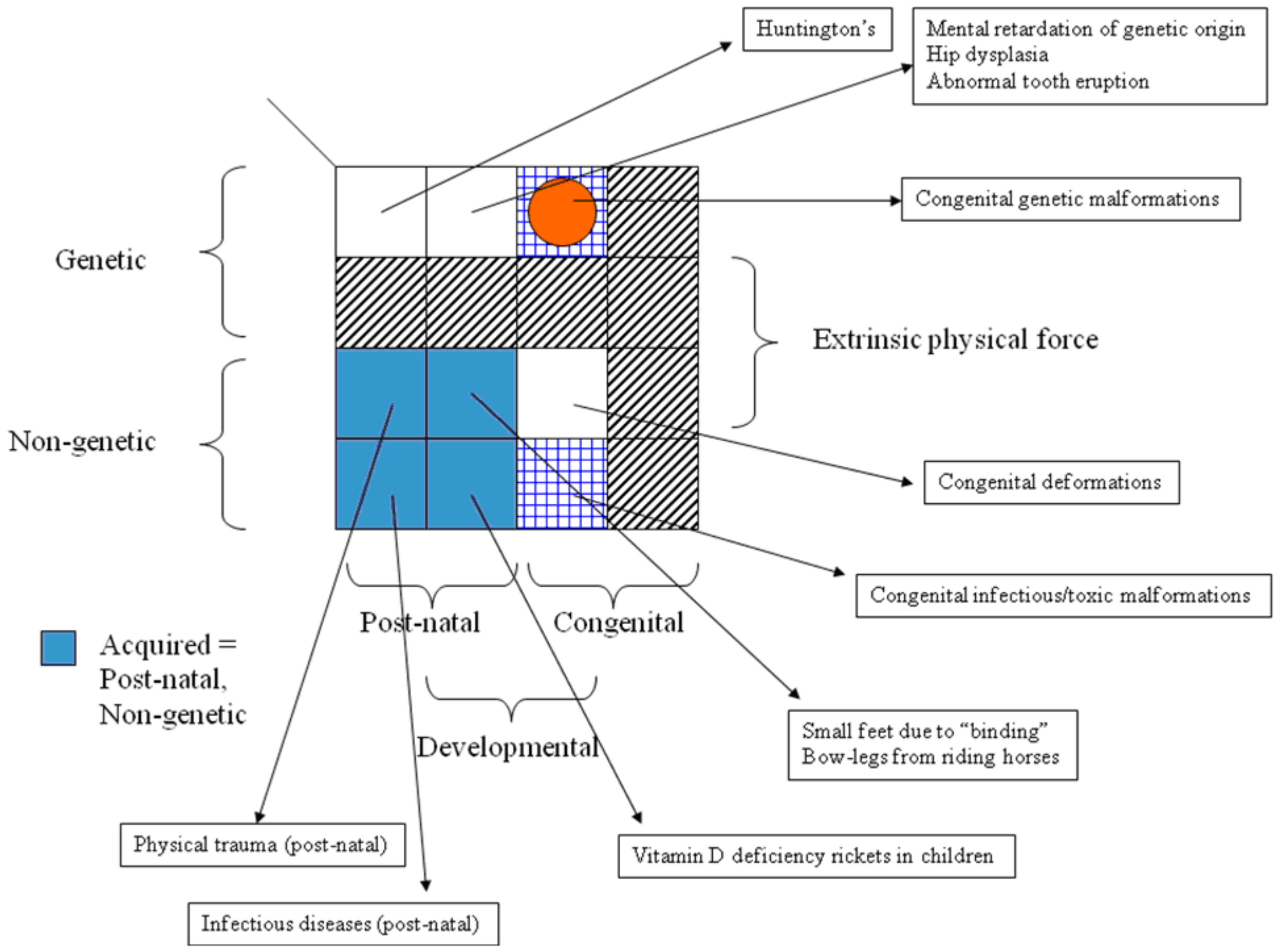


Figure 60: Diagram of the relationships of genetic, congenital, developmental, and acquired

6.2.3.2.3.13 Malformation, deformation, anomaly

As illustrated in figure 1 and described in section 4.1, a deformation is a structural abnormality that is due to an extrinsic physical force. Malformations are those structural abnormalities that result from intrinsically disordered structural development. The word anomaly is, by itself, ambiguous because it may be used to mean any abnormality including non-structural ones, or it may be used to mean malformation, or it may be a general term that includes both malformation and deformation. Terms that contain the word anomaly must therefore be examined to see whether the additional words provide sufficient specificity to overcome the inherent ambiguity. Congenital anomaly of <x structure> is definitely structural but is not the same as congenital malformation, and therefore it can be regarded as having the more general meaning of structural abnormality present at birth.

Hematologic

There is more than one meaning of "hematologic". A structural definition based on "hematological system structure" would include hematopoietic and lymphoid structures (including bone marrow, spleen, thymus, lymph nodes, etc) as well as the cellular components of blood. Hematologic neoplasms clearly fit this definition.

A definition based on what hematologists do is broader. Disorders of hemostasis and thrombosis are managed by hematologists, but these do not have a common structural overlap with the lymphoid and hematopoietic systems (with the exception of platelets and megakaryocytes). For clarity, "hematologic disorder" is a navigational *concept* that could be used to define a *reference set* that would include disorders of blood and blood forming organs, as well as disorders of hemostasis and thrombosis, depending on what is intended.

If a patient says they have a hematologic disorder or a blood disorder, the navigational *concept* could be used to record and capture what they said, but the variability in meaning is too great to assign necessary and sufficient conditions to this phrase.

6.2.3.2.3.14.1 Hematologic disorders, lymphoid and myeloid neoplasms

When a clinician or patient (or the literature) says "hematologic disorder," they could be referring to disorders with a morphology of hematopoietic cell origin, disorders affecting the blood forming organs (bone marrow, lymph nodes, spleen, thymus, and other lymph tissues), disorders of the cellular components of blood, and/or disorders of the function of hemostatic and thrombotic systems.

Within *SNOMED CT*, diseases of the cellular components of blood are most readily defined in terms of their definitional manifestations (for example, | anemia (disorder) | | HAS DEFINITIONAL MANIFESTATION | | Erythropenia (finding) |), because there is no clearly defined body site (a cell type is not considered a body site) and there may be no defined morphology.

Diseases of the blood forming organs (bone marrow, lymph nodes, etc.) can be defined in terms of any one or a combination of the following:

1. The morphology. For neoplastic diseases this is understood, at a minimum, to include those morphologies covered by the neoplasms listed in ICD-O.

   Example*:*

   | Hodgkin's disease (disorder) | | ASSOCIATED MORPHOLOGY | | Hodgkin lymphoma, no ICD-O subtype (morphologic abnormality) |
2. The body site involved especially specific lymph node groups or skin sites.

   Example:

   | primary cutaneous T-cell lymphoma (disorder) | | FINDING SITE | | skin structure |
3. The definitional manifestations of the disease — for those diseases without a specific neoplastic morphology and/or without a specific topographical site of involvement other than the blood-forming organs in general.

   Example:

   | toxic neutropenia (disorder) | | HAS DEFINITIONAL MANIFESTATION | | neutropenia (finding) |

Important examples of where it is important to distinguish disorders defined by morphology versus site versus manifestation include the T-cell lymphomas and disorders of plasma cells/immunosecretory disorders. T-cell lymphomas can be subcategorized according to the site of the primary: a lymph node versus the skin or other extranodal site. This means that a site of “lymphoid structure” *cannot* therefore be the *defining characteristic* of the parent *concept* “T-cell lymphoma”. Its defining attribute should be morphology alone. In the case of plasma cell disorders/immunosecretory disorders, some of them (monoclonal gammopathy, heavy chain disease, Waldenstrom's, etc) are defined in terms of their manifestation, i.e. the type of monoclonal protein they secrete, while others (myeloma, plasmacytoma) are defined in terms of their morphology, regardless of whether they are secretory or not. It would be incorrect to add "HAS DEFINITIONAL MANIFESTATION = monoclonal paraproteinemia" to myeloma, because not all myelomas are secretory. However, we can safely give immunosecretory disorders a morphology of "plasma cell neoplasm", even though no mass may have been identified and the monoclonal protein may be the only evidence that there is a clonal neoplasm.

In general, lymphoid and myeloid neoplasms *can* be modeled with their morphology alone, without a site. Leukemias and myelodysplastic syndromes are,*in addition*, modeled with site of bone marrow structure. Hairy cell leukemia has site bone marrow and site spleen, because both are uniformly involved.[11]

[11] For reference see *Harris NL et al, "World Health Organization Classification of Neoplastic Diseases of the Hematopoietic and Lymphoid Tissues: Report of the Clinical Advisory Committee Meeting-Airlie House, Virginia, November 1997" in J. Clinical Oncology, Vol.17, No 12 (December), 1999: pp 3835-3849.* For additional information on the classification of the myeloid neoplasms, see *Vardiman JW, Harris NL and Brunning RD. " World Health Organization (WHO) Classification of the Myeloid Neoplasms", Blood, 1 October 2002, Volume 100, Number 7, pp 2292-2302*.

6.2.3.2.3.14.2 Coagulation, Hemostasis and Thrombosis

There is more than one meaning of “coagulation”. A broad sense of “coagulation” as the stopping of bleeding is better described as hemostasis. A more narrow definition limited to the formation of the fibrin clot might exclude certain components of hemostasis, such as the ability to stop hemorrhage through the actions of blood vessels, collagen, endothelial cells, and platelets, in the absence of clotting. Individuals with congenital fibrinogen deficiency cannot form fibrin clots, yet they are able to stop bleeding. Therefore, coagulation disorders are kinds of hemostatic disorders.

6.2.3.2.3.15 Hernia, herniated structure, and hernial opening

Hernias involve two different structures, the structure herniated and the structure through which the hernia passes. Each of these might need to be described by different morphologies. There are two morphology codes, 414403008: | herniated structure (morphologic abnormality) |, and 414402003: | Hernial opening |. The herniated structure morphology should be the value of | ASSOCIATED MORPHOLOGY |, and this should be grouped with a | FINDING SITE | that has as its value the code for the anatomical structure that herniates.

Example:

Intestinal hernia (disorder)

- Group 1
  - | ASSOCIATED MORPHOLOGY | = herniated structure
  - | FINDING SITE | = intestinal structure
- Group 2
  - | ASSOCIATED MORPHOLOGY | = hernial opening
  - | FINDING SITE | = abdominal structure

The hernial opening morphology should be grouped with a | Finding site | attribute, with a value which is the code for the anatomical structure through which the hernia passes. It can be a general *concept* such as abdominal structure if the *concept* is non-specific about what is being herniated through. One or the other of these role groups should be omitted if the hernia does not necessarily entail a particular herniated structure or a particular hernial opening. For example, abdominal wall hernia specifies the hernial opening but not the herniated structure. In this case, the definition should omit the group with a morphology of herniated structure.

6.2.3.2.3.16 Osteoarthritis

Osteoarthritis is classically regarded as a degenerative disease, despite the "-itis" in its name. Because of this, | osteoarthritis | is not a *subtype* of | arthritis | in the disorder hierarchy. All cases of arthritis must be inflammatory by definition, but osteoarthritis has a subclass in the medical literature called non-inflammatory osteoarthritis. In fact, according to many authoritative sources, osteoarthritis is usually regarded as a non-inflammatory disease, and therefore it is not strictly a *subtype* of arthritis.

Structuring the hierarchy this way does not imply that there are no cases of osteoarthritis with inflammation, nor does it rule out inflammation as an etiologic or contributory factor in many cases. It is well established that inflammation occurs in many cases of osteoarthritis, and treatment with anti-inflammatory agents has been shown to be more effective than pure analgesics in many cases. The key point is that despite growing evidence of a role of inflammatory cytokines in many cases, osteoarthritis is not always necessarily an inflammatory disorder of the joint.

6.2.3.2.4 Changes and historical notes

6.2.3.2.4.1 EPISODICITY no longer modeled in active content

| EPISODICITY | originated in the *National Health Service Clinical Terms Version 3* where it was used not to specify the first episode of a disease for a patient but rather, the first time a patient presented to their general practitioner (GP) for a particular disorder. A first episode of asthma was not intended to represent the first time a patient had asthma, but rather the first time a patient presented to their GP with

asthma. | EPISODICITY | has been removed from existing *concepts* and is no longer used in *precoordinated* definitions. It can still be used in *postcoordination* as a *qualifier*.

6.2.3.2.4.2 ONSET and COURSE retired

In earlier releases, there were two attributes named | ONSET | and | COURSE |. These were *retired* because they could not be used reproducibly. While | ONSET | was intended to specify the rapidity of onset or the temporal pattern of presentation for a given condition, it was easily confused with the attribute | COURSE | used to represent the duration of a condition. There was not consistent agreement between observers making this distinction.

6.2.3.2.4.3 Dose form values moved

The *concept* 105904009 | Type of drug preparation (product) | and its *subtypes* were moved to the *Qualifier* value *hierarchy* as of the July 2007 release. 105904009 | Type of drug preparation (qualifier value) | better represents these *concepts* because they are not products.

6.2.3.2.4.4 Renaming the context/situation hierarchy

The *hierarchy* named 243796009 | situation with explicit context (situation) | was called | context-dependent category | until the July 2006 release. The *hierarchy* was renamed to better describe the meanings in this *hierarchy*.

6.2.3.2.4.5 Domain change for measurement/evaluation attributes

In releases prior to July 2009, six *attributes* were approved for use for | measurement procedure | only. For the July 2009 release, the *domain* for these *attributes* was expanded to | evaluation procedure |. See *Measurement procedures and laboratory procedures* on page 311 for a definition and full discussion of | evaluation procedure | and | measurement procedure |.

6.2.3.2.4.6 Move of findings to events

In January 2006, a number of *concepts* from the | Clinical finding | *hierarchy* were moved to the *Event hierarchy*. The attributes used to define those *concepts* when they were *descendants* of | Clinical finding | were retained after the *concepts* were moved to the *Event hierarchy*. Additional editorial policies for the use of attributes in the *Event hierarchy* have yet to be established.

6.2.3.3 Procedures

The top level of the procedure hierarchy has the following structure:

SNOMED CT concept

- procedure
  - administrative procedure
  - community health procedure
  - environmental care procedure
  - general treatment
  - laboratory procedure
  - obstetric procedure
  - outpatient procedure
  - patient encounter procedure
  - preoperative/postoperative procedures
  - procedure by device
  - procedure by intent
  - procedure by method
  - procedure by priority
  - procedure by site
  - procedure in coronary care unit
  - procedure related to anesthesia and sedation

- procedure with a clinical finding focus
- procedure with a procedure focus
- provider-specific procedure
- regimes and therapies
- social service procedure
- specimen collection
- staff related procedure

6.2.3.3.1 Procedure

I Procedure I *concepts* represent activities performed in the provision of health care. This *hierarchy* represents a broad variety of activities, including but not limited to, invasive procedures (e.g. I Excision of intracranial artery (procedure) I), administration of medicines (e.g. I Pertussis vaccination (procedure) I), imaging procedures (e.g. I Ultrasonography of breast (procedure) I), education procedures (e.g. I Low salt diet education (procedure) I), and administrative procedures (e.g. I Medical records transfer (procedure) I).

Examples of Procedure concepts:

- I Removal of urethral catheter (procedure) I ;
- I Intravenous steroid injection (procedure) I ;
- I Irrigation of oral wound (procedure) I ;
- I Appendectomy (procedure) I .

Note: See also *Attributes used to define Procedure concepts*.

6.2.3.3.2 Attributes used to define Procedure concepts

Table 125: Approved Procedure attributes summary

Defining Attribute	Subsumed Attribute	Allowable Values
I PROCEDURE SITE I		I Anatomical or acquired body structure I 442083009 (<<)
	I Procedure site - Direct I	I Anatomical or acquired body structure I 442083009 (<<)
	I Procedure site - Indirect I	I Anatomical or acquired body structure I 442083009 (<<)
I PROCEDURE MORPHOLOGY I		I Morphologically abnormal structure I 49755003 (<<)
	I Direct morphology I	I Morphologically abnormal structure I 49755003 (<<)
	I Indirect morphology I	I Morphologically abnormal structure I 49755003 (<<)
I METHOD I		I Action I 129264002 (<<)

<table>
<tr><th>Defining Attribute</th><th>Subsumed Attribute</th><th>Allowable Values</th></tr>
<tr><td rowspan="5">| PROCEDURE DEVICE |</td><td></td><td>| Device | 49062001 (<<)</td></tr>
<tr><td>| DIRECT DEVICE |</td><td>| Device | 49062001 (<<)</td></tr>
<tr><td>| INDIRECT DEVICE |</td><td>| Device | 49062001 (<<)</td></tr>
<tr><td>| USING DEVICE |</td><td>| Device | 49062001 (<<)</td></tr>
<tr><td>| USING ACCESS DEVICE |</td><td>| Device | 49062001 (<<)</td></tr>
<tr><td colspan="2">| ACCESS |</td><td>| Surgical access values | 309795001 (<=)(< Q)</td></tr>
<tr><td colspan="2">| DIRECT SUBSTANCE |</td><td>| Substance | 105590001 (<<)
| Pharmaceutical / biologic product | 373873005 (<<)</td></tr>
<tr><td colspan="2">| PRIORITY |</td><td>| Priorities | 272125009 (<=)(< Q)</td></tr>
<tr><td colspan="2">| HAS FOCUS |</td><td>| Clinical finding | 404684003 (<<)
| Procedure | 71388002 (<<)</td></tr>
<tr><td colspan="2">| HAS INTENT |</td><td>| Intents (nature of procedure values) | 363675004 (<=)</td></tr>
<tr><td colspan="2">| RECIPIENT CATEGORY |</td><td>| Person | 125676002 (<<)
| Family | 35359004 (<<)
| Community | 133928008 (<<)
| Donor for medical or surgical procedure | 105455006 (<<)
| Group | 389109008 (<<)</td></tr>
<tr><td colspan="2">| REVISION STATUS |</td><td>| Primary operation | 261424001 (<<)
| Revision - value | 255231005 (<<)
| Part of multistage procedure | 257958009 (<<)</td></tr>
<tr><td colspan="2">| ROUTE OF ADMINISTRATION |</td><td>| Route of administration value | 284009009 (<<)</td></tr>
<tr><td colspan="2">| SURGICAL APPROACH |</td><td>| Procedural approach | 103379005 (<=)(< Q)</td></tr>
<tr><td colspan="2">| USING ENERGY |</td><td>| Physical force | 78621006 (<<)</td></tr>
<tr><td colspan="2">| USING SUBSTANCE |</td><td>| Substance | 105590001 (<<)</td></tr>
</table>

Note:

Meaning of Allowable Values (*Range*) notations:

(<<) this code and *descendants*,
(<) *descendants* only,
(<=) *descendants* only (stated) except for supercategory groupers,
(==) this code only,
(< Q) *descendants* only when in a qualifying *Relationship*,
(< Q only) *descendants* only, and only allowed in a qualifying *Relationship*.

Note:

Attributes should be grouped with the | METHOD | attribute to which they apply; in the absence of a | METHOD | attribute, attributes that are related to each other should be grouped. The one exception is | RECIPIENT CATEGORY |, because a single procedure code should not be *precoordinated* in situations where more than one recipient category is involved. Such complex statements should utilize two or more procedure codes that are placed into an appropriately structured information model.

Note: See also *Procedure*.

6.2.3.3.2.1 PROCEDURE SITE

The | PROCEDURE SITE | attribute describes the body site acted on or affected by a procedure. This attribute subsumes, in an attribute *hierarchy* (see *Attribute Hierarchies in SNOMED CT* on page 272), the more specific attributes (| Procedure site - Direct | and | Procedure site - Indirect |) that should be used if possible. The anatomical site may be directly acted on (| Procedure site - Direct |) or indirectly acted upon (| Procedure site - Indirect |).

When modeling procedures where the | METHOD | is | Removal - action | or one of its *subtypes* (e.g. | Excision |, | Surgical biopsy |, etc.), removals **of** the structure itself should use | Procedure site - Direct |. Removals of tissue lesions (cysts, tumors, etc.) are considered to be removals of the site, and should also use | Procedure site - Direct |. Removals of devices, calculi, thrombi, foreign bodies and other non-tissue entities **from** the structure should use | Procedure site - Indirect |.

Table 126: Permissible values for PROCEDURE SITE

Attribute Values	**Examples**
| Anatomical or acquired body structure | 442083009 (<<)	| Procedure on colon (procedure) | • | PROCEDURE SITE | | colon structure (body structure) |

Procedures need not necessarily be categorized by site. | Human body structure | should *not* be assigned as a default value of this attribute because many procedures can be performed on non-human subjects, and because this attribute does *not* necessarily need to be present in a procedure *concept* definition in order for *classifier* algorithms to work properly.

The general | PROCEDURE SITE | attribute is used to model the site for high-level grouper type procedure *concepts*. It is most likely to be used for *concepts* that do not require a | METHOD | (action) attribute. Relatively few *concepts* will be modeled using | PROCEDURE SITE |, rather than the more specific direct and indirect site attributes (see below).

6.2.3.3.2.1.1 PROCEDURE SITE DIRECT

This attribute is used when the action of the procedure is directly aimed at an anatomical or acquired body structure or site rather than at something else (such as a device) located there.

Table 127: Permissible values for PROCEDURE SITE DIRECT

Attribute Values	**Examples**
I Anatomical or acquired body structure I 442083009 (<<)	I Amputation of the foot (procedure) I • I METHOD I I Amputation - action (qualifier value) I • I Procedure site - Direct I I Foot structure (body structure) I
	I Biopsy of femur (procedure) I • I METHOD I I Biopsy - action (qualifier value) I • I Procedure site - Direct I I Bone structure of femur (body structure) I

6.2.3.3.2.1.1.1 Multiple values for PROCEDURE SITE DIRECT

When the I METHOD I (action) acts directly on a morphological abnormality (more simply, a lesion) arising from, or existing in, the cells of the tissue in which it occurs [e.g. a tumor (including metastatic tumors), granuloma, polyp, or cyst] the attribute I DIRECT MORPHOLOGY I is used to model the morphological abnormality. Most *concept* definitions where I DIRECT MORPHOLOGY I is used, which also require a site in the definition, will use I Procedure site - Direct I. Thus, there can be more than one direct object of the I METHOD I for a *concept*. For example, the I DIRECT MORPHOLOGY I and the I Procedure site - Direct I can both be direct objects of the I METHOD I. An example of an exception to this rule would be removal of a calculus from the ureter. In this case, the calculus is the direct object, but there is no procedure site that is that direct object, since the ureter is an indirect object.

The most common *concepts* that have more than one direct object of the I METHOD I are *Subtypes* of I Removal (procedure) Iwhere the object of the removal (e.g. a neoplasm) can be considered to be a part of the tissue at the anatomical site in which it occurs. When a part of an anatomical structure (however abnormal) has been removed, both the morphological abnormality and the anatomical structure in which it is located are to be modeled as direct objects for the I METHOD I I Removal - action (qualifier value) I. Grafts that become attached via in-growth of capillaries, fibroblasts, and/or other cells or tissues would also be regarded as biologically connected, and therefore modeling their removal would include the anatomical structure as a direct object of the action. The anatomical structure is not to be modeled as a direct object of a removal only when the procedure does not necessarily involve removal also of part of the anatomy; examples include removals of things such as a foreign body, a catheter, a renal calculus, or a mechanical implant like a pacemaker.

6.2.3.3.2.1.2 PROCEDURE SITE INDIRECT

This attribute describes the anatomical site, which is acted upon, but is not the direct object of the procedure. (The site is indirectly acted on by the procedure.) Usually in these procedures there is another value that is the direct object of the action. Exceptions (*concepts* that do not specify a direct object, but only an indirect object) are usually general groupers such as I Arm implantation (procedure) I (meaning implantation of something into the arm), since the thing implanted could be either a device or a substance (material).

Table 128: Permissible values for PROCEDURE SITE INDIRECT

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td rowspan="2">| Anatomical or acquired body structure | 442083009 (<<)</td><td>| Removal of catheter from brachial vein (procedure) | | METHOD | | Removal - action (qualifier value) | | DIRECT DEVICE | | Catheter, device (physical object) | | Procedure site - Indirect | | Structure of brachial vein (body structure) |</td></tr>
<tr><td>| Removal of calculus of urinary bladder (procedure) | | METHOD | | Removal - action (qualifier value) | | DIRECT MORPHOLOGY | | Calculus (morphologic abnormality) | | Procedure site - Indirect | | Urinary bladder structure (body structure) |</td></tr>
</table>

6.2.3.3.2.2 PROCEDURE MORPHOLOGY

| PROCEDURE MORPHOLOGY | is the attribute used to specify the morphology or abnormal structure involved in a procedure. This attribute subsumes the more specific attributes | DIRECT MORPHOLOGY | and | INDIRECT MORPHOLOGY | that should be used if possible (see below). | DIRECT MORPHOLOGY | is used when the procedure method acts directly on the morphologic abnormality. | INDIRECT MORPHOLOGY | is used when the procedure method acts directly on something else (e.g. a device, substance or anatomical structure) that is associated with the morphologic abnormality. The more general attribute | PROCEDURE MORPHOLOGY | is used when defining general *concepts* that subsume both kinds of sub - *concepts*.

Table 129: Permissible values for PROCEDURE MORPHOLOGY

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td>| Morphologically abnormal structure | 49755003 (<<)</td><td></td></tr>
</table>

Hematoma , calculus, foreign body, blood clot, embolus, and some other entities are not strictly body structures, but are in the body structure *hierarchy* under morphologically abnormal structure, and are valid values for the | PROCEDURE MORPHOLOGY | attributes.

6.2.3.3.2.2.1 DIRECT MORPHOLOGY

This attribute describes the morphologically abnormal structure that is the direct object of the METHOD action.

Table 130: Permissible values for DIRECT MORPHOLOGY

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td>| Morphologically abnormal structure | 49755003 (<<)</td><td>| Excision of benign neoplasm (procedure) | | METHOD | | Excision - action (qualifier value) | | DIRECT MORPHOLOGY | | Neoplasm, benign (morphologic abnormality) |</td></tr>
</table>

6.2.3.3.2.2.2 INDIRECT MORPHOLOGY

This attribute represents a morphology that is acted upon, but is not the direct target of the action being performed (i.e. the procedure's method acts directly on something else, such as a device, substance, or anatomical structure).

Table 131: Permissible values for INDIRECT MORPHOLOGY

Attribute Values	Examples
| Morphologically abnormal structure | 49755003 (<<)	| Removal of mesh from wound (procedure) | • | METHOD | | Removal - action (qualifier value) | • | DIRECT DEVICE | | Mesh (physical object) | • | INDIRECT MORPHOLOGY | | Wound (morphologic abnormality) |

6.2.3.3.2.3 METHOD

This attribute represents the action being performed to accomplish the procedure. It does not include the surgical approach (e.g. translumbar), equipment (e.g. sutures), or physical forces (e.g. laser energy).

Table 132: Permissible values for METHOD

Attribute Values	Examples
| Action | 129264002 (<<)	| Incision of ureter (procedure) | • | METHOD | | Incision - action (qualifier value) | • | Procedure site - Direct | | Ureteric structure (body structure) |

The | METHOD | can be considered the anchor of each *relationship group* that defines a procedure; if there are two methods, there should be two different *relationship groups*. It is correct to regard each *relationship* group as a kind of sub-procedure that defines the overall procedure. Each method can be regarded as the verb of a sentence, and the verbs direct and indirect objects are specified by the site, morphology, device, substance or energy attributes (below) that are grouped with it.

6.2.3.3.2.4 PROCEDURE DEVICE

| PROCEDURE DEVICE | is a general attribute used to model devices associated with a procedure. It subsumes the more specific attributes | DIRECT DEVICE |, | INDIRECT DEVICE |, | USING DEVICE |, and | USING ACCESS DEVICE |, which should be used instead of | PROCEDURE DEVICE | if possible. The general attribute | PROCEDURE DEVICE | is mainly useful for defining high-level, general *concepts* that aggregate procedures according to the device involved.

Table 133: Permissible values for PROCEDURE DEVICE

Attribute Values	Examples
| Device | 49062001 (<<)	| Catheter procedure (procedure) | • | PROCEDURE DEVICE | | Catheter, device (physical object) |

When the device is the direct object of the action (| METHOD |), the attribute | DIRECT DEVICE | is used. If the action is done indirectly to the device, that is, the action is done to something that is located in or on a device, but is not done directly to the device itself, then the attribute | INDIRECT DEVICE | is used. If the device is used to carry out the action, then the attribute | USING DEVICE | is used. If the device is used to access the site of the action, then the attribute | USING ACCESS DEVICE | is used.

Note: The permissible values for attributes in the | PROCEDURE DEVICE | role *hierarchy* include | Device (physical object) | and its *descendants*. However, there are a limited number of products in *SNOMED CT* which are devices that also deliver drugs. These *concepts* descend from | Drug-device combination product (product) | which is a *descendant* of both | Device (physical object) | and | Pharmaceutical / biologic product (product) |. Therefore, although they carry the *hierarchy* tag of (product), they are valid values for attributes in the | PROCEDURE DEVICE | role *hierarchy*.

Example:

| Removal of drug coated stent (procedure) |

- | METHOD | | Removal - action (qualifier value) |
- | DIRECT DEVICE | | Drug coated stent (product) |

6.2.3.3.2.4.1 DIRECT DEVICE

This attribute represents the device on which the method directly acts.

Table 134: Permissible values for DIRECT DEVICE

Attribute Values	**Examples**
| Device | 49062001 (<<)	| Removal of arterial stent (procedure) | • | METHOD | | Removal - action (qualifier value) | • | DIRECT DEVICE | | Arterial stent (physical object) |

6.2.3.3.2.4.2 INDIRECT DEVICE

This attribute models action done on something that is located in or on a device, but is not done directly on the device itself.

Table 135: Permissible values for INDIRECT DEVICE

Attribute Values	**Examples**
| Device | 49062001 (<<)	| Excision of vegetations from implanted mitral valve (procedure) | • | METHOD | | Excision - action (qualifier value) | • | DIRECT MORPHOLOGY | | Vegetation (morphologic abnormality) | • | INDIRECT DEVICE | | Mitral valve prosthesis, device (physical object) | • | Procedure site - Indirect | | Mitral valve structure (body structure) |

Note:

In the above example, the vegetation is being excised. The mitral valve prosthesis is where the excised vegetation is located but the mitral valve prosthesis itself is not excised. Thus, mitral valve prosthesis is the | INDIRECT DEVICE |.

Note:

The attribute | INDIRECT DEVICE | is infrequently needed. When using this attribute, a second look is advisable to be sure it is needed.

6.2.3.3.2.4.3 USING DEVICE

This attribute refers to the instrument or equipment utilized to execute an action. | USING DEVICE | is used when the device is actually used to carry out the action that is the focus of the procedure. If the device is simply the means to access the site of the procedure, then | USING ACCESS DEVICE | is used instead of | USING DEVICE |.

Table 136: Permissible values for USING DEVICE

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td>| Device | 49062001 (<<)</td><td>| Core needle biopsy of larynx (procedure) |
• | METHOD | | Biopsy - action (qualifier value) |
• | USING DEVICE | | Core biopsy needle, device (physical object) |
• | Procedure site - Direct | | Laryngeal structure (body structure) |</td></tr>
</table>

6.2.3.3.2.4.4 USING ACCESS DEVICE

This attribute specifies the instrument or equipment used to access the site of a procedure.

Table 137: Permissible values for USING ACCESS DEVICE

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td>| Device | 49062001 (<<)</td><td>| Arthroscopic synovial biopsy (procedure) |
• | METHOD | | Biopsy - action (qualifier value) |
• | USING ACCESS DEVICE | | Arthroscope, device (physical object) |
• | Procedure site - Direct | | Structure of synovial tissue of joint (body structure) |</td></tr>
</table>

6.2.3.3.2.5 ACCESS

This attribute describes the route used to access the site of a procedure. It is used to distinguish open, closed, and percutaneous procedures.

Table 138: Permissible values for ACCESS

Attribute Values	**Examples**
| Surgical access values | 309795001 (<=)(< Q)	| Open removal of bile duct stent (procedure) | • | ACCESS | | Open approach - access (qualifier value) |

6.2.3.3.2.6 DIRECT SUBSTANCE

This attribute describes the | Substance | or | Pharmaceutical / biologic product | on which the procedure's method directly acts.

Table 139: Permissible values for DIRECT SUBSTANCE

Attribute Values	**Examples**
| Substance | 105590001 (<<) | Pharmaceutical / biologic product | 373873005 (<<)	| Injection of prostaglandin (procedure) | • | METHOD | | Injection - action (qualifier value) | • | DIRECT SUBSTANCE | *| Prostaglandin (substance) |*

Note: As an editorial policy, in the distribution form of the *International Release*, | Pharmaceutical / biologic product (product) | and its *descendants* are not used as values for | DIRECT SUBSTANCE |.

6.2.3.3.2.7 PRIORITY

This attribute refers to the priority assigned to a procedure.

Table 140: Permissible values for PRIORITY

Attribute Values	**Examples**
| Priorities | 272125009 (<=)(< Q)	| Emergency cesarean section (procedure) | • | PRIORITY | | Emergency (qualifier value) |

6.2.3.3.2.8 HAS FOCUS

This attribute specifies the | Clinical finding | or | Procedure | which is the focus of a procedure.

Table 141: Permissible values for HAS FOCUS

Attribute Values	**Examples**
| Clinical finding | 404684003 (<<) | Procedure | 71388002 (<<)	| Cardiac rehabilitation assessment (procedure) | • | HAS FOCUS | | Cardiac rehabilitation (regime/therapy) |

6.2.3.3.2.9 HAS INTENT

This attribute specifies the intent of a procedure.

Table 142: Permissible values for HAS INTENT

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td>| Intents (nature of procedure values) | 363675004 (<=)</td><td>| Diagnostic bronchoscopy (procedure) |
• | HAS INTENT | | Diagnostic intent (qualifier value) |</td></tr>
</table>

6.2.3.3.2.10 RECIPIENT CATEGORY

This attribute specifies the type of individual or group upon which the action of the procedure is performed. For example, it can be used in blood banking procedures to differentiate whether the procedure was performed on the donor or the recipient of a blood product. In other words, | RECIPIENT CATEGORY | is | Donor for medical or surgical procedure (person) | if the subject of the record is the donor.

It is not used for a procedure where the subject of the procedure is someone other than the subject of record.

Table 143: Permissible values for RECIPIENT CATEGORY

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td>| Person | 125676002 (<<)
| Family | 35359004 (<<)
| Community | 133928008 (<<)
| Donor for medical or surgical procedure | 105455006 (<<)
| Group | 389109008 (<<)</td><td>| Social service interview of family (procedure) |
• | RECIPIENT CATEGORY | | Family (social concept) |</td></tr>
</table>

6.2.3.3.2.11 REVISION STATUS

This attribute specifies whether a procedure is primary or a revision.

Table 144: Permissible values for REVISION STATUS

<table>
<tr><th>Attribute Values</th><th>Examples</th></tr>
<tr><td rowspan="2">| Primary operation | 261424001 (<<)
| Revision - value | 255231005 (<<)
| Part of multistage procedure | 257958009 (<<)</td><td>| Primary repair of inguinal hernia (procedure) |
• | REVISION STATUS | | Primary operation (qualifier value) |</td></tr>
<tr><td>| Revision of knee arthroplasty (procedure) |
• | REVISION STATUS | | Revision - value (qualifier value) |</td></tr>
</table>

6.2.3.3.2.12 ROUTE OF ADMINISTRATION

This attribute allows representation of the route by which a procedure introduces a given substance into the body.

The domain for this attribute is the *sub-hierarchy* below | Administration of substance via specific route (procedure) | 433590000.

Table 145: Permissible values for ROUTE OF ADMINISTRATION

Attribute Values	**Examples**
I Route of administration value I 284009009 (<<)	I Intravitreal steroid injection (procedure) I • I ROUTE OF ADMINISTRATION I I Intravitreal route (qualifier value) I

6.2.3.3.2.13 SURGICAL APPROACH

This attribute specifies the directional, relational, or spatial access to the site of a surgical procedure. The domain for I SURGICAL APPROACH I is *descendants* of I Surgical procedure (procedure) I 387713003.

Table 146: Permissible values for SURGICAL APPROACH

Attribute Values	**Examples**
I Procedural approach I 103379005 (<=)(< Q)	I Intranasal ethmoidectomy (procedure) I • I SURGICAL APPROACH I I Intranasal approach (qualifier value) I
	I Abdominal hysterectomy (procedure) I • I SURGICAL APPROACH I I Abdominal approach (qualifier value) I

6.2.3.3.2.14 USING SUBSTANCE

This attribute describes the I Substance I used to execute the action of a procedure, but it is not the substance on which the procedure's method directly acts (the I DIRECT SUBSTANCE I).

Table 147: Permissible values for USING SUBSTANCE

Attribute Values	**Examples**
I Substance I 105590001 (<<)	I Contrast radiography of esophagus (procedure) I • I METHOD I I Radiographic imaging - action (qualifier value) I • I Procedure site - Direct I I Esophageal structure (body structure) I • I USING SUBSTANCE I I Contrast media (substance) I

6.2.3.3.2.15 USING ENERGY

This attribute describes the energy used to execute an action. I USING ENERGY I has been introduced because the new attribute I USING DEVICE I is now used only to represent the instrument or equipment used to execute the action. Unlike the attribute USING, which it replaces, I USING DEVICE I does not take values from the I physical force I *hierarchy*.

Table 148: Permissible values for USING ENERGY

Attribute Values	**Examples**
| Physical force | 78621006 (<<)	| Gamma ray therapy (procedure) | • | USING ENERGY | | Gamma radiation (physical force) |

6.2.3.3.2.16 Direct and indirect objects

Procedures that have a | METHOD | *attribute* can be described using an action verb that corresponds to the method. The direct object(s) of the action verb should be represented using (at least) one of the four direct object *attributes*, depending on whether the direct object on which the method acts is a device (| DIRECT DEVICE |), anatomical structure (| Procedure site - Direct |), morphologic abnormality (| DIRECT MORPHOLOGY |) or substance (| DIRECT SUBSTANCE |).

When the type (body structure, device, or substance) of direct object is indeterminate, the direct-object *attributes* should not be used.

6.2.3.3.3 Attributes used to define Evaluation Procedure concepts

Table 149: Approved Evaluation Procedure attributes summary

Defining Attribute	**Allowable Values**
| HAS SPECIMEN |	| Specimen | 123038009 (<=)(< Q)
| COMPONENT |	| Substance | 105590001 (<=)(< Q) | Observable entity | 363787002 (<=)(< Q) | Cell structure | 4421005 (<=)(< Q) | Organism | 410607006 (<=)(< Q)
| TIME ASPECT |	| Time frame | 7389001 (<=)(< Q)
| PROPERTY |	| Property of measurement | 118598001 (<=)(< Q)
| SCALE TYPE |	| Quantitative | 30766002 (<<) | Qualitative | 26716007 (<<) | Ordinal value | 117363000 (<<) | Ordinal or quantitative value | 117365007 (<<) | Nominal value | 117362005 (<<) | Narrative value | 117364006 (<<) | Text value | 117444000 (<<)
| MEASUREMENT METHOD |	| Laboratory procedure categorized by method | 127789004(<=)

Note:

Meaning of Allowable Values (*Range*) notations:

> **(<<)** this code and *descendants*,
> **(<)** *descendants* only,
> **(<=)** *descendants* only (stated) except for supercategory groupers,
> **(==)** this code only,
> **(< Q)** *descendants* only when in a qualifying *Relationship*,
> **(< Q only)** *descendants* only, and only allowed in a qualifying *Relationship*.

Note: See also *Observable entity*.

6.2.3.3.3.1 HAS SPECIMEN

This attribute specifies the type of specimen on which a measurement or observation is performed.

Table 150: Permissible values for HAS SPECIMEN

Attribute Values	
I Specimen I 123038009 (<=)(< Q)	

6.2.3.3.3.2 COMPONENT

This attribute refers to what is being observed or measured by a procedure.

Table 151: Permissible values for COMPONENT

Attribute Values	**Example**
I Substance I 105590001 (<=)(< Q) I Observable entity I 363787002 (<=)(< Q) I Cell structure I 4421005 (<=)(< Q) I Organism I 410607006 (<=)(< Q)	I Protein measurement (procedure) I • I COMPONENT I I Protein (substance) I

6.2.3.3.3.3 TIME ASPECT

This attribute specifies temporal *relationships* for a measurement procedure.

Table 152: Permissible values for TIME ASPECT

Attribute Values	
I Time frame I 7389001 (<=)(< Q)	

6.2.3.3.3.4 PROPERTY

This attribute specifies the kind of property being measured (e.g. concentration).

Table 153: Permissible values for PROPERTY

Attribute Values	
	Property of measurement	118598001 (<=)(< Q)	

6.2.3.3.3.5 SCALE TYPE

This attribute refers to the scale of the result of an observation of a diagnostic test (i.e. quantitative, qualitative, semi-quantitative).

Table 154: Permissible values for SCALE TYPE

Attribute Values	
	Quantitative	30766002 (<<) | Qualitative	26716007 (<<) | Ordinal value	117363000 (<<) | Ordinal or quantitative value	117365007 (<<) | Nominal value	117362005 (<<) | Narrative value	117364006 (<<) | Text value	117444000 (<<)	

6.2.3.3.3.6 MEASUREMENT METHOD

This attribute specifies the method by which a procedure is performed.

Table 155: Permissible values for MEASUREMENT METHOD

Attribute Values	
	Laboratory procedure categorized by method	127789004(<=)	

For measurement procedures, the attribute | METHOD | is given the value | Measurement - action (qualifier value) |. The attribute | MEASUREMENT METHOD | can be used to provide additional specificity.

6.2.3.3.4 Attribute overlap or interaction

6.2.3.3.4.1 Method vs intent

Some methods intrinsically have intent stated in their name, such as "diagnostic surgical action".

6.2.3.3.5 Specific procedure types

6.2.3.3.5.1 Surgical procedure

A surgical procedure is defined as a procedure that involves intentional non-transient alteration of structures of the body, and/or a procedure that necessarily involves cutting into the body. From a practical standpoint, this definition is implemented to include all procedures defined by the METHOD attribute with a value of any action that is listed under surgical action.

In *SNOMED*, operation is synonymous with surgical procedure.

Medical procedure is a deprecated term, because of its clear lack of reproducible meaning. It might be considered to be defined as a procedure done by a physician, but even in this case, it would be deprecated on the basis that it is provider-specific. On the other hand, it seems quite clear (despite some dictionary definitions) that surgical procedures are *not* defined simply as procedures done by a surgeon; a surgeon can carry out many non-surgical actions (examining patients, prescribing, advising, etc). Even more important, a surgical procedure need not necessarily be performed by a surgeon; if a non-surgeon does a procedure that is surgical, it still remains a surgical procedure.

6.2.3.3.5.1.1 Operation vs. Procedure

While there may never be a complete consensus as to what constitutes a surgical procedure, the agreement has been to classify *concepts* as surgical procedures if their method is a surgical action based on the action hierarchy. In turn, the surgical action hierarchy distinguishes surgical from non-surgical actions based on the working definition above. Note the or in the sentence; actions that do not involve cutting or incision, but do involve the intentional non-transient alteration of anatomy, are still surgical.

Examples of non-surgical actions include fine-needle or brush biopsies, phlebotomy, aspiration, and closed reduction of dislocations - since they both do not significantly or non-transiently alter anatomy and do not necessarily involve cutting.

Examples of borderline actions that are currently classified as surgical include core needle biopsies - these are more invasive and result in more tissue removal than fine-needle biopsies - and centesis, on the theory that combining puncture with removal alters body structure.

Unresolved issue: fine-needle biopsy could be viewed as a kind of centesis, but the former is non-surgical and the latter is surgical. This appears inconsistent.

Sampling, in general, is not necessarily a surgical action. If what is intended is a sampling that involves surgical removal of part of something, then surgical biopsy action should be the action specified.

6.2.3.3.5.1.2 Surgical Repair

The definition of surgical repair is Restoring, to the extent possible, the natural anatomical structure, using a surgical action. When we clearly distinguish the means by which a procedure is accomplished from the need for the procedure (modeled using the FOCUS attribute), and distinguish these from the objective or intended accomplishment, we can see that repair is clearly an objective or intended accomplishment,*not* a means (which can be suturing, transplanting, etc) *nor* a need (normal functioning, cosmetic appearance, pain relief, etc). Although restoring natural structure will be intended to restore natural function and appearance, functional restoration is not necessary for a procedure to be a repair. On the other hand, it is possible to do an operation intended to restore function without restoring structure (such as with surgery to allow attachment of prosthetic limb replacements after amputation) and this type of surgery would not be strictly categorized as a repair.

Since the current model does not have different attributes to distinguish the *objective* of a procedure from the *means* used to accomplish it, we continue to use METHOD to model both types of information. If the definition of a procedure requires both a *repair action* and another action that is not a kind of repair, then two role groups should be used.

6.2.3.3.5.1.3 Plastic operation, -plasty, and plastic repair

The word plastic refers to reshaping, and operations that accomplish a repair (a structural restoration) often use the suffix -plasty. The term plastic repair also occurs, and in order for this not to be a redundancy, there must be a distinction between plastic and repair. The distinction can be found in the contrasting use of the terms plastic and prosthetic. A prosthetic repair uses external (non-body) materials to accomplish the structural restoration, while a plastic repair reshapes the body to accomplish the structural restoration. Plastic repairs therefore are distinguished from general repairs, and distinguished from prosthetic repairs.

The suffix -plasty is widely used in terms that apply to prosthetic repairs (such as in total hip arthroplasty), so we must interpret -plasty to mean any general repair (either prosthetic or plastic or other), and not necessarily just plastic repairs (which reshape existing tissues).

6.2.3.3.5.2 Regimes and Therapies

Regimes and therapies are *subtypes* of procedure. The *Fully Specified Name* semantic tag is "(regime/therapy)". This part of the hierarchy contains procedures that are defined as aggregates of sub-procedures which are either: (1) repeated multiple times, over the space of an extended period of time, or (2) are focused on a single purpose but do not have any single sub-procedure as a necessary part. As an example of a "regime" type procedure that has sub-procedures repeated multiple times, consider | antineoplastic chemotherapy regimen |. This term refers to a set of procedures which might include individual instances of administration of a chemotherapy agent, the instances being done at separate times over a pre-determined or planned period of days or weeks. Another example could be "low-dose aspirin therapy for prevention of coronary thrombosis," though this is not currently present in the terminology. In this example, the meaning refers to a repeated administration of aspirin in a small dose, for an indefinite period of time.

An example of a regime/therapy that involves a single purpose but has no defining necessary sub-procedure is | cast care |. In this case, the sub-procedures are all done for the purpose of properly monitoring and maintaining an orthopedic cast, but the exact sub-procedures done in any one instance may vary from one cast to the next, or from one patient to the next, or from one hospital or care setting to the next. Sub-procedures might include inspection for moisture, examination for cracks, checking skin condition, examining and/or replacing padding, and so forth. The common defining feature is not any one necessary sub-task, but rather the purpose for which the set of sub-tasks is done - in this case, to take care of an orthopedic cast.

Regimes/therapies may be the value (the *object*) of the attribute | has focus |, as in the case of | cardiac rehabilitation assessment |, with | has focus | = | cardiac rehabilitation (regime/therapy) |. They may also be the *subject* of | has focus |, as in the case of | occupational therapy surveillance (regime/therapy) | with | has focus | = | occupational therapy (regime/therapy) |.

6.2.3.3.5.2.1 Navigational concepts versus regimes or therapies

To tell the difference between a disjunctive/navigational procedure and a regime, ask whether an instance of this category would be a set of procedures done on a single patient (in which case it is a regime), or instead is a single procedure done on a single patient (in which case it is disjunctive/navigational). The instances of a disjunctive or navigational procedure category (the acts in reality that the category refers to) represent one particular procedure that is done on one particular patient. A disjunctive/navigational term example is “excision of lesion of ear, nose, or throat”. The excision of a lesion from the nose of Mr. Smith on April 23rd would be an instance (individual) belonging to the class "excision of lesion of ear, nose, or throat". In contrast, the instances of a regime are sets, sequences or groups of procedures. For example, an instance of "cast care" could be the specific cast care for Mr. Smith’s cast on the morning of April 23rd, consisting of the set of procedures: (1) examining the cast; (2) examining his arm; (3) asking for his symptoms; and (4) cleaning the skin. It is possible to have an instance of cast care in which the set includes only one procedure (e.g. examining Mr. Smith’s cast), but regimes must have the ability to reference an instance consisting of a set of multiple procedures, while disjunctive categories reference instances that are single.

6.2.3.3.5.3 Endoscopy, endoscopic procedures

Endoscop**ic** procedures are distinguished from endo**scopy** procedures. The chief distinction depends on the main action (| METHOD |). If the main action is inspection using an endoscope, this is an endo**scopy**. If the main action is some other value for | METHOD | that is carried out by gaining access to the procedure site via an endoscope, this is an endoscop**ic** procedure, and it is modeled with| USING ACCESS DEVICE | to specify that the endoscope is used to access the site. The main action of an endoscop**ic**procedure is not the action of inspection with the endoscope. Therefore, the action of inspecting using the endoscope is not modeled unless the FSN also specifies endoscopy or inspection. For procedures where the emphasis is on inspection using a device (e.g. endoscopy), the attribute| USING DEVICE | is used instead of | USING ACCESS DEVICE |.

6.2.3.3.5.4 Centesis procedures

Centesis may be defined as the act of puncturing a body cavity or space with a hollow needle and drawing out fluid. Each centesis procedure thus involves both a puncture action and a needle aspiration action. It is correct to have two role groups for centesis procedures. One group has a METHOD equal to puncture action and a | Procedure site - Direct | equal to the structure being punctured. For thoracentesis, the direct site would be the pleura. The second role group has a METHOD equal to aspiration action and has a PROCEDURE-SITE INDIRECT equal to the space being aspirated. For thoracentesis, the indirect site would be the pleural cavity.

The value *centesis action* should be retired and not used, since it is actually two different actions with different direct and indirect objects.

6.2.3.3.5.5 Transplantation and grafting

Transplantation and grafting have very similar meanings, but they are not the same. Some procedures are both transplantation and grafting. Some are grafting but not transplantation, and some are transplantation but not grafting.

Grafting includes procedures that are not transplantation: The noun graft might be defined as any free (unattached) tissue or organ for transplantation. However, over time the meaning has been extended to include artificial grafts and implants that are not biological in origin. On the other hand, transplantation has not acquired this extended meaning; all transplants consist of biological material.

In the action hierarchy, grafting is defined as a kind of surgical introduction. Thus all procedures defined with the action grafting action will be surgical procedures.

Transplantation includes procedures that are not grafting: The action grafting necessarily implies that the action by the performer of the deed involves attachment or fixation of the (biological or artificial) graft into its place in the recipient. Most transplantations also involve such attachment or fixation, but not all. Most notably, bone marrow and stem cell transplantation does not involve the action of attachment or fixation. Rather, the action or deed is merely infusion. The infused cells individually find their way to the bone marrow or other sites where conditions are right for their growth and differentiation.

In the action hierarchy, transplantation is not a kind of surgical action. If it were, then bone marrow transplantation would be a surgical procedure, but it clearly is not.

Summary: Procedures that involve the attachment or fixation of biological tissue are both kinds of grafting and kinds of transplantation. If the grafted material is not biological, then the procedure cannot be a type of transplantation. If the transplanted material is not attached or fixed in place, then the procedure cannot be a type of grafting.

6.2.3.3.5.6 Measurement procedures and laboratory procedures

Measurements are observations that designate the value of a property, quality or *attribute* that is inherent in the individual or population (or their specimens, by proxy), according to specified rules. Although measurement is generally considered to be the observation of a quantitative value for a quality or *attribute*, measurements need not necessarily result in a numeric or ordinal result. In other words, detection (detected/not detected) and identification (selection of one or more possibilities from a specified set by detecting their presence or absence) are considered types of measurement procedures. This is admittedly a broad definition, but does require that measurement procedures be done according to pre-determined rules and that they specify the property, quality or *attribute* that is being measured. Measurement can definitely be done by physical examination techniques as well as by laboratory techniques, but physical examination by itself is not a kind of measurement. Of course, several of the routine procedures carried out during a physical examination involve measurements of properties such as height, weight, vital signs, *range* of motion, deep tendon reflexes, etc. However, the interpretation of primary observations as being normal or abnormal is not considered a kind of measurement, since normality is not an inherent property, quality or attribute that can be measured but rather a second-level interpretation of where the primary value lies relative to a *range* determined externally to the individual.

6.2.3.3.5.6.1 Laboratory procedures

The difference between a measurement and a laboratory measurement is difficult to reproducibly define. Common language convention allows the category lab tests to include procedures like prothrombin time (PT or INR), even when the test is not performed in a laboratory. Diabetics routinely measure their blood glucose are they doing a lab test? Bedside testing and intraoperative testing are expanding, and blurring the distinction between what is done in a laboratory and what is not. As a result, measurement procedure is to be preferred for naming procedures that measure all types of analytes. The term lab test should be regarded as a useful grouper for interface terminology, but not as a necessary definitional supertype of measurements that are ordinarily done in a hospital laboratory (since they might also be done in a non-laboratory setting).

Measurement procedure *concepts* that do not otherwise specify should be assumed to refer to the entire process of measurement, which may include obtaining an appropriate specimen, preparing the specimen, and carrying out the analysis and reporting the result. These procedure codes do not narrowly refer only to an intra-laboratory activity. In other words, unless otherwise specified, the tests are named primarily from the perspective of what is done for the patient (what is measured), rather than from the perspective of what each health care worker does to accomplish the task.

Likewise, non-measurement procedures involving patient specimens should be regarded as referring to the entire procedure that is done for that patient. The difference this makes inmodeling is that we should not add an *attribute-value pair* that means done in a laboratory in order to define these procedures (not even as necessarily also conditions).

Examples:

Serum sodium measurement [104934005 | Sodium measurement, serum |]: This procedure *concept* refers to a sequence of actions that may include obtaining the specimen (if necessary - but may not be if measured by an intravascular device), preparing the specimen, running the test, and reporting the result.

PAP test [119252009]: This *concept* refers to a sequence of actions including obtaining the specimen, making the smear, staining, screening, interpreting and reporting.

Preparation of cytologic smear from genital source [90226004 | Cytopathology procedure, preparation of smear, genital source |]: This is the smear preparation only, which includes staining if staining is done, but excludes obtaining the specimen, examining the slide microscopically, interpreting the findings, or reporting.

6.2.3.3.5.7 Radiographic procedures

| Radiographic imaging procedure | [363680008 | Radiographic imaging procedure |] was created as the top of a hierarchy of imaging procedures utilizing x-rays. | Diagnostic radiologic examination | [38743002] had a synonym of "X-ray", but this code has been retired because it may have been interpreted more narrowly, in particular because of the possibly narrower interpretation of radiologic versus radiographic, and the possibly narrower interpretation of diagnostic. Nevertheless, the phrase "diagnostic radiography is allowed in many of the FSNs of *subtypes* of | radiographic imaging procedure |.

6.2.3.3.5.8 Imaging guidance procedures

The imaging guidance aspect of procedures can be modeled using the existing attribute HAS INTENT. The *concept* | guidance intent (qualifier value) |, a child of | Intents (nature of procedure values) (qualifier value) |, was created to be the value for HAS INTENT for imaging guided procedures.

Example:

Computerized tomography guided biopsy of brain (procedure)

- Role Group 1
  - METHOD: Biopsy - action (*qualifier* value)
  - PROCEDURE SITE-DIRECT: Brain structure (body structure)
- Role Group 2
  - METHOD: Computed tomography imaging - action (*qualifier* value)
  - PROCEDURE SITE-DIRECT: Brain structure (body structure)
  - HAS INTENT: Guidance intent (*qualifier* value)

Computerized tomography guided biopsy of brain (procedure) would then be subsumed by Biopsy of brain (procedure) and by Computerized axial tomography of brain (procedure).

The term "Fluoroscopic ***Y***" will be interpreted as meaning "***Y*** using fluoroscopic guidance (procedure)". Procedures such as | Biopsy of wrist using fluoroscopic guidance (procedure) | will be subtypes of | Fluoroscopy (procedure) |.

6.2.3.3.5.9 Procedure on bone - Procedure on skeletal system

There are five anatomical *concepts* related to "bone".

1. Bone (tissue): the tissue type that makes up bones.
2. Bone (organ): individual particular bones, such as femur, tibia, ulna, scaphoid, lunate, etc.
3. Skeletal system subdivision: groupings of bones taken together, such as spine, skull, bony pelvis.
4. Bone (system): the pars ossea systematis skeletalis, the bone part of the skeletal system
5. Skeletal system: the entire skeletal system, including both the bones and the part of the skeleton composed of cartilage.

Because bone (tissue) is part-of bone (organ), and bone (organ) is part-of bone (system), we can use bone (system) to define aggregate terms that involve bones. Since the skeletal system includes the bones and cartilage of the skeleton, it may be possible to have a procedure on the skeletal system that is not a procedure on bone.

[Note: For now, we have made "skeletal system subdivision" also a part-of bone (system). This may need to change if there are procedures on cartilaginous skeleton that involve skeletal system subdivisions.]

6.2.3.3.5.10 Repair of fistula Closure of fistula

These are considered to have the same meaning, because *closure - action* is a kind of *repair action*, and because repair of a fistula involves closing it. In other words, all fistula closures are auto-classified as kinds of repair procedures, and we model all fistula repairs using *closure action*.

6.2.3.3.5.11 Excisional Incisional ectomy, Excision, total excision, partial excision

These terms are sometimes very difficult to interpret. We have organized excisions of or from organs according to the following general structure:

- [Organ] excision = any excisional act involving the organ (usually [organ]-ectomy is a synonym, but see next)
  - Complete excision of [organ] (sometimes [organ]-ectomy)
    - Excisional biopsy of entire [organ] (e.g. lymph nodes, testis, ovaries)
  - Partial excision of [organ]
    - Excision of lesion from [organ] (may be partial or complete removal of lesion)
      - Incisional biopsy of [organ] = incisional biopsy of lesion of [organ]
      - Excisional biopsy of [organ] structure (excisional biopsy of lesion of [organ], excisional biopsy of tissue of [organ])

- Biopsy
  - Open biopsy (biopsy done by open approach; usually is incisional)
  - Incisional biopsy
  - Excisional biopsy
    - Excisional biopsy (entire [organ])
    - Excisional biopsy (lesion)

Notes on the structure:

1. "[Organ]" is a generic placeholder for any particular organ.
2. "[Organ] excision" does not specify whether it is complete or partial, nor does it specify what is excised.
3. Sometimes, the -ectomy word is a synonym of complete removal; when it is, the full name will specify "complete organ-ectomy".

4. An excision is not necessarily a biopsy, nor are all biopsies excisions (e.g. brush biopsy).
5. | Incisional biopsy of [organ] | necessarily implies incision and removal of a lesion, and is by definition a partial excision, since the site is the organ, and an excision is done, but the entire lesion is not necessarily removed.
6. | Excisional biopsy of [organ] | generally means that *tissue* or a *lesion* or suspected lesion is necessarily entirely excised, not the entire organ, except in the case of small endocrine glands and lymph nodes, in which an excisional biopsy takes the entire gland.
7. An excisional biopsy of a lesion of an organ is a partial excision of (from) the organ. This is true even when small polyps are removed. Specifying "partial excision" does not differentiate between those excisions that remove irreplacable tissue and those that do not (e.g. a segmental resection vs polypectomy of intestines).

6.2.3.3.5.11.1 Biopsy and Excision

Biopsies, like other removals, can have two direct objects, one the morphology and the other the site. It is therefore alright to use | Procedure site - Direct | for biopsies, even if *subtypes* might have a direct object that is a morphology.

6.2.3.3.5.12 Immunization and Vaccination

Immunization may be accomplished by active immunization (introduction of a vaccine), or by passive immunization (introduction of immunoglobulin / antibodies). Vaccination, by definition, is the introduction of a vaccine, and is therefore synonymous with active immunization, since a vaccine is a substance that can induce active immunity. We have changed the preferred name of some terms that formerly said "immunization" to be "vaccination", where it is clear that vaccination was intended. In other cases, we have created a new subordinate term for vaccination, and left the original immunization term as a superordinate term to encompass both active and passive immunization procedures.

6.2.3.3.5.13 Division, incision, transection, bisection, and osteotomy

Division action is defined as a *subtype* of | Incision - action |. This does not necessarily mean that all procedure names that include the word division are necessarily to be modeled with| METHOD | = *division-action*.

The exception is those procedures where the division is accomplished using blunt dissection and not incision. For example, division of adhesion *concepts* should be modeled the same was as lysis of adhesions procedure *concepts* using *dissection - action*. Adhesions are broken down by blunt dissection, often without incising them, though in an open-world model this does not exclude procedures that may also involve division by incision.

The preferred name of the division of adhesion *concepts* can be changed to lysis of adhesions for the sake of consistency and to avoid the incorrect modeling that might occur from interpreting these divisions as necessarily being kinds of incision.

6.2.3.3.5.13.1 Transection and bisection

Transection is a division across the longitudinal axis of a structure. Bisection is division into two parts by cutting. The action *concept* | Transection - action | is a *subtype* of *bisection action*, which in turn is a *subtype* of *division action*, and is also a *subtype* of | Incision - action |.

This assumes that transection is accomplished by cutting.

6.2.3.3.5.13.2 Osteotomy

One definition of the word osteotomy is cutting into or through a bone. This creates three possible meanings in the terminology:

1. Cutting into a bone, regardless of whether the bone is divided thereby (general *concept*)
2. Cutting through a bone and dividing it
3. Cutting into a bone without cutting through it and therefore without dividing it (incision without division)

The first meaning, incision, is to be modeled using METHOD = incision action, and PROCEDURE-SITE-DIRECT = bone structure (or *subtypes*).

The second meaning, division by cutting, is to be modeled using METHOD =*division action*, and PROCEDURE-SITE-DIRECT = bone structure (or *subtypes*).

The third meaning, incision without division, does not appear to be needed. Those procedures that do not explicitly involve division can be modeled as simply incision. Any incision procedure that necessarily *must not* involve division (as opposed to *ordinarily would not* involve division) would have to remain *primitive* until such time as a negation operator is added to the logic repertoire.

Osteotomy - action (*qualifier* value) is regarded as ambiguous and should be (or remain) retired.

Incision of bone had a synonym of incision of bone without division, but it cannot retain this synonym and at the same time retain division of bone as its *subtype*. Therefore this synonym is retired as not valid.

6.2.3.3.5.14 Open reduction and internal fixation (ORIF)

This phrase includes two accomplishments (reduction and fixation) which are accomplished by two different means (open manipulation of the fracture, and insertion of an orthopedic fixation device). This provides a clear opportunity for general *concept* inclusion axioms (GCIs) in order to fully represent the meanings without imposing a heavy modeling burden on those who do *postcoordination*.

The simplest path is to say that open reduction of a fracture necessarily involves open manipulation of the fracture; and that internal fixation of a fracture necessarily involves the insertion of an orthopedic internal fixation device.

6.2.3.3.5.15 Encounter

An encounter is defined as an in-person meeting between a patient and a health care provider for the purpose of the provision of health care services to the patient. These are defined as kinds of procedure.

An indirect encounter is not actually an encounter, since there is no face-to-face meeting. Therefore encounter and indirect encounter are siblings in the procedure hierarchy.

6.2.3.3.6 Changes and historical notes

6.2.3.3.6.1 Surgical procedures in Clinical Terms version 3

The surgical and related procedure domains in *Clinical Terms Version 3* was generated from the OPCS4-based Chapter 7 of Read Version 2 by the addition of new *concepts* during the Clinical Terms Project (CTP) and subsequent refinement.

6.2.3.3.6.2 ROUTE OF ADMINISTRATION added

This attribute was added in the January 2006 release to allow a procedure to be more fully modeled so that its definition includes the route of administration of a given substance.

6.2.3.3.6.3 USING DEVICE replaced USING

| USING DEVICE | replaced the attribute | USING | which was retired as of the January 2007 release. The retired attribute | USING | allowed values that included *descendants* of | Physical force (physical force) | which are not actually devices. Additionally, the new | DEVICE | attributes are intended to clarify the inconsistency that existed over when to use the attribute | USING | versus | ACCESS INSTRUMENT | versus | ACCESS |, particularly for | Endoscopic procedure | .

6.2.3.3.6.4 | ACCESS | not used to model endoscopic route of access

As of the January 2007 release, | ACCESS | was no longer used to capture the fact that the route used to access a procedure was endoscopic. The information that was previously captured by | ACCESS | | Endoscopic approach-access (qualifier value) | is adequately captured with | USING ACCESS DEVICE || Endoscope, device (physical object) |.

6.2.3.3.6.5 USING SUBSTANCE and USING ENERGY added

These attributes were introduced in the January 2007 release.

6.2.3.3.6.6 APPROACH retired

The attribute APPROACH was retired for the January 2008 releases because its use for non-surgical procedures was not reproducible

6.2.3.4 Observable Entities

The observable hierarchy is currently at the top level, with several immediate groupers as children:

- *SNOMED CT concept*
  - observable entity
    - age AND/OR growth period
    - body product observable
    - clinical history/examination observable
    - device observable
    - drug therapy observable
    - environment observable
    - feature of entity
    - function
    - general clinical state
    - hematology observable
    - identification code
    - imaging observable
    - interpretation of findings
    - molecular, genetic AND/OR cellular observable
    - monitoring features
    - population statistic
    - process
    - radiation therapy observable
    - sample observable
    - social / personal history observable
    - substance observable
    - temporal observable
    - tumor observable
    - vital signs

Observables are considered to be partial observation results, where there is a defined part of the observation missing. In many cases, what is missing is a numeric value, or a numeric value with units. In other cases, the observable is like a question, and what is missing can be regarded as the answer.

Among the immediate children of observable entity, some of the categories are now regarded as not fitting this definition and therefore should be moved. At a minimum, I function I and I process I are in this group that clearly needs to be moved.

6.2.3.4.1 Observable entity

Concepts in this *hierarchy* can be thought of as representing a question or procedure which can produce an answer or a result. For instance, I Left ventricular end-diastolic pressure (observable entity) I could be interpreted as the question, “What is the left ventricular end diastolic pressure?” or “What is the measured left ventricular end-diastolic pressure?”

Observables are entities that could be used to code elements on a checklist or any element where a value can be assigned. I Color of nail (observable entity) I is an observable. I Gray nails (finding) I is a finding.

One use for | Observable entity | in a clinical record is to code headers on a template. For example, | Gender (observable entity) | could be used to code a section of a template titled “Gender” where the user would choose “male” or “female”. “Female gender” would then constitute a finding.

Note: See also *Attributes used to define Evaluation Procedure concepts*.

6.2.3.4.2 Observable Entities and Evaluation Procedures

The *concept models* for observable entities and evaluation procedures are tightly linked. They employ the same set of attributes, with the exception that evaluation procedures also use METHOD. In terms of creation of codes, there will not necessarily be a one-to-one correspondence between the two hierarchies. Not every evaluation procedure will have a corresponding observable entity, and neither will every observable entity have a corresponding evaluation procedure.

The addition of new laboratory observables and procedures is subject to a Cooperative agreement with Regenstrief Institute / LOINC, which requires that there be a formal request from more than one IHTSDO Member Country before adding new SNOMED CT content that is covered by the agreement. The purpose is to minimize unnecessary duplication of effort between SNOMED CT and LOINC.

Note: At the time of publication of this guide, the concept model for observables has not yet been adopted for use with the released observables hierarchy; and some of the defining attributes (object properties) listed here have **not yet** been added to SNOMED CT in the concept model attributes hierarchy.

6.2.3.4.2.1 Evaluation Procedures

The evaluation procedure hierarchy is currently classified under "procedure by method", with several immediate groupers as children::

- Procedure by method
  - evaluation procedure
    - measurement
    - physical examination
    - monitoring
    - imaging
    - spectroscopy

Evaluation procedures can be defined by METHOD = evaluation - action in the general case. *Subtypes* of evaluation - action are used to define other *subtypes* of evaluation procedure according to a few action values: measurement, physical examination (in the medical sense of using direct inspection, palpation, percussion or auscultation), monitoring (a set of temporally repeated evaluations), imaging (the creation and evaluation of an image), or spectroscopy (the detection and evaluation of a spectrum).

6.2.3.4.3 Attributes for observable entities and evaluation procedures

6.2.3.4.3.1 Observable entity and evaluation procedure attributes: overview

Table 156: Observable entity and evaluation procedure attributes summary

| DEFINING ATTRIBUTE |	**Permissible Values (*Concepts* listed and their *descendants*)**
| PROPERTY TYPE |	| Property | |{new}| | Measurement property | 118598001 | Measurement property |

| DEFINING ATTRIBUTE |	Permissible Values (*Concepts* listed and their *descendants*)
| INHERES IN |	| Body structure | 123037004 | Body structure | | Organism | 410607006 | Organism | | Substance | 105590001 | Substance | | Specimen | 123038009 | Specimen | | Physical object | 260787004 | Physical object | | Pharmaceutical / biologic product | 373873005 | Pharmaceutical / biologic product | | Record artifact | 419891008 | Record artifact |
| INHERENT LOCATION |	| Body structure | 123037004 | Body structure | | Organism | 410607006 | Organism | | Physical object | 260787004 | Physical object |
| INHERENT INGREDIENT |	| Substance | 105590001 | Substance |
| CHARACTERIZES |	| Process | 415178003 | Process |
| PROCESS AGENT |	| Body structure | 123037004 | Body structure | | Organism | 410607006 | Organism |
| PROCESS DURATION |	| Time frame | 7389001 | Time frame |
| PROCESS OUTPUT |	| Substance | 105590001 | Substance |
| TOWARDS |	| Body structure | 123037004 | Body structure | | Organism | 410607006 | Organism | | Substance | 105590001 | Substance | | Specimen | 123038009 | Specimen | | Physical object | 260787004 | Physical object | | Pharmaceutical / biologic product | 373873005 | Pharmaceutical / biologic product | | Record artifact | 419891008 | Record artifact |
| RELATIVE TO |	| Substance | 105590001 | Substance | | Body structure | 123037004 | Body structure |

| DEFINING ATTRIBUTE |	**Permissible Values (*Concepts* listed and their *descendants*)**
| REL-TO PART-OF |	| Body structure | 123037004 | Body structure | | Organism | 410607006 | Organism | | Substance | 105590001 | Substance | | Specimen | 123038009 | Specimen | | Physical object | 260787004 | Physical object | | Pharmaceutical / biologic product | 373873005 | Pharmaceutical / biologic product | | Record artifact | 419891008 | Record artifact |
| PRECONDITION |	| Clinical finding | 404684003 | Clinical finding | | Precondition value | 703763000 | Precondition value |
| SCALE |	| Quantitative | 30766002 | Quantitative | | Qualitative | 26716007 | Qualitative | | Ordinal value | 117363000 | Ordinal value | | Ordinal or quantitative value | 117365007 | Ordinal OR quantitative value | | Nominal value | 117362005 | Nominal value | | Narrative value | 117364006 | Narrative value | | Text value | 117444000 | Text value |
| UNITS |	| Unit | 258666001 | Unit |
| TECHNIQUE |	| Technique (qualifier value) | 272394005 | Technique |
| DIRECT SITE |	| Body structure | 123037004 | Body structure | | Organism | 410607006 | Organism | | Physical object | 260787004 | Physical object | | Specimen | 123038009 | Specimen |

Note:

Permissible values for these attributes include the *concepts* listed and their *descendants*.

6.2.3.4.3.2 Property Type

This attribute is used to specify the type of inherent quality or process that is to be observed. Its values are abstract types of quality (length, odor, concentration) or abstract types of process features (rate, speed), and do **not** include qualities that are located (length of arm, odor of urine), or given a value (elevated concentration).

Table 157: Permissible values for Property Type

Concept Values	Examples
| Property type (qualifier value) | |{new}|	
| Measurement property | 118598001 | Measurement property |	| Blood glucose mass concentration (observable entity) | • | PROPERTY TYPE | | mass concentration (property) | • | TOWARDS | | glucose (substance) |

Note:

In a coming release of *SNOMED CT*, | Measurement property | will become a *subtype* of a new general *concept* | Property type (qualifier value) |

6.2.3.4.3.3 Inheres In

This attribute specifies the independent continuant in which the quality inheres, and on which the dependent quality (of this observable) depends.

Table 158: Permissible values for Inheres In

Concept Values	Examples
| Body structure | 123037004 | Body structure |	| catalytic activity content of alpha-L-iduronidase in fibroblasts (observable entity) | • | PROPERTY TYPE | | catalytic activity content | • | INHERES IN | | fibroblast (cell) | • | TOWARDS | | L-Iduronidase (substance) | • | DIRECT SITE | | fibroblast specimen |
| Organism | 410607006 | Organism |	| moxalactam susceptibility MLC (observable entity) | • | PROPERTY TYPE | | susceptibility | • | INHERES IN | | bacterium | • | TOWARDS | | moxalactam (substance) | • | TECHNIQUE | | minimum lethal concentration (technique) |
| Substance | 105590001 | Substance |	| glutamine substance concentration in plasma (observable entity) | • | PROPERTY TYPE | | substance concentration | • | INHERES IN | | plasma (substance) | • | TOWARDS | | glutamine (substance) |

Concept Values	Examples
| Specimen | 123038009 | Specimen |	| volume of 24-hour urine sample (observable entity) | • | PROPERTY TYPE | | volume | • | INHERES IN | | 24 hour urine sample (specimen) |
| Physical object | 260787004 | Physical object |	| warming-cooling mattress temperature (observable entity) | • | PROPERTY TYPE | | temperature | • | INHERES IN | | warming-cooling mattress (physical object) |
| Pharmaceutical / biologic product | 373873005 | Pharmaceutical / biologic product |	
| Record artifact | 419891008 | Record artifact |	

6.2.3.4.3.4 Inherent Location

This attribute is used to specify a body site or other location of the independent continuant in which the property inheres.

Table 159: Permissible values for Inherent Location

Concept Values	Examples
| Body structure | 123037004 | Body structure |	| DNA taxon of mycobacterium from bronchial secretions (observable entity) | • | PROPERTY TYPE | | Organism DNA taxon | • | INHERES IN | | Genus Mycobacterium (organism) | • | INHERENT LOCATION | | bronchus | • | DIRECT SITE | | bronchial secretion specimen |
| Organism | 410607006 | Organism |	
| Physical object | 260787004 | Physical object |	

6.2.3.4.3.5 Inherent Ingredient

This attribute is used to specify the ingredient substance type of the independent continuant in which the property inheres.

Table 160: Permissible values for Inherent Ingredient

Concept Values	Examples
| Substance | 105590001 | Substance |	| millimoles of lactose administered per os (observable entity) | • | PROPERTY TYPE | | Substance amount | • | INHERES IN | | dose of pharmaceutical/biologic product | • | INHERENT INGREDIENT | | lactose | • | PRECONDITION | | post administration of dose per os | • | UNITS | | millimole |

6.2.3.4.3.6 Characterizes

This attribute specifies the process which the property describes, and on which the property (of this observable) depends. The process can be very general (e.g. "excretion").

Table 161: Permissible values for Characterizes

Concept Values	Examples
| Process | 415178003 | Process |	| mass concentration ratio of silver to creatinine in 24 hour urine (observable entity) | • | PROPERTY TYPE | | mass concentration ratio | • | CHARACTERIZES | | excretion process | • | PROCESS DURATION | | 24 hours | • | PROCESS OUTPUT | | silver | • | RELATIVE TO | | creatinine | • | DIRECT SITE | | 24 hour urine sample |

6.2.3.4.3.7 Process agent

This attribute is used to specify the continuant (such as a body structure or organism) that is causally active in the process on which the property depends. It appears to have the same meaning as 'has_agent' in the OBO Relations Ontology. It may specialize the meaning of the process named as the value of CHARACTERIZES, or it may simply recapitulate the meaning that is already there. The PROCESS AGENT can be left unspecified.

Table 162: Permissible values for Process agent

***Concept* Values**	**Examples**
I Body structure I 123037004 I Body structure I	I substance rate of secretion of somatotropin by pituitary following clonidine per os (observable entity) I • I PROPERTY TYPE I I substance rate I • I CHARACTERIZES I I secretion process I • I PROCESS AGENT I I pituitary gland I • I PROCESS OUTPUT I I somatotropin I • I PRECONDITION I I post clonidine administration per os I
I Organism I 410607006 I Organism I	

6.2.3.4.3.8 Process duration

This attribute specifies the duration of the process characterized by the observable property type.

Table 163: Permissible values for Process duration

Attribute Values	**Examples**
I Time frame (qualifier value) I 7389001 I Time frame I	I mass rate of excretion of cortisone in 24 hour urine (observable entity) I • I PROPERTY TYPE I I mass rate I • I CHARACTERIZES I I excretion process I • I PROCESS OUTPUT I I cortisone I • I PROCESS DURATION I I 24 hours I • I DIRECT SITE I I 24 hour urine sample I

6.2.3.4.3.9 Process output

This attribute specifies the substance produced by the process characterized by the observable property type.

Table 164: Permissible values for Process output

Attribute Values	**Examples**
I Substance I 105590001 I Substance I	I substance rate of excretion of pregnanediol in micromoles per day (observable entity) I • I PROPERTY TYPE I I substance rate I • I CHARACTERIZES I I excretion process I • I PROCESS OUTPUT I I pregnanediol I • I UNITS I I umol/day I • I DIRECT SITE I I urine specimen I

Toward

This attribute is used to specify the third element of a relational quality, the first two elements being the type of property and the entity in which the quality inheres.

Table 165: Permissible values for Toward

<table>
<tr><th>Concept Values</th><th>Examples</th></tr>
<tr><td>| Body structure | 123037004 | Body structure |</td><td></td></tr>
<tr><td>| Organism | 410607006 | Organism |</td><td>| arbitrary concentration of Varicella-Zoster virus (observable entity) | | PROPERTY TYPE | | arbitrary concentration | | INHERES IN | (not specified) | TOWARDS | | Human herpesvirus 3 (organism) | | DIRECT SITE | | specimen |</td></tr>
<tr><td>| Substance | 105590001 | Substance |</td><td>| mass concentration of sodium in plasma (observable entity) | | PROPERTY TYPE | | mass concentration | | INHERES IN | | plasma | | TOWARDS | | sodium |</td></tr>
<tr><td>| Specimen | 123038009 | Specimen |</td><td></td></tr>
<tr><td>| Physical object | 260787004 | Physical object |</td><td></td></tr>
<tr><td>| Pharmaceutical / biologic product | 373873005 | Pharmaceutical / biologic product |</td><td></td></tr>
<tr><td>| Record artifact | 419891008 | Record artifact |</td><td></td></tr>
</table>

6.2.3.4.3.11 Relative To

This attribute is used to specify the denominator of a relational property type, such as a ratio or proportion.

Table 166: Permissible values for Relative To

<table>
<tr><th>Concept Values</th><th>Examples</th></tr>
<tr><td>| Substance | 105590001 | Substance |</td><td>| Urine alpha aminobutyrate to creatinine ratio (observable entity) | | PROPERTY TYPE | | substance (concentration) ratio | | INHERES IN | | urine | | TOWARDS | | alpha aminobutyrate | | RELATIVE TO | | creatinine | | DIRECT SITE | | urine sample |</td></tr>
</table>

***Concept* Values**	**Examples**
| Population of cells in portion of fluid | 702948002 | Population of cells in portion of fluid |	| Neutrophils per 100 leukocytes in blood (observable entity) | • | PROPERTY TYPE | | Number fraction | • | TOWARDS | | population of all neutrophils in portion of fluid | • | RELATIVE TO | | population of all leukocytes in portion of fluid | • | DIRECT SITE | | blood specimen | • | INHERES IN | | blood | • | SCALE TYPE | | quantitative | • | UNIT OF MEASURE | | percentage |

6.2.3.4.3.12 RelTo Part of

This attribute specifies the independent continuant which the value of "relative to" is part of, if different from the independent continuant in which the property type inheres. Its main use is for relative substance concentrations, where the same substance has a concentration in two different fluids. In this case, TOWARDS and RELATIVE TO will have the same substance value, and the two fluids will be represented as values of INHERES IN and REL-TO PART OF.

Table 167: Permissible values for Rel To Part of

***Concept* Values**	**Examples**
| Body structure | 123037004 | Body structure |	
| Organism | 410607006 | Organism |	
| Substance | 105590001 | Substance |	| Relative substance concentration of cerebrospinal fluid IgM to plasma IgM (observable entity) | • | PROPERTY TYPE | | relative substance concentration | • | INHERES IN | | cerebrospinal fluid | • | TOWARDS | | immunoglobulin M | • | RELATIVE TO | | immunoglobulin M | • | REL TO PART OF | | plasma |
| Specimen | 123038009 | Specimen |	
| Physical object | 260787004 | Physical object |	
| Pharmaceutical / biologic product | 373873005 | Pharmaceutical / biologic product |	
| Record artifact | 419891008 | Record artifact |	

6.2.3.4.3.13 Precondition

This attribute is used to specify body state, timing, challenges, and other situations that must be true of the entity to be observed.

Table 168: Permissible values for Precondition

Concept Values	Examples
| Precondition value | 703763000 | Precondition value |	| Plasma creatinine concentration 7 days post challenge | • | PROPERTY TYPE | | substance concentration | • | PRECONDITION | | 7 days post challenge | • | TOWARDS | | creatinine | • | INHERES IN | | plasma |
| Clinical finding | 404684003 | Clinical finding |	| Lying blood pressure | • | PRECONDITION | | supine body position |

6.2.3.4.3.14 Scale type

This attribute refers to the scale of the result of an observation or a diagnostic test (i.e., quantitative, qualitative, semi-quantitative).

Table 169: Permissible values for Scale type

Attribute Values	Examples
| Quantitative (qualifier value) | 30766002 | Quantitative | | Qualitative | (qualifier value) 26716007 | Qualitative | | Ordinal value | (qualifier value) 117363000 | Ordinal value | | Ordinal or quantitative value | (qualifier value) 117365007 | Ordinal OR quantitative value | | Nominal value (qualifier value) | 117362005 | Nominal value | | Narrative value (qualifier value) | 117364006 | Narrative value | | Text value | (qualifier value) 117444000 | Text value |	

6.2.3.4.3.15 Units

This attribute represents the units used in assigning a value to an observation.

Table 170: Permissible values for Units

Attribute Values	Examples
| Unit | 258666001 | Unit |	

6.2.3.4.3.16 Technique

This attribute links *concepts* in the | Observable entity | hierarchy to their related | Technique |.

Table 171: Permissible values for Technique

Attribute Values	**Examples**
| Techniques values (qualifier value) | 272394005 | Techniques |	

Note:

In a coming release of *SNOMED CT*, | Techniques values (qualifier value) | will have a number of new values added.

6.2.3.4.3.17 Direct Site

Direct site represents the specific entity on which the observation is directly made, and is used when the observation is indirect, such as when a direct observation is not possible to be done on the entity in which the observable inheres.

Table 172: Permissible values for Direct Site

Attribute Values	**Examples**
| Body structure | 123037004 | Body structure |	
| Organism | 410607006 | Organism |	
| Physical object | 260787004 | Physical object |	
| Specimen | 123038009 | Specimen |	

6.2.3.4.4 Converting LOINC or NPU codes into observable entities

6.2.3.4.4.1 Rules for converting LOINC names into observable entities

These rules are being developed and reviewed in collaboration with the Regenstrief Institute and LOINC, as part of a cooperation project. The rules are subject to change as the project progresses.

1. For peak and trough level measurements: COMPONENT = substance, and PRECONDITION = At time of peak level [or At time of trough level]
2. The single point in time value in the TIMING part of LOINC terms is already implied by the observable model, which refers to entities that exist at a single point in time. As a result, an additional object property is not needed in the observables model to represent the TIMING of a single point in time. In the case of tests that have TIMING values that extend across time, the object property PROCESS DURATION is used.
3. The LOINC part "^Patient" is already implied in the observables model and is not explicitly modeled. If the observable refers to someone or something other than the patient, this will be explicitly represented.
4. For antibiotic susceptibility: PROPERTY-TYPE= susceptibility, and INHERES-IN = Organism [or *subtype* of organism]
5. If the specimen is serum or serum/plasma, INHERES-IN = plasma
6. If *LOINC* component ends with .RAST class, this represents a kind of score. It is still the technique, but it is not “radioallergosorbent technique”, rather it is “RAST scoring” technique.
7. For cell antigen measurements, if the PROPERTY-TYPE is arbitrary concentration, then InheresIn should take a value = population of cells (or population of the appropriate *subtypes* of cell, e.g. population of erythrocytes, population of neutrophils, etc.)
8. For fractions and ratios, the *LOINC* component is split into TOWARDS and RELATIVE-TO
9. When *LOINC* system = xxx, INHERES-IN has no value – it remains null, and DIRECT-SITE = specimen
10. Challenge information is extracted from Component, and put in PRECONDITION.
    a. The *LOINC* “xxx” is not included in the string for PRECONDITION values.

11. Ratios and fractions are split, the “numerator” substance is put in the TOWARDS field and “denominator” substance is put in the RELATIVE-TO field
    a. The values put into TOWARDS and RELATIVE-TO are the codes for substances, not the codes for concentrations of the substances
    b. For RELATIVE-TO values, “total amount” is assumed, and the word “total” need not be added to the substance term.
12. Impression/interpretation is a valid value of PROPERTY-TYPE, but the rest of the model does not fit and therefore these terms require an expanded model (i.e. they must remain incompletely modeled at present).
13. When PROPERTY-TYPE is a feature of a process, use CHARACTERIZES instead of INHERES-IN, and the value of CHARACTERIZES is a process.
    a. Catalytic activity (and catalytic activity ratio) is measured in the lab by a process of actual catalysis by the enzyme in the sample; but the observable is intended to characterize a specifically dependent continuant, which is in this case a disposition: the point-in-time catalytic disposition of the existing quantity of enzyme in the plasma at the point in time the sample is drawn. The observable is not intended to characterize a process of catalysis that extends over time in the patient (if it were, multiple samples over time would be necessary).
14. For coagulation, many subtleties can be avoided by simply naming the TECHNIQUE. For example, the International Normalized Ratio (INR) test has TECHNIQUE = INR.
15. When *LOINC* system = dose, the PROPERTY-TYPE (e.g. mass) INHERES-IN the *LOINC* component, (put the *LOINC* component value into the INHERES-IN field). This is the active ingredient of the substance that is administered, and it is the amount of the active ingredient which is the value of the observable.
16. When measuring the total of two types of cells, the TOWARDS should be a single value representing the disjunctive category (“cell type A or cell type B”).
17. HLA antigen measurements should modeled as INHERES-IN = 108353004 cell surface structure. If the test is done on leukocytes, then SPECIMEN = 258591005 white blood cell sample.
18. For microbiology organism presence/identity observables, the component is split into PROPERTY-TYPE = Linnaean taxon, INHERES-IN = organism (or bacterium, virus, fungus, protozoan, etc if known prior to testing), and INHERES-IN-LOCATION = site of the culture. PRECONDITION is used to specify which one of a series of organisms is being identified.
19. For titers (titres), the PROPERTY-TYPE is arbitrary concentration, and the TECHNIQUE is titration (or a *subtype* thereof).
20. If *LOINC* system = XXX, DIRECT-SITE and/or INHERES-IN may be null (i.e. not given a value).
21. For microbiology (cultures, etc) reported as arbitrary concentration, *LOINC* system always generates two values, one (a body structure) for INHERES-IN and another (a specimen) for DIRECT-SITE
22. For arbitrary concentration of a cross-reacting antigen, the value of TOWARDS should be the disjunction (inclusive OR) of the entities that cross-react.
23. Gene mutation analysis in the Component field is translated into a property called “gene taxon” which can take values that name various mutations
24. For microbiology sensitivity tests measuring sensitivity to combined drug products (e.g. sulfamethoxazole/trimethoprim), the TOWARDS should be a code for the combined substance (a “portion of a mixture of sulfamethoxazole with trimethoprim”), NOT two values of TOWARDS (ie. “TOWARDS sulfamethoxazole” AND “TOWARDS trimethoprim”) – this is wrong NOR a conjunction or disjunction of the two values : “TOWARDS (a portion of sulfamethoxazole OR a portion of trimethoprim)” – also wrong “TOWARDS (a portion of sulfamethoxazole AND a portion of trimethoprim)”. – also wrong The latter really means TOWARDS null because there is no portion of sulfamethoxazole that is also a portion of trimethoprim.

6.2.3.4.4.2 Rules for converting NPU codes into observable entities

1. NCCLS antigen codes in the “proc#” field (for example, NCCLS/f89 is the allergen code for mustard) defines the value of TOWARDS.
2. IRP (international reference plasma) defines the TECHNIQUE by which the assay is calibrated.
3. System spec. = fPt (fasting patient) is translated to PRECONDITION = fasting

4. Comp spec = administered with proc# = p.o. translates to INHERES-IN = dose and PRECONDITION = administered p.o.
5. For fractions and ratios, system becomes the value of RELATIVE TO. Total amount is assumed in the RELATIVE TO slot. A substance is used as the value of RELATIVE TO, rather than a property of the substance. (creatinine, not creatinine concentration).
6. When the RELATIVE TO value is located or inherent in something other than the value of INHERES-IN, then that other site/location/substance is represented as the value of REL-TO-INHERES-IN.
7. proc# = T23:30 means PRECONDITION = 11:30 PM specimen
8. fungal DNA is defined as DNA that part-of fungus
9. coagulum retraction: define as retracted coagulum vs full coagulum. Possibly could be hidden in a TECHNIQUE value without losing any interoperability.

6.2.3.4.5 Observables and results for microbiology tests

- When Microbiology Laboratory results are encoded it is important for users to be aware of the context provided by the observation (i.e. the test performed) and therefore the implied meaning for the result value (i.e. the organism). For example, the combination of LOINC for the lab test and SNOMED CT for the organism, provides a unique and specific meaning when encoding Microbiological Laboratory tests and organism result values.
    - LOINC provides microbiology reporting codes with attributes including the property through the use of “PRID” (presence or identity) and the scale through the use of “NOM” which indicates a nominal or categorical response that does not have a natural ordering as the result value (typically the name of organism).
    - Use of Organism concepts in combination with such LOINC codes will imply that a specific organism is seen, detected, identified, isolated, or present.

  Note: An organism code on its own (in the absence of a lab test and in some cases assisted by an information model) cannot imply anything about it being present or absent, or anything other than the definition of that organism in SNOMED CT. Its detection or presence can only be implied when it is paired with other information that may come from the information model including the LOINC observation.
- Reports should use “X or Y (finding)” concepts when the use case indicates that it is assumed to be a single isolate and the lab is unable (for any reason) to differentiate for the result instance. E.g. “Human coxsackie virus or human echovirus (finding)”
- The use of terms of the form “X species”, such as "Salmonella species", is routine in laboratory reporting, but *in the context of the Linnaean organism hierarchy* there is no difference between "Salmonella species" and simply "Salmonella" the genus. In the context of a laboratory report, the term "Salmonella species" sometimes is intended to convey additional information beyond the place of the identified organism in the Linnaean hierarchy, but the intended connotation may vary from lab to lab and from organism to organism. Since the organism code represents a class of organisms, it cannot also represent what was or was not done, or what will be done, to identify the organism. It also cannot properly represent other information about the result. If there is additional information that needs to be communicated, it should be in a separate statement or comment (e.g. "further species identification pending" or "sent to reference laboratory for further identification", or "further identification to be done if clinically indicated").

6.2.3.5 Context terminological Model

6.2.3.5.1 Situation with explicit context

Concepts in the | Situation with explicit context | hierarchy (given the appropriate record structure) can be used in a clinical record to represent:

- Conditions and procedures that have not yet occurred (e.g. | Endoscopy arranged (situation) |);
- Conditions and procedures that refer to someone other than the patient (e.g. | Family history: Diabetes mellitus (situation) |, | Discussed with next of kin (situation) |);
- Conditions and procedures that have occurred at some time prior to the time of the current entry in the record (e.g. | History of - aortic aneurysm (situation) |, | History of - splenectomy (situation) |).

In each of these examples, clinical context is specified. The second example, in which someone other than the patient is the focus of the *concept*, could be represented in an application or record structure by combining a header *term* Family history with the value Diabetes. The specific context (in this case, family history) would be represented using the record structure. In this case, the *precoordinated* context-dependent *concept* | Family history: Diabetes mellitus (situation) | would not be used because the information model has already captured the family history aspect of the diabetes.

Concepts in the | Procedure | and | Clinical finding | *hierarchy* have a default context of the following:

- The procedure **has actually occurred** (versus being planned or canceled) or the finding is actually present (versus being ruled out, or considered);
- The procedure or finding being recorded **refers to the patient of record** (versus, for example, a family member);
- The procedure or finding **is occurring now or at a specified *time*** (versus some time in the past).

In addition to using the record structure to represent context, there is sometimes a need to override these defaults and specify a particular context using the formal logic of the terminology. For that reason, *SNOMED CT* has developed a context model to allow users and/or implementers to specify context using the terminology, without depending on a particular record structure. The | Situation with explicit context | *hierarchy* and various attributes assigned to *concepts* in this *hierarchy* accomplish this.

Examples of Situation with explicit context concepts:

- | Family history: Myocardial infarction (situation) | ;
- | No family history of stroke (situation) | ;
- | Nasal discharge present (situation) | ;
- | Suspected epilepsy (situation) | .

Note: See also *Attributes used to define Situation with Explicit Context concepts*.

6.2.3.5.2 The meaning of "situation with explicit context"

SNOMED CT contains some *Concepts* that include "context" information and some that are regarded as "context-free". The precise meaning of "context" in this discussion is rather elusive and it is almost certainly colored by perspective. [12] The following definition seeks to encapsulate the essential aspect of "context" in relation to *SNOMED CT*.

A *concept* includes "context" information if the name of the *concept* explicitly represents information that might otherwise be represented by another less "context-rich" *concept* in a particular structural placement within a record. Context elements typically alter the meaning in such a way that the resulting *concept* is no longer a *subtype* of the original *concept*. [13]

Context can be expressed in at least three ways:

1. *precoordinated expressions* can represent context embedded within the meaning of an existing *concept* name.
2. *postcoordinated expressions* can embed context by combining codes to make a composed *expression*.
3. Context can be represented by a code or *expression* placed in an electronic record field which has predefined meaning.

Examples:

[12] From one perspective there is arguably no such thing as a "context-free" concept since the freedom from context is itself a frame of context around a concept!

[13] The potential for a *Concept* to undergo axis modification as a result of surrounding contextual information in patient records has been explored in various initiatives. These include the standards work of CEN, HL7 and a variety of initiatives in UK, Europe, Australia and the US. Different labels have been given to similar aspects of this issue "major modifiers" (ENV13606) or "primary status terms" (NHS Clinical Terms Project), "modifiers" (GEHR, GP2GP).

- The *precoordinated expression* 266897007 | Family history of myocardial infarction | might be put directly in a blank field in a record. A family history of myocardial infarction is not a *subtype* of | myocardial infarction |, so “family history” modifies context.
- Alternatively, family history of a myocardial infarction can be represented using the *postcoordinated expression* 281666001 | family history of disorder | : 246090004 | associated finding | = 22298006 | myocardial infarction |.
- "Hip replacement <u>planned</u>" might be represented as 397956004 | prosthetic arthroplasty of the hip | within a section of record on “Planned actions”. A planned hip replacement is not a kind of hip replacement, so the “Planned actions” record section modifies context.
- The *precoordinated expression* 54355006 | intracranial injury, without skull fracture | might be placed directly in the record, or might instead be represented as 127296001 | intracranial injury | in a record that also contains a negative finding “**Absent**"+”| fracture of skull |. The record might make the *Relationship* between these two entries explicit or it might depend on their temporal *Relationship*. A disorder "without skull fracture" is not a *subtype* of skull fracture, so “without” modifies context.
- | Normal blood pressure | might be placed in a field labelled as “<u>Goal</u>”. A goal of a normal blood pressure is not a kind of | normal blood pressure |, so “goal” modifies context.

SNOMED CT requires a terminological model to consistently cope with both *precoordinated* and *postcoordinated* contextual constructs. Designers and implementers of systems need guidance to identify which fields within their record structure will critically affect the meaning of the placed code. They require open strategies of dealing with such instances to preserve meaning if retrieved or transferred and allow detection of *equivalence* to constructs derived from alternative approaches.

6.2.3.5.2.1 Variable meanings according to context of use

The implicit contextual assumptions noted above are by no means universal. The same *Concepts* can be used in many different ways with quite different intended meanings. The following list identifies some of the different contexts of use.

- A | disorder | *Concept* might also be recorded to represent:
  - A possible diagnosis or part of a differential diagnosis
  - A diagnosis applied to a family member or some other contact person
  - A diagnosis explicitly excluded
  - A diagnosis now known to be incorrect but which was the basis for a particular course of treatment.
  - An absent feature of a related disorder.
  - A diagnosis that the patient believes or fears they have.
- A | procedure | *Concept* might also be recorded to represent:
  - A requested, recommended or planned procedure
  - A procedure for which consent has been given or withheld
  - A procedure that is contra-indicated
  - A procedure that has been canceled or postponed
  - A procedure for which follow up is now being arranged.
  - A procedure which is the cause of a complication.
- A **symptom** *Concept* might also record
  - Confirmed absence of that symptom
  - A symptom deduced and reported by a third party as a witness of a clinical event.
  - Inability or failure to obtain information about a symptom
  - A symptom which the patient is advised to respond to in a particular manner.
- A **finding** *Concept* might also record
  - Absence of that finding
  - Inability or failure to check for that finding

    - A finding which if present is to trigger a particular change in clinical management.
    - A finding stated which is the goal or target of a treatment.

- A **therapeutic product** *Concept* might also record

    - An allergy or other contraindication to that product.
    - An assertion than that product caused a particular side effect.

        - Various types of therapeutic activity involving the product
        - Advice to a patient to take a treatment (i.e. over the counter products)
        - Clinical authorization of one or more prescriptions
        - The issuing of prescription for a course of treatment
        - Supply (dispensing) of a specified quantity of that product
        - Administration of a single dose of a product
        - Changes treatment dosage
        - Discontinuation of a course of treatment with that product

    - A recommendation from a specialist to treat a patient with a particular product if certain circumstances apply.

6.2.3.5.2.2 Critical Record Instances

When a user places a *concept* from *SNOMED CT* within an electronic record the action of chronicling transforms the *concept* from being a theoretical representation of a clinical notion to an actual instance within the owner (patient) of the record. For example if the *concept* Meningococcal meningitis is entered into a patients electronic record it usually indicates that the patient has had an instance of this disease. Similarly the entry of Cholecystectomy would imply that the patient has undergone this procedure.

Each record system is structured differently and might use different field definitions that could subtly affect the exact meaning of the entered *concept*. The placement of a *concept* into a record field can be identified to affect the meaning in one of two ways:

- **Affect the quality of the meaning but not the instance**

  The placement of Angina in fields labelled “Current problems”, "Past medical history" or “History of” all indicate that an instance of angina has occurred in the patient but assign different timescales and possible significance to the event (and the *concept* assumes a status compatible with its default value status).

- **Critically affect the meaning and the instance**

  In this case the placement in the record field critically affects meaning. For example the placement of Parkinsons disease within a field designed for recording “Family history” or Coronary angioplasty within a field allocated for “Planned procedure” does not indicate an instance of the *concept* in the owner (patient) of the record. The adopted status is not compatible with the default value status. In these circumstances the designer of the system will need to identify the appropriate context values from a authoritative list to be linked to *concepts* placed within the fields to substitute their default values.

The context model uses attributes which reference values which critically affect the meaning of a *concept* when applied (and change the axis of class to which they relate).

6.2.3.5.2.3 Systematic (weak) defaults

Procedures and clinical findings have systematic un-stated defaults. These are weak defaults in that they are the default value if the *concept* appears in a record with no *explicit context*. For example, if knee pain appears in a record with no context, we assume it is present in the patient, and current. If gastrectomy appears with no context, we assume it is done to the patient, at the current time. Observable entities behave like clinical findings when given a numerical value or qualitative interpretation but they behave like procedures otherwise.

6.2.3.5.2.4 Axis modifiers

An axis modifier is an *attribute-value pair* that does not result in a *subtype* of the modified *concept* when it is understood as carrying the systematic default meaning. For example, family history applied to diabetes makes I subject relationship context I=I family member I, and this is not a *subtype* of I subject relationship context I=I subject of record I (the default) for a disease (e.g. diabetes), so family history is an axis modifier and critically affects the meaning within the patient record. In contrast, specifying the precise family member with a given family history does not axis modify a representation of the combined *concept* family history diabetes e.g. ischemic heart disease I subject relationship context I=I father I

6.2.3.5.3 Attributes used to define Situation with Explicit Context concepts

Table 173: Approved Situation attributes summary

Defining Attribute	Allowable Values
I ASSOCIATED FINDING I	I Clinical finding I 404684003 (<=)(< Q) I Event I 272379006 (<=)(< Q) I Observable entity I 363787002 (< Q only) I Link assertion I 416698001 (< Q only) I Procedure I 71388002 (< Q only)
I FINDING CONTEXT I	I Finding context value I 410514004 (<=)(< Q)
I ASSOCIATED PROCEDURE I	I Procedure I 71388002 (<=)(< Q) I Observable entity I 363787002 (< Q only)
I PROCEDURE CONTEXT I	I Context values for actions I 288532009 (<=)(< Q)
I TEMPORAL CONTEXT I	I Temporal context value I 410510008 (<=)(< Q)
I SUBJECT RELATIONSHIP CONTEXT I	I Person I 125676002 (<=)(< Q)

Note:

Meaning of Allowable Values (*Range*) notations:

(<<) this code and *descendants*,
(<) *descendants* only,
(<=) *descendants* only (stated) except for supercategory groupers,
(==) this code only,
(< Q) *descendants* only when in a qualifying *Relationship*,
(< Q only) *descendants* only, and only allowed in a qualifying *Relationship*.

Note: See also *Situation with explicit context*.

6.2.3.5.3.1 Context

The meaning conveyed by a *SNOMED CT* code in a medical record is affected by the context in which it is recorded. For instance, the code for "breast cancer" might be used to indicate a family history of breast cancer, a past history of breast cancer, or a current diagnosis of breast cancer. Each of these three meanings differs in regard to the context in which breast cancer is being described. Family history of breast cancer refers to breast cancer occurring in a family member of a patient. Past history of breast cancer indicates that the breast cancer occurred in the patient, at some *time* in the past, and it is not necessarily present now. Current diagnosis of breast cancer indicates that the breast cancer is present now, and in this patient. These differences are important for data retrieval, because it would be incorrect when searching for patients with breast cancer to retrieve those who merely have a family history of breast cancer.

6.2.3.5.3.2 Default Context

When a *SNOMED CT* code appears in a record without any explicitly stated context, that code is considered to have a default context. The default is "soft" in that it can be over-ridden by information carried in the structure of the record or its information model.

The default context for a clinical finding code implies that the finding has actually occurred (vs. being absent), that it applies to the subject of the record (the patient), and that it is occurring currently or occurred at a past *time* that is given by a date - *time* record linked to the code.

The default context for a procedure code implies that the procedure was completed, that it was performed on the subject of the record (the patient), and that it was done at the present *time* or in the past at a *time* that is given by a date - *time* record linked to the code.

6.2.3.5.3.3 Axis Modifiers

The six attributes used to define situation codes permit explicit (rather than default) representation of various contexts. These attributes can change the meaning of a clinical finding or procedure code in a way that changes the *hierarchy* (or "axis") of the code from | Clinical finding | or | Procedure | to | Situation with explicit context |. The resulting modified meaning is not a *subtype* of the original meaning of the code, and therefore the axis-modifying attributes are not used to qualify the code, but instead are used to qualify a "situation" code.

For instance, if | Fine needle biopsy (procedure) | is given the non-context modifying attribute | Procedure site - Direct | and a value of | Urinary bladder structure (body structure) |, the resulting *concept* | Fine needle biopsy of urinary bladder (procedure) | is still a *subtype* of the original *concept* | Fine needle biopsy (procedure) |.

However, the *concept* | Urine protein test not done (situation) | uses the context-modifying attribute | PROCEDURE CONTEXT | and a value of | Not done (qualifier value) |, and the resulting *concept* is not a *subtype* of | Urine protein test (procedure) |. Its axis (*hierarchy*) has been modified.

6.2.3.5.3.4 Overview of context attributes

Of the six attributes applied to *concepts* in the | Situation with explicit context | *hierarchy*, two are used only in representing the context in which a | Clinical finding | is recorded, (| ASSOCIATED FINDING | and | FINDING CONTEXT |); two are used only in representing the context in which a | Procedure | is recorded (| ASSOCIATED PROCEDURE | and | PROCEDURE CONTEXT |); and two attributes are used in representing the context of both | Procedure | and | Clinical finding | (| SUBJECT RELATIONSHIP CONTEXT | and | TEMPORAL CONTEXT |).

6.2.3.5.3.5 ASSOCIATED FINDING

This attribute links *concepts* in the | Situation with explicit context | *hierarchy* to their related | Clinical finding |. It specifies the | Clinical finding | *concept* whose context is being modified.

Table 174: Permissible values for ASSOCIATED FINDING

Attribute Values	**Examples**
| Clinical finding | 404684003 (<=)(< Q) | Event | 272379006 (<=)(< Q) | Observable entity | 363787002 (< Q only) | Link assertion | 416698001 (< Q only) | Procedure | 71388002 (< Q only)	| Family history of stroke (situation) | • | ASSOCIATED FINDING | | Cerebrovascular accident (disorder) |

Note:

When | ASSOCIATED FINDING | is used in *postcoordinated expressions*, its range is broader than when used in distributed content.

| ASSOCIATED FINDING | must not reference *concepts* that already have *precoordinated* context themselves.

For example, the following definition uses | FH: Thyroid disorder | incorrectly:

| History of thyroid disease in father |:

- | SUBJECT RELATIONSHIP CONTEXT |=| father |
- | ASSOCIATED FINDING |=| FH: Thyroid disorder |.

The following is the correct definition:

| History of thyroid disease in father |:

- | SUBJECT RELATIONSHIP CONTEXT |=| father |
- | ASSOCIATED FINDING |=| thyroid disease |.

6.2.3.5.3.6 FINDING CONTEXT

The FINDING CONTEXT attribute is used to represent a situation in which a *Clinical finding* is known or unknown, and if known, whether it is present, absent, or uncertain (possible); and also to express the meaning that the finding is not actual but instead an anticipated or possible future finding.

Table 175: Permissible values for FINDING CONTEXT

Attribute Values	**Examples**
| Finding context value | 410514004 (<=)(< Q)	| No cough (situation) | • | ASSOCIATED FINDING | | Cough (finding) | • | FINDING CONTEXT | | Known absent (qualifier value) |

6.2.3.5.3.7 ASSOCIATED PROCEDURE

This attribute links *concepts* in the | Situation with explicit context | *hierarchy* to *concepts* in the | Procedure | *hierarchy* for which there is additional specified context.

Table 176: Permissible values for ASSOCIATED PROCEDURE

Attribute Values	Examples
	Procedure	71388002 (<=)(< Q) | Observable entity	363787002 (< Q only)		Operative procedure planned (situation) | •	ASSOCIATED PROCEDURE		Surgical procedure (procedure)	

6.2.3.5.3.8 PROCEDURE CONTEXT

This attribute indicates the degree of completion, or *status*, of a | Procedure |, as well as its various possible future states prior to its being initiated or completed.

Table 177: Permissible values for PROCEDURE CONTEXT

Attribute Values	Examples
	Context values for actions	288532009 (<=)(< Q)		Operative procedure planned (situation) | •	ASSOCIATED PROCEDURE		Surgical procedure (procedure) | •	PROCEDURE CONTEXT		Planned (qualifier value)	

6.2.3.5.3.9 TEMPORAL CONTEXT

This attribute indicates the *time* of occurrence of the situation, indicating whether the procedure or finding that it represents is actual and therefore occurred in the present, in the past, or at a specified *time*; or that it is planned or expected, that is, temporally located in the future. The most general value is simply | Current or past (actual) |, meaning that the *concept* was actual (not planned or expected), but not specifying anything further about its *time*. The word "specified" in the | TEMPORAL CONTEXT | values means that there is a date - *time* stamp associated with the code in the record, that gives a date and/or *time*, as a point and/or interval, that applies to the *concept*.

Table 178: Permissible values for TEMPORAL CONTEXT

Attribute Values	Examples
	Temporal context value	410510008 (<=)(< Q)		History of - hematuria (situation) | •	ASSOCIATED FINDING		Blood in urine (finding) | •	TEMPORAL CONTEXT		In the past (qualifier value)	

6.2.3.5.3.10 SUBJECT RELATIONSHIP CONTEXT

This attribute is used to specify the subject of the | Clinical finding | or | Procedure | being recorded, in relation to the subject of the record. In the example below, the subject of the record is the patient and the subject who smokes is the patient's father.

Table 179: Permissible values for SUBJECT Relationship CONTEXT

***Concept* Values**	**Examples**
| Person | 125676002 (<=)(< Q)	| Father smokes (situation) | • | ASSOCIATED FINDING | | Smoker (finding) | • | SUBJECT RELATIONSHIP CONTEXT | | Father of subject (person) |

6.2.3.5.4 Contextualized Observable Entities

Observable entities, when they have not been given a value, behave like procedures with respect to the *concept model* for context. Observable entities, when they have been given a value, behave like Clinical Findings with respect to the *concept model* for context.

6.2.3.5.5 Rules for Situation Definition and Modeling

Once a *concept* code has context-shifted and become context-dependent, it is nonsense to use that *concept* code in an *expression* that once again shifts context. In other words, when one context attribute is given an axis modifying value, it forces the other context attributes to be fixed. For example, the model for Family history of diabetes is:

IS A situation-with-explicit-context

- [408732007 | subject relationship context |] family member
- [AssociatedFinding] diabetes
- [TemporalContext] current-or-past
- [FindingContext] known-present

Even though the family part of the *concept* results in an explicit axis shift of the [408732007 | subject relationship context |] only, we impose a rule that requires [TemporalContext] and [FindingContext] to be given their default values, rather than being left unspecified.

The *concept* that would be expressed as No family history of diabetes allows negation of Family history of diabetes by switching the value of [FindingContext], and changing [TemporalContext] to all times past:

IS A Situation-with-explicit-context

- [408732007 | subject relationship context |] Family member
- [AssociatedFinding] diabetes
- [TemporalContext] all times past
- [FindingContext] known-absent

What is NOT a legitimate *expression* is:

- [FindingContext] known-absent
- [AssociatedFinding] Family history of diabetes

6.2.3.5.5.1 Role grouping

In context-dependent *expressions*, [AssociatedFinding] or [AssociatedProcedure] are role grouped with the other context attributes that have a value. Also, [AssociatedFinding] and [AssociatedProcedure] must not themselves reference *concepts* that have embedded context.

The following examples illustrate the rule that, once a *concept* code has context-shifted and become context-dependent, it is nonsense to use that *concept* in an *expression* that once again shifts context. To negate a *concept* code with [FindingContext] KnownPresent, the value of [FindingContext] becomes KnownAbsent.

For example:

Family history of asthma

IS A situation-with-explicit-context

- [AssociatedFinding] asthma
- [SubjectRelContext] family member
- [FindingContext] known-present
- [TemporalContext] current-or-specified

No family history of asthma

IS A situation-with-explicit-context

- [AssociatedFinding] asthma
- [SubjectRelContext] family member
- [FindingContext] known-absent
- [TemporalContext] all-times-past

No asthma

IS A situation-with-explicit-context

- [AssociatedFinding] asthma
- [SubjectRelContext] subject-of-record
- [FindingContext] known-absent
- [TemporalContext] current-specified

6.2.3.5.5.2 Mandatory context statements

When asserting a *concept* within a record it should be mandatory to apply the three context attributes and an applicable value in order to guarantee accurate meaning if that *concept* (plus context) is subsequently transferred to another record environment.

6.2.3.5.6 Background — SNOMED CT Glossary of Context

6.2.3.5.6.1 Introduction

One problem with any discussion of context is that we use words and phrases without necessarily sharing an understanding of what they mean. To address this problem, the Context Group agreed to use the working definitions in the following sub-section.

6.2.3.5.6.2 Context

When we talk about context we are describing the effects of embedding a *concept* in a *clinical situation*.

- A *concept* is embedded in a *clinical situation* when it is used in a clinical record.

For example:

- In the abstract the *concept* "myocardial infarction (disorder)" (*ConceptId* =22298006) could be used to refer to the idea of this disorder in a general way as a pure notion. This *concept* has the potential to be used in many ways (e.g. as a literature reference, as one of a set of possible complications of smoking, as a link to a protocol for care of a patient admitted with chest pain, as a contraindication for a particular medication, as a disorder suffered by a particular patient, as a possible diagnosis justifying a particular investigation, as a diagnosis excluded by an investigation, as a condition suffered by a close relative of a particular patient, etc).
- When the *concept* "myocardial infarction (disorder)" is used in a clinical record it takes on a specific contextualized meaning. This specific meaning might be an assertion by a particular author that on a given date patient was diagnosed as having had a "myocardial infarction".

Embedding a *concept* in a *clinical situation* may elaborate the semantic interpretation of a *concept* in one of several ways. Four distinct types of elaboration are recognized for the purpose of this discussion.

- *Subtype* qualification
- Axis modification
- Affirmation or Negation
- Combination

6.2.3.5.6.3 Elaboration

We use “elaboration” to refer to any addition to or change of the meaning of a *concept* that may be brought about when it is embedded in a *clinical situation*.

6.2.3.5.6.4 Subtype qualification

We use “*subtype* qualification” [14] to refer to an elaboration of a *concept* that results in a *concept* that is a *subtype* of the original unelaborated *focus concept*.

- *Subtype* qualification refines or increases the precision of meaning of a *concept*.

For example:

- The *concept* | Fracture of femur | can be elaborated by an indication of whether the fracture is open, whether it is the left or right femur that is fractured. A patient who has an "<u>open</u> fracture of the <u>neck</u> of the <u>left</u> femur" has a type of "fracture of the femur". Therefore, refining the morphology, site and adding laterality all act as *subtype* qualifications.
- The *concept* “Asthma attack” can be elaborated by an indication of severity. A patient who has had a | severe asthma attack | has had a type of “asthma attack”. Therefore, severity acts as a *subtype* qualification.
- The *concept* “Hysterectomy” may be elaborated by specifying a priority and a particular approach. A patient who has had a "routine vaginal hysterectomy" still has had a type of “hysterectomy”. Therefore both priority and approach is a *subtype* qualification.

6.2.3.5.6.5 Axis modification

We use “axis modification” [15] to refer to an elaboration of a *concept* that results in a *concept* that is **not** a *subtype* of the original unelaborated *concept*.

- Axis modification places the elaborated *concept* in a different axis of the logical semantic hierarchy.

For example:

- The *concept* “Myocardial infarction” can be elaborated by including it in part of a clinical record specifying "family history". A record of a "family history of myocardial infarction" does <u>not</u> imply that the individual patient has had any type of “myocardial infarction”. Therefore, "family history" acts as an axis modification.
- The *concept* | Total hip replacement | can be elaborated by stating that this procedure is planned to be carried out at some future date. A record of | planned total hip replacement | does <u>not</u> imply that the patient has actually had a | total hip replacement |.

6.2.3.5.6.6 Affirmation or Negation

A *concept* may be stated in the negative in a *clinical situation* e.g. meningism not present. This creates potential for a *concept* to be used represent two meanings one of which is the inverse of the other.

[14] *Subtype qualification* has also been referred to in other works as “qualifiers” (ENV136060, GEHR, CTV3) and "secondary status terms" (NHS Context of Care). The adjective “subtype” expresses more clearly the distinctive property of a qualifier. This is helpful because “modify” and “qualify” are treated by many dictionaries and some ISO authorities as synonymous.

[15] *Axis modification* has also been referred to in other works as “major modifiers” (ENV136060), “modifiers” (GEHR), "primary status terms" (NHS Context of Care). The Context Group view was that none of these labels were sufficiently specific to convey the intended meaning unambiguously. The adjective “axis” expresses the sense of a fundamental shift in meaning in a way that should be familiar to those who used earlier “multi-axial” releases of SNOMED.

- According to perspective affirmation and negation may simplistically be viewed as inversion of the meaning of an unelaborated *concept* representing a clinical finding.
- However, the effects of negation on interpretation are profound and distinct from other elaborations and must be considered separately.

Negation, like axis modification, results in a *concept* that is not a *subtype* of the unelaborated *concept*. However, negation is a special case in that:

- It explicitly rules out the unelaborated *concept*.
  - The statement I family history of myocardial infarction I does **not** imply that "patient has had a myocardial infarction" is untrue. But “no headache” (if true) implies the statement I patient has headache I is untrue.
- Furthermore, the implications of a negative statement propagate in the opposite direction from those of a positive statement.
  - If “headache” is a *subtype* of “pain” then I patient has headache I implies the patient has some pain. However, I patient has no headache I does not imply the patient has no pain.
  - Conversely I patient has headache I does not imply the patient has an occipital headache but "patient has no headache" implies the patient does not have an occipital headache.

6.2.3.5.6.7 Uncertainty

A *concept* may be stated to be possible in a *clinical situation*. Statements that explicitly indicate uncertainty can be considered in two possible ways:

- As points on the arc between affirmation and negation.
- As a kind of axis modification.

6.2.3.5.6.8 Combination

Two or more *concepts* may be embedded in a *clinical situation* in a way that links them together.

- Linkages may include:
- Simple combination of *concepts*;
- Combination of a *concept* that is stated as present and another stated to be absent;
- An explicit typed *Relationship* between *concepts*.

6.2.3.6 Body structure

I Body structure I *concepts* include normal as well as abnormal anatomical structures. Normal anatomical structures can be used to specify the body site involved by a disease or procedure.

Examples of Body structure concepts:

- I Mitral valve structure (body structure) I ;
- I Uterine structure (body structure) I .

Morphologic alterations from normal body structures are represented in the *sub-hierarchy* I Body structure, altered from its original anatomical structure (morphologic abnormality) I.

Examples of Body Structure, altered from its original anatomical structure concepts:

- I Adenosarcoma (morphologic abnormality) I ;
- I Polyp (morphologic abnormality) I.

Note: See also *Attributes used to define Body structure concepts*.

6.2.3.6.1 Anatomical structures

The top level of the anatomy hierarchy appears as those *concepts* under physical anatomical entity which is located under the body structure hierarchy as shown in the following list:

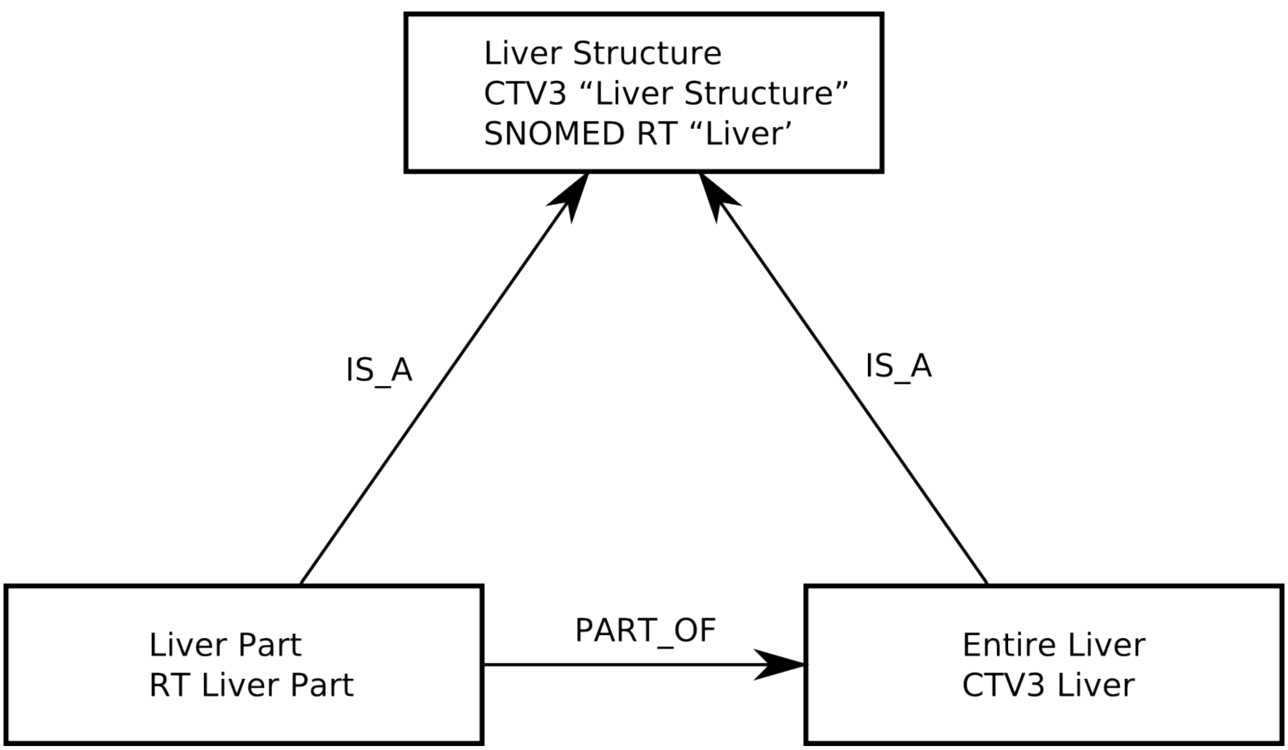
- *SNOMED CT concept*
  - Body structure
    - Anatomical or acquired body structure
      - Acquired body structure
      - Anatomical structure
    - Anatomical organizational pattern
    - Anatomical site notations for tumor staging (this needs to be *moved elsewhere*)
    - Morphologically altered structure
    - Physical anatomical entity
      - Group of anatomical entities
      - Anatomical spatial entity
      - Anatomical structure (the reified S part of the SEP triple)
        - Entire anatomical structure (the E part of the SEP triple)
        - [multiple other immediate *subtypes* of anatomical structure]

6.2.3.6.1.1 General Principles Underlying the SNOMED CT Model

6.2.3.6.1.1.1 The Structure-Entire-Part (SEP) model

SNOMED CT uses a *structure-entire-part* triple, known as the SEP triple, to represent anatomical structures.

The following *Relationships* provided a way for the anatomy in *CTV3* to be mapped to RT:



The *SNOMED CT* anatomy hierarchy differentiates classes of l entire l anatomical entities from classes of "parts of" entire anatomical entities.

Entire *concept*: Denotes a class that is instantiated by entire anatomical entities of some kind: entire heart is instantiated by all individual hearts.

Entity **P**art *concept*: Denotes a class that is instantiated by all anatomical entities that are a proper part of some entity of a given kind: heart part is instantiated by all entities that are a proper part of some heart, e.g. my mitral valve, your right ventricle, Joe's sinus node. Heart part is NOT instantiated by any heart.

Entity **S** tructure *concept*: Subsumes both the related Entire and Part *concepts*. Consequently, it denotes a class which is instantiated by anything that instantiates either the Entire or the Part. For instance, Heart structure is instantiated by my heart, my mitral valve, your heart, your right ventricle, Joe's sinus node, Joe's heart, etc.

The code named Liver structure in *CTV3* is equivalent to Liver structure in the diagram above. Both the *CTV3* code for Liver structure and the *SNOMED RT* code for Liver are interpreted to mean Some or all of the liver. *Site* attributes (| PROCEDURE SITE |, | FINDING SITE |) will usually take the value liver structure rather than | entire liver |, since typically the site of a "liver disorder" or "procedure on the liver" is not necessarily the entire liver.

6.2.3.6.1.1.1.1 Purpose of the Structure concept

Adding the Entity **S** tructure codes is a convenience to assist with the logic-based aggregation of references to the entity or its parts. The implication of this view is that the E of the SEP triple is the code that should be regarded as the one that represents the real anatomical entity that is named. For example, the code for entire liver is the one that should correspond to the code for liver in the Foundational Model of Anatomy (FMA). The *subtype hierarchy* for entire liver fits much better with the FMA hierarchies, and indeed it might be possible to completely reconcile *SNOMEDs* non-Structure components with FMA anatomy.

A database has been developed that categorizes codes in the physical anatomical entity hierarchy according to their status as S structure, P Part or E Entire, and provides the corresponding S and P code for each E code. This should provide some value to implementers. It can help with navigation, coordination with formal ontologies of anatomy, and selection of codes for *postcoordination*.

6.2.3.6.1.1.1.2 Naming conventions

"S" *concepts* are usually named | x structure (body structure) | or | structure of x (body structure) |.

"E" *concepts* are usually named | entire x (body structure) | or | x entire (body structure) |.

"P" *concepts* are usually named | x part (body structure) | or | part of x (body structure) |.

6.2.3.6.1.1.1.2.1 Plurals

Outside the anatomy section of *SNOMED*, plurals were primarily used as headers, while the individual *concept* names were singular. In the anatomy section, we have taken plurals to represent meaningful differences from their singular counterparts.

For example, cranial nerves in the FSN would mean a group of cranial nerves, while cranial nerve would not imply more than one.

6.2.3.6.1.1.1.3 Conventions for merging concepts from SNOMED RT and Clinical Terms v.3

Where there were two *concepts* with the same name, the *SNOMED RT* code was to become the "S" code, and the *CTV3* code was to become the "E" code. There are still instances of unrecognized pairing of the RT-*CTV3* "S"-"E" pair, where neither codes FSN has been changed according to the naming conventions in this document. When these unmatched pairs are identified, it is our practice to change the FSNs accordingly, and to make the "E" code have a *subtype* (*is-a*) link to the "S" code.

6.2.3.6.1.1.1.4 "S" concepts without a corresponding "E" concept

Some "S" codes do not currently have a corresponding "E" code *subtype*, and there was no policy that required that such "E" codes be created during the merger of *SNOMED RT* and *CTV3*. However, it is likely that such a policy will be enforced in the future.

6.2.3.6.1.1.1.5 S Structure codes can subsume entities other than E or P

The SEP triple may give the impression that all S codes have exactly two children, one E and one P, with all of the remaining *descendants* placed under P. Again, in the past this degree of modeling consistency was not always followed. Some codes were purposely made *subtypes* of the S that are not strictly part of the corresponding E. For example, perirenal tissue is a kidney structure but not a part of the kidney. It is used to define perirenal abscess so that it is subsumed by renal abscess. While a perirenal abscess is not strictly within the substance of the kidney, it is still considered a kind of renal abscess, and the S anatomy hierarchy is used to support this inference.

This policy has introduced undesirable variation and arbitrariness into the terminology, and future revisions will seek to eliminate these variations. Where a code is needed for a site that is really meant to extend to entities that are not part of any kidney, this will be made clear in the name, e.g. Structure of kidney and perirenal tissue.

6.2.3.6.1.1.1.6 Countable vs non-countable "E" entities

The "E" code needs to be interpreted with care when the "x" name refers to entities that do not have the property of identity, meaning that they are not countable wholes, or could be interpreted as non-countable. In this circumstance, the interpretation of E means some portion of the thing being named. Examples include tissue and types of tissue such as fascia, muscle, tendon, bone tissue, connective tissue, skin, mucosa/mucous membrane, nerve tissue, etc. Muscle, tendon, bone and skin can identify a type of tissue as well as an individual organ of that type. Bone tissue has no identity, but a particular bone does have identity.

To use skin as the archetypal example, the E code for skin of finger means a portion of the skin of a finger, so all of its *subtypes* must also be portions of skin. The S code for skin of finger then has a *subtype* P which would mean proper part of a portion of skin of finger. This admits *subtypes* that are not kinds of skin, but may be parts of skin, including layers. For example, epidermis of finger (meaning a portion of epidermis of finger) could be a proper part of a portion of skin of finger.

6.2.3.6.1.1.1.6.1 Tissues, layers, membranes: portions

We regard the E code for x tissue, x layer to have the meaning portion of x tissue, and therefore regional subdivisions of tissue types are direct *subtypes*. For example, transitional epithelium of urinary tract, as an E kind of code, should be a supertype of transitional epithelium of urinary bladder. The reason is that (portion of) transitional epithelium of urinary bladder is a kind of (portion of) transitional epithelium of urinary tract.

We also deal with layers the same way. For example, we regard serosal layer and serosa tissue as meaning the same thing, since all serosal tissue is conifigured as a layer, and it cant be a serosa without being a layer; and their E codes mean portion of serosal layer or portion of serosal tissue.

As another example, layer of retina would be a supertype of nerve fiber layer of retina, and also a supertype of retinal epithelium, where retinal epithelium represents a portion of the epithelium of the retina and is therefore a kind of (portion of) a layer.

6.2.3.6.1.1.1.7 Groups

The identity/countability issue extends to a problem differentiating groups of entities from one of the group. For example, consider x= "lymph node group", y=lymph node. In this case, the group should be linked to the member via an appropriate *Relationship* (not yet in *SNOMED*), such as has-member. In those cases where y is always necessarily a member of group x, it could be linked via a member-of *Relationship* (also not yet in *SNOMED*).

6.2.3.6.1.1.1.8 What does part-of mean?

There are several possible ways of interpreting part-of. In *SNOMED*, A part-of B means that in normal anatomy, the entire structure A is structurally included in B. Another way of saying it is that A is part-of B if there is no part of A that is not also part of B. For example, the humerus is not part-of the shoulder region, because the distal humerus is part of the humerus, and the distal humerus is not part of the shoulder region.

We do *not* use part-of for non-anatomical meanings, such as grouping tests together in batteries, nor do we use it to indicate *Relationships* that are not strict anatomical inclusion.

Some recent work has begun to differentiate between part-of that is reflexive (that is, an entity is in some sense a part-of itself, much the same that a set can be viewed as a subset of itself), versus proper-part-of, where an entity cannot be a proper-part-of itself. For now, we regard part-of *Relationships* as implying strict partonomy.

There is sometimes confusion about parthood as opposed to location. For example, an embryo is not part of a mother's body, but a kidney is. The anatomy section is composed mainly of canonical parts; but a few abnormal parts are included to permit them to be used as the location of tumors or injuries. For example, a Meckels diverticulum is a body structure that is part of the small intestine, and it is also a morphological abnormality. Likewise some stomas and other post-surgical structures are considered part of the body. A transplanted liver or kidney would be considered part of the body, as a post-surgical structure, even though the transplanted organ is not genetically identical. Likewise transplanted bone marrow is part of the body.

Non-living implants and devices, and foreign bodies, on the other hand, are considered to be located in the body but not part of the body.

6.2.3.6.1.1.1.9 Can the SNOMED CT relationships table be used to construct a part-of hierarchy?

The currently distributed part-of *Relationships* need to be much more extensively modeled and quality assured. At present they are not "defining", that is, their *CharacteristicType* in the *relationship file* is "additional", and therefore they do not affect the classifier behavior. A substantial amount of effort has gone into a draft of the updated part-of *Relationships*; these will require review and approval before incorporation into the release. This will eventually result in the SEP triplet structures and part-of relations being strictly paralleled. It is a matter of time to implement and quality assure the changes.

6.2.3.6.1.1.1.10 Why are part-of relationships not "defining"?

The SEP structure, combined with the inference mechanism that is used with *SNOMED CT*, allows us to take advantage of anatomical *Relationships* to infer subsumption (*is-a*) *Relationships* between disorders, procedures, and other entities without reference to part-of *Relationships*. The SEP structure also permits us to fully define anatomical structures without reference to part-of *Relationships* (making them "necessarily true" but not among the "necessary and sufficient" conditions). For example, the | Structure of left hand | can be *fully defined* as a | hand structure | with | laterality |=*left*. This definition is sufficient. Converting the part-of *Relationships* to have CharacteristicStatus = defining will require significant changes to the current model.

6.2.3.6.1.1.1.11 Entities with mass versus purely spatial massless entities

Points, lines, and surfaces can be considered to be massless. The FMA calls these immaterial. It is important to differentiate the codes/names for these entities from those that are intended to represent entities that have mass. At present, the *concepts* under anatomical spatial entity represent massless entities. Massless entities are not represented using the SEP model. It is conceivable that users may want to reference parts of a surface, and to enable this we would need to apply the SEP model to anatomical spatial entities, or else adopt defining part-of *Relationships*.

6.2.3.6.1.1.2 Attributes used to define Body structure concepts

Just one attribute is used in Anatomy, namely, | Laterality | .

Note: See also *Body structure*.

6.2.3.6.1.1.2.1 LATERALITY

This attribute provides information on whether a body structure is left, right, bilateral or unilateral. It is applied only to bilaterally symmetrical body structures which exist on opposite sides of the body.

Table 180: Permissible values for LATERALITY

Attribute Values	**Examples**
| Side | 182353008 (<=)	| Left kidney structure (body structure) | • | LATERALITY | | Left (qualifier value) |

Note:

Permissible values for this attribute include the *descendants* of the *concept* listed, except for super category grouper *concepts*.

6.2.3.6.1.2 Specific policies related to anatomy

6.2.3.6.1.2.1 Body parts – Body regions

We have made some use of the FMA definition of "body part" and "body part subdivision". The various joint regions listed below are classified as body part subdivisions, since that is what is intended by the various diseases and procedures that use these codes in their definitions. They are not body parts because they are defined not by a set of bones but rather by a particular joint and its surrounding structures. However, our interpretation of the word region is according to common usage and is intended as a three-dimensional structure, *not* the FMA two-dimensional definition of body region. In other words, these regions are not simply virtual surface regions, but include the three dimensional structures as well. They include the overlying skin, the subcutaneous tissues, the bones, muscles, tendons, fascia, vessels and other included organs and tissues in the region.

6.2.3.6.1.2.1.1 Surface regions

Several codes contain the phrase surface region. These could be interpreted as massless (immaterial) mathematical surfaces, but a clinical terminology would have no direct use for such meanings in clinical records. They could be interpreted as having mass (not immaterial), but the depth then is a matter for fiat declaration. Should it be just skin deep, or should it include deeper layers of the surface? If only skin deep, the meaning of these codes would overlap with codes for skin regions. If deeper, the meaning would possibly be the same as the generic structure codes.

In the absence of a clear use case for these codes, it is also interesting to see what use has been made of them in the logic-based definitions. Many of those that are used (as of 20060131) also have corresponding codes that do not contain the surface region designation, and this creates inconsistency in modeling, with some using x structure and others using x surface region. Buttock and shoulder are the two most egregious examples.

Therefore, for the 20060731 release, all surface region codes will be retired as ambiguous, with MAYBE-A references to their corresponding codes that are clearly not immaterial, including x structure, entire x, and skin of x. Where the x structure codes do not currently exist, they will be created, without the surface region phrase, which is ambiguous.

6.2.3.6.1.2.1.2 Joints – Joint regions

In many diseases and procedures, reference is made to areas of the body that may ambiguously imply either a joint or a region surrounding the joint. The main words that may ambiguously refer to either a joint or a region are:

- Ankle = Ankle joint structure [70258002] Ankle region structure [344001]
- Knee = Knee joint structure [49076000] Knee region structure [72696002]
- Hip = Hip joint structure [24136001] Hip region structure [29836001]
- Wrist = Wrist joint structure [74670003] Wrist region structure [8205005]
- Elbow = Elbow joint structure [16953009] Elbow region structure [127949000]
- Shoulder = Shoulder joint structure [85537004] Shoulder region structure [16982005]

Bone structure of shoulder girdle [272691005]: This code is used to define procedures and diseases affecting some bone tissue of the shoulder region. It is therefore **not** a kind of bone (organ), but it is a kind of bone structure, and is part of the shoulder region. Any part of the proximal humerus, scapula or clavicle is included.

Intertarsal joint structure (synonym: "tarsal joint") [27949001]: This is a structure that is part of a group of joints forming articulations between the seven bones of the tarsus. The talocalcaneonavicular joint [27162001] is the articulation between the talus and the other bones of the tarsus, and is thus assumed to be what is meant by the rarely-used term "talotarsal joint". The subtalar joint [127863007] is the same as the talocalcaneal joint. Dislocations of the subtalar joint will ordinarily also involve the talonavicular joint [127864001]. The subtalar and talonavicular joints taken together constitute the talocalcaneonavicular joint.

6.2.3.6.1.2.1.3 Orbital region

I Orbital region structure I is a synonym for I eye region structure I which subsumes bony orbit, eye and ocular adnexa.

6.2.3.6.1.2.1.4 Limbs

The meanings of the words "arm" and "leg" are prone to misinterpretation and ambiguity. In general English usage, "arm" can mean the upper limb, but it can also mean the upper arm, i.e. the part of the upper arm between the shoulder and elbow. "Leg" can mean the lower limb, but it can also mean the lower leg, i.e. the part of the lower limb between the knee and ankle.

In all hierarchies, including disorders, procedures, anatomy, and others, the fully specified name should not rely on the word "arm" or the word "leg" alone to designate the anatomy being referenced.

External sources, such as the WHO Classifications, may have conventions for interpreting the meaning of phrases that contain the words "arm" and "leg", and those conventions should be followed to determine the meanings of ICD rubrics in mapping and other actions that rely on the meanings from those external sources. The ICD conventions are sometimes at variance with common clinical parlance and will not necessarily match exactly the terms in SNOMED CT. For example, the rubric "injury of leg" in ICD may mean "injury of lower leg" in SNOMED CT.

6.2.3.6.1.2.1.5 Shoulder region – Upper limb; Hip region – Lower limb

The shoulder region is part of the upper extremity, and the hip region is part of the lower limb. This follows the general pattern used in the International Classification of Diseases 9th Edition (ICD-9) where one finds upper limb including shoulder and lower limb including hip. It also follows the rigorous ontological views of the FMA, in which the upper limb consists of the free upper limb and the pectoral girdle (of which the shoulder region is part), and the lower limb consists of the free lower limb and pelvic girdle (of which the hip region is part). A code has been added for free lower limb, i.e. the lower limb not including the pelvic girdle. One could in the future be added for free upper limb, i.e. upper limb not including pectoral girdle.

6.2.3.6.1.2.1.6 Axilla – Upper limb – Trunk

The axilla is bounded by the upper limb laterally and the thorax medially; therefore from one perspective it is not strictly part of either the upper limb or the trunk. The alternative view is that it is both. Axilla structure is currently defined in *SNOMED* as being both an upper limb structure and a thoracic structure.

6.2.3.6.1.2.1.7 Lower limb – Lower Leg – Leg – Foot

The lower limb (syn: lower extremity) includes the foot, but the lower leg (syn: leg) does not. Stedmans definition of lower leg is "The segment of the inferior limb between the knee and the ankle". "Leg" is used in ICD classifications to mean "lower leg". Common usage in English makes "leg" a synonym of "lower extremity". In order to avoid confusion, FSN's should always specify "lower leg" or "lower extremity". When an FSN (for a procedure or clinical finding code) has only the word "leg" with no other wording in the FSN that would allow determination of which meaning is intended, the code is ambiguous and should be retired.

6.2.3.6.1.2.1.8 Mouth region – Oral region of face – Teeth – Tongue – Larynx

6.2.3.6.1.2.1.8.1 Meaning of the word "mouth"

There are several different meanings of the word mouth. These include mouth region, oral region of face, and rima oris.

6.2.3.6.1.2.1.8.1.1 Mouth region

The mouth region includes structures surrounding the oral cavity as well as structures of the oral region of the face. Most disorders that have a *finding-site* of mouth should use mouth region.

6.2.3.6.1.2.1.8.1.2 Oral region of face (labial part of mouth)

The oral region of the face includes the skin and subcutaneous tissues of the lips and perioral region, plus the orbicularis oris muscle, and any vessels and nerves in these structures.

6.2.3.6.1.2.1.8.1.3 Rima oris

The rim of the opening bounded by the lips is called the rima oris.

6.2.3.6.1.2.1.8.2 Teeth – Maxilla – Mandible

Even though teeth are supported by the maxillary or mandibular bone, they are *not* "part-of" the maxilla [70925003] or mandible [91609006]. Teeth *are* part of upper jaw [4335006] and lower jaw [48077000].

6.2.3.6.1.2.1.8.3 Root of tongue

Prior versions had different codes for the base and root of the tongue. We found no reproducible distinction, and have retired base of tongue [7283002] as a duplicate of root of tongue. The four regional parts of the tongue are the ventrum, dorsum, root and body. The root of the tongue is the posterior third, the dorsal surface of which forms the anterior wall of the oropharynx. The root of the tongue rests on the floor of the mouth. The nerves and vessels that supply the intrinsic muscles of the tongue traverse the root of the tongue.

6.2.3.6.1.2.1.8.4 Inferior surface of tongue

Even though *SNOMED* 2 and *SNOMED* 3 had separate codes for inferior surface of tongue and ventral surface of tongue, we regard them as synonyms. There is no ventral surface of the posterior third of the tongue, so the ventral surface of the anterior two thirds is the same as the ventral surface, which is the inferior surface.

6.2.3.6.1.2.1.8.5 Larynx – Inlet of larynx – Interarytenoid fold – Hypopharyngeal aspect of interarytenoid fold

The interarytenoid fold forms part of the inlet of the larynx. The fold has two surfaces, one forming part of the wall of the supraglottic larynx, the other forming part of the wall of the hypopharynx (the "food tube" behind the larynx, leading to the esophagus). Is the "hypopharyngeal aspect of the interarytenoid fold [102295003]" a part of the hypopharynx, the larynx, or both? A tumor of this site should be categorized as a tumor of the hypopharynx, and not as a tumor of the larynx, but the interarytenoid fold [105585004] is considered part of the larynx. Given these two facts, we do ***not*** give a part-of *Relationship* between the hypopharyngeal aspect of the interarytenoid fold and the interarytenoid fold. This emphasizes the fact that we determine how to model anatomical entities based on the way that model causes disorders and procedures to be organized, ***not*** based on a simple reading of the term names.

6.2.3.6.1.2.1.9 Abdominal regions

The named regions of the abdomen are by tradition divided horizontally by the transpyloric plane and the interspinous plane, and vertically by the midclavicular plane. The lateral regions are therefore bounded above by a plane that is inferior to the ribs. In contrast, the flank is the lateral region of the abdomen bounded above by the ribs. Thus some parts of the hypochondriac regions, which are superior to the transpyloric plane but inferior to the ribs, would be considered also part of the flank. The hypogastric region is also sometimes called the pubic region.

6.2.3.6.1.2.2 Skin or skin-associated mucosa

This is an example of a body structure that is used to group and aggregate related terms. The term I Structure of skin and/or skin-associated mucous membrane (body structure) I intentionally employs disjunction (inclusive 'or'). Its referents include structures in the layers deeper than the surface epithelium, but exclude any non-skin-associated mucosal epithelium, such as bronchial, gastrointestinal, and genitourinary sites of squamous cell neoplasms. The “skin and/or surface epithelium” concept was created to represent the sites of these neoplasms.

Skin and/or skin-associated mucosa is intended for use in dermatology. It is not intended to subsume all mucosal structures, which are under I Mucous membrane structure (body structure) I. For the meaning of “diseases of the skin”, refer to the draft of ICD-11: “Diseases of the skin incorporate conditions affecting the epidermis, its appendages (hair, hair follicle, sebaceous glands, apocrine sweat gland apparatus, eccrine sweat gland apparatus and nails) and associated mucous membranes (conjunctival, oral and genital), the dermis, the cutaneous vasculature and the subcutaneous tissue (subcutis).”

6.2.3.6.1.2.3 Skin regions – Skin of <named body part>

Since the phrase | skin of finger | can mean "some or all of the skin of finger" (if interpreted as a structure rather than an entire in the SEP model), we could use "*is-a*" to represent the *Relationship* between | skin of finger | and | skin of hand |. Thus | skin of finger | *is-a* | skin of hand |, *is-a* | Skin structure of upper extremity |, *is-a* "skin region". We have refrained from adding the word "region" to all of these names, since it could be confusing without a clear distinction between the entire region and some subregion.

6.2.3.6.1.2.3.1 Scalp

Formal definitions of scalp include layers beneath the skin. Therefore we make a distinction between the scalp and the skin of the scalp.

6.2.3.6.1.2.4 Organs – Organ system subdivisions

The FMA notion of body organ is also used. Organs include individual bones, joints, muscles, arteries, veins, lymph vessels, nerves, etc. Codes with a meaning that includes groups of such organs are frequently listed in *SNOMED*. In most cases, these have been interpreted to be entities in the subsumption hierarchy (*is-a* hierarchy) of the particular organ type, that is, they are kinds of organ. When we also need a *concept* that means the collection of organs (rather than an organ in the collection), we have created another entity (code) that is a kind of organ system subdivision. But many such collections don't yet have such a corresponding organ system subdivision code. The default has been to interpret codes as denoting organs rather than organ system subdivisions.

Examples:

Organ	Organ system subdivision
Vertebra (bone of vertebral column)	Spine (subdivision of skeletal system)
Cervical vertebra	Cervical spine (subdivision of spine)
Third cervical vertebra	
Bone of skull	Skull (subdivision of skeletal system)
Bone of thoracic cage	Thoracic cage (subdivision of skeletal system)
Rib	
Third rib	
Right third rib	
Quadriceps femoris muscle	
Quadriceps femoris muscle, left	
Vastus medialis muscle	

6.2.3.6.1.2.5 Cell, Tissue, Organ

In general, organs are made up of tissue, and tissue is made up of cells. However, a cell is not necessarily part of tissue, and tissue is not necessarily part of a named organ.

6.2.3.6.1.2.6 Body systems and tracts

Many terms are used imprecisely in clinical practice and in medical publications to refer to body systems or tracts, and ambiguities frequently arise with many of these terms. In particular, the terms for the gastrointestinal, alimentary, genitourinary, genital, urinary, respiratory, biliary, lymphatic, lymphoid, immune, reticuloendothelial, andhematopoietic systems of the body may have multiple interpretations. We have (arbitrarily) made the following definitions and distinctions in order to achieve internal consistency of the terminology. We recognize that it may not be possible to get universal consensus regarding the names that should be used for each of these codes. The goal is to be consistent and clear in defining the meaning of each code, and to allow users and system designers to present the terms that best reflect these meanings in their own implementation contexts.

6.2.3.6.1.2.6.1 Urinary system – Urinary tract – Genitourinary system – Genitourinary tract

Urinary system:

The urinary system includes the organs involved in the formation and secretion of urine, including the kidney, ureters, bladder, and urethra. *Urinary system* includes the prostatic urethra (since it is a male urinary outflow structure) but excludes other parts of the prostate (and the prostate as a whole) and also excludes the seminal vesicles (see lower urinary tract). Unless clearly specified otherwise, *urinary tract* and *urinary system* are considered synonyms, and terms that include the phrases are interchangeable. For example, *computed tomography of urinary tract* is the same as *computed tomography of urinary system*. Broad categories that are intended to exclude the kidney should specifically use the term *urinary tract proper* (see next). Examples include operation on urinary tract proper and disease of urinary tract proper.

Urinary tract proper:

The urinary tract proper includes the organs involved in the secretion of urine but excludes the kidney itself; it includes the renal pelvis, ureters, bladder, and urethra. It is a fairly subtle distinction from urinary system, but may be useful for categorizing disorders affecting the flow of urine (as opposed to its formation), such as urinary tract obstruction, and as the site of tubular structures lined with urothelium. Because *urinary tract* is ordinarily used as a synonym of *urinary system*, we have added the word proper to distinguish this more specific meaning (which excludes the non-collecting parts of the kidney) from the broader meaning.

Upper urinary tract:

The upper urinary tract is the urinary system above the junction of ureter with the bladder, and consists of the kidneys and ureters. Since upper urinary tract infections include kidney infection, the upper urinary tract must include the kidney. The FSN of this *concept* is kidney and/or ureter structure (body structure), and it has a synonym of *upper urinary system*.

Upper urinary tract proper:

The upper urinary tract proper is the part of the urinary tract proper above the junction of the ureter with the bladder. It consists of the renal collecting system and the ureter.

Lower urinary tract:

The lower urinary tract is the urinary system below the junction of the ureter with the bladder. It consists of the bladder and urethra. Lower urinary tract and lower urinary system are the same. The male and female specific components are located under male urinary outflow structure and female urinary outflow structure, respectively.

Genitourinary system:

The genitourinary system includes the entire urinary system as well as the genital system. We consider genitourinary tract to be synonymous with genitourinary system.

Genital system:

The genital system is comprised of both internal genital organs and external genitalia. Genital tract is defined only for the female: The female genital tract is comprised of ovaries, fallopian tubes, uterus, vagina and vulva.

6.2.3.6.1.2.6.2 Digestive system – Digestive tract – Alimentary tract – Gastrointestinal tract

Digestive tract is the same as the alimentary tract, and includes the entire passage for food through the body, including mouth, oral cavity (both vestibule of mouth and cavitas oris propria), oropharynx, esophagus, stomach, duodenum, jejunum, ileum, colon, rectum, and anal canal.

Digestive system: includes the digestive tract as well as the associated organs of digestion, including tongue, teeth, salivary glands, liver, exocrine pancreas, gallbladder and biliary tract.

Gastrointestinal tract: There are two meanings in common usage of this term. The first would more properly be named the esophago-gastrointestinal tract, since the esophagus is ordinarily included. Endoscopists frequently adopt this meaning, even though it is contrary to some dictionary definitions, which exclude the esophagus. Including the esophagus also does not follow a strict lexical interpretation.

Upper gastrointestinal tract: By convention in describing upper GI bleeding and upper GI radiographic and endoscopic procedures, this includes the esophagus, stomach and duodenum. It is part of the gastrointestinal tract that includes the esophagus, but obviously not part of the more restricted stomach-intestine entity.

Lower gastrointestinal tract: By common convention in describing lower GI bleeding, lower GI radiographic and endoscopic procedures, and lower GI output from ileostomies and colostomies, this includes the jejunum, ileum, cecum, colon, rectum and anal canal. The ligament of Treitz may be used as the dividing line between upper and lower GI tract (and the dividing line between duodenum and jejunum). See J Vasc Interv Radiol 9:747 for an example that shows inclusion of the jejunum and below as part of the lower GI tract. Also, since the upper GI tract is said to end at the duodenum-jejunum boundary, and there is no code meaning middle GI tract, the jejunum can be inferred to be in the lower GI tract.

6.2.3.6.1.2.6.3 Biliary tract – Liver

Biliary tract: includes the gallbladder and the intrahepatic and extrahepatic bile ducts, and the common bile duct. It does not include the liver itself. We use "biliary system" as a synonym for biliary tract. (Another code might be created to mean an entity that includes the entire liver with the biliary tract, but we do not at present perceive a need for it).

6.2.3.6.1.2.6.4 Lymphoid – Lymphatic – Immune – Mononuclear phagocytic – hematologic – hematopoietic – dendritic cell systems

Lymphatic system (Lymphatic system structure [89890002]): is conceptually the set of structures through which lymph flows. It includes the lymph nodes (lymph node structure 59441001) and lymphatics (structure of lymphatic vessel 83555006). It supports the categorization of findings, disorders and procedures that relate to the flow of lymph.

Lymphoid system (Lymphoid system structure [122490001]): is conceptually the set of structures made up of aggregates of lymphoid cells. It includes lymphoid aggregates of the intestine, marrow, liver, and other locations, and the lymph nodes, spleen, and thymus, and tonsils & adeonoids. It excludes the lymph vessels. It supports categorization of lymphomas.

Immune system (Immune system structure [116003000]): includes all of the lymphoid system, as well as the mononuclear phagocytic system. There are also essential components of the immune system that are cellular and sub-cellular and are involved in cellular and humoral immunity.

Mononuclear phagocytic system (Mononuclear phagocyte system structure [127908000]): a collection of true macrophages, distributed widely in the body (splenic sinusoids, liver Kuppfer cells, pulmonary alveolar macrophages, osteoclasts, macrophages in serous membranes, and microgliocytes). It is part of the immune system.

Dendritic cell system (Dendritic cell system structure [127909008]): a collection of antigen-presenting cells, including epidermal Langerhans cells, dendritic reticulum cells, and interdigitating cells. Class I histiocytoses (Langerhans cell histiocytosis) are disorders of the dendritic cell system.

Reticuloendothelial system (Reticuloendothelial system structure [6013009]): an outdated term, includes the true macrophages (the mononuclear phagocytic system) and also additional endothelial cells that line lymphoid sinusoids and hematopoietic tissues.

Hematologic system (hematological system structure [414387006]) : includes the bone marrow, the lymphoid system, the hematopoietic system, and the terminal cells of all lineages of the hematopoietic system (red cells, white cells, platelets, histiocytes, plasma cells, etc). This means that disorders of the hematologic system do *not* necessarily include disorders of the hemostatic system, even though bleeding and thrombosis are usually categorized as hematologic.

Hematopoietic system (hematopoietic system structure [57171008]): includes the structures and cells responsible for erythropoiesis, granulocytopoiesis, monocytopoiesis, thrombocytopoiesis, and lymphopoiesis. Hematopoietic should be differentiated from hematologic, since the terminal cells of each lineage (the erythrocyte, segmented neutrophil, monocyte, histiocyte, platelet, mature T- and B-cells, plasma cells, etc.). are no longer strictly hematopoietic.

6.2.3.6.1.2.6.5 Circulatory system – systemic, central, peripheral, cerebrovascular, intracranial, extracranial

6.2.3.6.1.2.6.5.1 Systemic vs pulmonary circulation

The systemic circulatory system is the combined arterial and venous circulation that begins where blood leaves the left ventricle and ends where blood enters the right atrium. It excludes the coronary circulation.

The pulmonary circulation is the combined arterial and venous circulation that begins where blood leaves the right ventricle and ends where blood enters the left atrium.

The heart chambers are also considered part of the circulatory system.

6.2.3.6.1.2.6.5.2 Central vs peripheral vs cerebrovascular system

The term central vascular is not in common use. In fact, the term does not appear in *SNOMED* at all. However, the term peripheral vascular is very common, and therefore it requires a definition that (by default) sets the boundary between central and peripheral vascular systems.

The simplest definition of peripheral vascular system is that it is the vascular system that is not central; and then the central vascular system includes the pulmonary circulation, coronary circulation, cerebrovascular system, thoracic aorta, superior vena cava, inferior vena cava, and mediastinal blood vessels.

Peripheral vascular disease is often distinguished from cerebrovascular disease and coronary artery disease. These are the three major categories of diseases caused by problems in vascular circulation in general, and atherosclerosis in particular. As a result of this clinical distinction, the cerebrovascular system is excluded from the peripheral vascular system.

Cerebrovascular is commonly defined in two ways: as either the blood vessels *in* the brain, or the blood vessels that *supply* the brain (including those within the brain). Because cerebrovascular disease includes extra-cranial occlusions of the vertebral and carotid arteries, we define the cerebrovascular system as those vessels involved in the supply and drainage of blood to the brain. Convention does, however, tend to exclude the innominate artery - which gives rise to the left common carotid and the arch of the aorta which gives rise to the right common carotid. Convention also excludes the subclavian arteries which give rise to the vertebral arteries.

6.2.3.6.1.2.6.5.3 Intracranial vs extracranial vascular system

Some vascular trees are located wholly within the cranial cavity, but some (internal carotid; vertebral) cross the boundary between extra- and intra-cranial. Intracranial segments of such vascular trees must be individually identified as such, and the entire vascular tree must not be categorized as either extra- or intra-cranial. See tree-structured organs below. These are regional parts of venous or arterial tree organs.

6.2.3.6.1.2.6.6 Blood – Cardiovascular system – Hematopoietic system

The blood is not necessarily part of the cardiovascular system, nor is it necessarily part of the hematopoietic system. If it were, then leukemia would be a cardiovascular disorder, and septicemia would be a hematopoietic disorder. Since these inferences violate our clinical expectations, we make the underlying model of anatomical *Relationships* support the kind of *Relationships* that are correct and expected. Thus blood is a body fluid, not strictly part of either the hematopoietic or cardiovascular systems.

6.2.3.6.1.2.6.7 Endocrine system

The endocrine system structurally is composed of the endocrine pancreas, pineal body, paraganglia, paraaortic bodies, parathyroid glands, endocrine ovary, endocrine testis, adrenal glands, pituitary gland, thyroid gland, the juxtaglomerular apparatus, and some diffuse neuroendocrine structures.

Certain parts of the thymus have been shown to be capable of producing endocrine hormones, but the thymus itself is not categorized as part of the endocrine system.

6.2.3.6.1.2.6.8 Hematopoietic system – Blood – Spleen – Lymph nodes – Thymus

Hematopoietic is used to mean the not-as-yet-mature cellular elements that eventually form the cellular components of blood. The blood itself cannot be strictly part of the hematopoietic system, since this would cause all components of blood to be part of the hematopoietic system (including components like albumin, clearly not 'hematopoietic'). Leukocytes, red cells and platelets are the *result* of hematopoiesis, but they are not blood-forming themselves, in the strict sense we are using (otherwise leukocytosis would become a disorder of hematopoiesis, whereas it can arise simply from a demargination of white cells following stress). We have created a code named "cellular components of blood"; note that platelets are not actually cells, but are 'cellular components'. Likewise, for spleen, lymph nodes and thymus, we have created "hematopoietic cells of spleen" etc. to indicate that they are part of the hematopoietic system. This enables differentiation of disorders of the hematopoietic system from infectious, traumatic and other disorders, and prevents incorrect autoclassification.

6.2.3.6.1.2.6.9 Nervous system

The nervous system is divided into central and peripheral subdivisions. The central nervous system, sometimes also called the neuraxis, consists of the brain and spinal cord. The pyramidal system is a subdivision of the central nervous system; the extrapyramidal system is part of the brain. The peripheral nervous system includes all neural structures outside the central nervous system. The nervous system is also subdivided into autonomic, somatic and enteric subdivisions. The autonomic system is further divided into sympathetic and parasympathetic subdivisions. The autonomic system is not entirely a part of the peripheral nervous system; but all autonomic nerves are peripheral (see the section on tree-structured organs and the meaning of nerve.)

6.2.3.6.1.2.6.10 Respiratory tract – Respiratory system – Upper aerodigestive tract

We have chosen to have I respiratory tract I mean the same as the Nomina Anatomica term "apparatus respiratorius", which includes the structures through which air passes from the nares to the alveoli. The oral cavity, however, is not included (even though functionally one might expect it to be). The phrase respiratory system is sometimes regarded as a synonym of I respiratory tract I, but we have given them separate meanings. Respiratory system does not, however, mean the global respiratory system that might include the CNS components of breathing. Pleura are part of the lower respiratory system, but not a part of the lower respiratory tract (see below).

Upper aerodigestive tract is a phrase that may have several meanings. The *SNOMED* code for "upper aerodigestive tract" has adopted the meaning defined by Muir and Weiland in "Upper aerodigestive tract cancers", Cancer 1995 Jan 1;75(1 Suppl):147-53, which states: "Cancers of the upper aerodigestive tract constitute approximately 4% of all malignancies. These include cancer of the lip, tongue, major salivary glands, gums and adjacent oral cavity tissues, floor of the mouth, tonsils, oropharynx, nasopharynx, hypopharynx and other oral regions, nasal cavity, accessory sinuses, middle ear, and larynx." This definition matches the tumors included in the CAP Cancer Checklist for upper aerodigestive tumors. Some publications include the esophagus, or at least the cervical esophagus, when referring to the upper aerodigestive tract, but we have decided to exclude esophagus.

Aerodigestive tract is a phrase with more variation in meaning than "upper aerodigestive tract." There is currently no code for this term, because of the variable meanings, and limited reference to "aerodigestive tract" in the literature. It certainly would include the upper aerodigestive tract plus the tracheobronchial tree, lungs, and esophagus, but the few literature citations using the term do not appear to intend it to include any of the digestive tract except the esophagus, in spite of the strict lexical interpretation that might lead one to expect inclusion of the entire digestive tract. The lower aerodigestive tract would be the combination of the esophagus and the lower respiratory tract. There is currently no code for this term.

Upper respiratory tract is that part of the respiratory tract from the larynx up, and includes the nasal cavity, paranasal sinuses, nasopharynx, oropharynx and larynx.

Lower respiratory tract begins below the larynx, and includes the tracheobronchial tree (from the trachea through the terminal bronchioles) as well as the lungs, including the alveolar respiratory tract or pulmonary region (which extends from the respiratory bronchioles to the alveoli).

Lower respiratory *system* includes the lower respiratory *tract* and the pleura.

6.2.3.6.1.2.6.11 Skeletal system – Bony skeleton/bone structure – Vertebral column

The skeletal system (systema skeletal in Nomina Anatomica) includes both bones and cartilages of the body. The bony skeleton includes just the bones. The vertebral column is part of the skeletal system, and includes the intervertebral discs (fibrocartilage). Individual vertebrae are part of the bony skeleton. The spinal region (also sometimes called the spine) includes the spine proper (same as vertebral column = vertebra and intervertebral discs) as well as the contents of the spinal canal, and also paraspinal ligaments, muscles, soft tissues, and skin.

6.2.3.6.1.2.6.12 Soft tissue

There are at least three different use cases (and therefore at least three different meanings) for the phrase soft tissue:

1. A category for tumors. So-called soft tissues give rise to similar types of neoplasms that appear to be of mesenchymal stem cell origin, generally termed the soft tissue neoplasms, and this appears to account for the inclusions and exclusions of the category. Non-neoplastic masses arising in the same tissues are included in the most recent WHO classification of soft tissue tumors. For tumors, soft tissue is defined as non-epithelial extraskeletal tissue of the body exclusive of the reticuloendothelial system, glia and supporting tissue of various mesenchymal organs.[16] Other explicit inclusions are: fibrous tissue, fascia, ligaments, tendons, tendon sheaths, synovia, bursas, skeletal muscle, smooth muscle, fatty tissue, adipose tissue, blood vessels, lymph vessels, peripheral nerves, sympathetic and parasympathetic nerves and ganglia. Subcutaneous tissue is included. Skin is excluded. Skeletal cartilage is excluded, along with pleura, pericardium and peritoneum, the central nervous system, endocrine glands, and viscera.

   FSN: Extraskeletal non-epithelial non-reticuloendothelial non-glial soft tissue (body structure)

2. A category for sites of non-bone disorders and injuries of the limbs, head, neck, and body wall. In this case, skeletal cartilage is included as a soft tissue. Skin and lymph nodes are not included, but otherwise all non-bone structures of the limbs are included. Subcutaneous tissue and fat are included. For the head, neck and torso, this category excludes reticuloendothelial system, central nervous system, endocrine glands, viscera and supporting tissues.

   FSN: Musculoskeletal and/or neurovascular soft tissue excluding central nervous system and visceral soft tissues (body structure)

3. A category for structures identified in images. In this case, soft tissues include everything except for mineralized bone tissue and teeth.

Note:

As of the 20080131 release, these three meanings have not been incorporated into the terminology, and therefore *concepts* using the phrase soft tissue require significant changes.

6.2.3.6.1.2.7 Tendons – Muscles

Is a muscle an entire functional unit, including attachments to the skeletal system, or merely the contractile part of this unit? Either choice could be made; clinically we think of the muscle as the contractile part only. The FMA definition of *organ* implies that tendons should be considered part of their corresponding muscles, rather than organs in their own right.

We have decided to model tendon structures as *subtypes* of their muscle structures. Thus the I Structure of achilles tendon I is a triceps surae (gastrocnemius and soleus) muscle structure. This causes the classifier to make a rupture of the Achilles tendon a kind of disorder of the triceps surae (gastrocnemius and soleus) muscle. Functionally this makes sense, even though to some users it may violate the natural sense of "muscle" as contractile tissue only.

6.2.3.6.1.2.7.1 Muscle functions

When modeling muscle categories according to their functions, assume they mean the function of the “entire muscle” unless otherwise stated. This follows the general heuristic that *CTV3* codes meant “entire entity” unless otherwise stated, while *SNOMED* codes meant “entity structure” unless otherwise stated. Most of the muscle functional groupings came from *CTV3*.

[16] Enzinger FM, Weiss SW: Soft-Tissue Tumors. St. Louis: Mosby, 1995.

6.2.3.6.1.2.8 Bones – Bone Tissue

6.2.3.6.1.2.8.1 The word "bone"

In ordinary usage, the word bone conflates the meanings bone organ and bone tissue onto the single word.

6.2.3.6.1.2.8.2 Definitions

6.2.3.6.1.2.8.2.1 Bone (tissue) structure:

a quantity of regular connective tissue which consists of osteocytes and related cells, the intercellular matrix of which is ossified. (or any part thereof)

6.2.3.6.1.2.8.2.2 Bone structure:

a bone organ or any part thereof.

6.2.3.6.1.2.8.2.3 Bone organ:

an organ with cavitated organ parts, which primarily consists of compact (cortical) and cancellous bone, which surround bone marrow cavities; other parts include periosteum, endosteum, (and, according to FMA, articular cartilage.)

6.2.3.6.1.2.8.3 Non-ossified parts of bone

Bone organs are composed primarily of bone tissue, but there are some non-ossified parts. In particular, periosteum is clearly a part of a bone organ, but is not ossified tissue. (Articular cartilage is in dispute: *SNOMED* doesn't currently model articular cartilage as a part of bone, although FMA does.)

6.2.3.6.1.2.8.4 Bone marrow and marrow cavity

Because bone marrow is contained within a marrow cavity, it is in one sense included within a whole bone. If you have a whole femur, you also have its bone marrow - at least in living subjects. But in another sense the marrow is not strictly part of the bone organ itself. In skeletons, the whole femur has a cavity where the marrow was, but there is no marrow. Clinical reasoning generally does not include marrow disorders under bone disorders, nor marrow procedures under bone procedures.

Examples:

Bone marrow disorders are not musculoskeletal disorders, but bone disorders are.

Bone marrow transplants are not considered types of bone transplant.

Osteomyelitis is not the same as osteitis.

The (empty) marrow cavity is part of the bone organ, and the marrow is contained in the marrow cavity but is not part of the bone organ. Therefore Bone marrow structure (body structure) is not a *subtype* of Bone structure (body structure).

All bone marrow was intended to be the E entire counterpart to the S of bone marrow structure, and so it also is **not** a *subtype* of Bone structure (body structure).

6.2.3.6.1.2.8.5 Structure of (named bone) versus Bone structure of (named bone)

To differentiate marrow, vessels, nerves and periosteum from the actual hard tissue of bones, we differentiate structure of tibia from bone structure of tibia. The bone marrow and other soft tissues of the tibia can then be categorized separately from the hard tissues. Bone marrow diseases are not considered musculoskeletal diseases, so bone marrow structures should not be placed in the bone (tissue) structure hierarchy.

6.2.3.6.1.2.8.6 Long bone – Short bone

ICD-9 and 10 do not use the standard anatomical definition of a long bone. For example, see ICD-9-CM 213.4 and 213.5, in which benign neoplasms of the "long bones" are distinguished from benign neoplasms of the "short bones," includes the bones of the hand as short bones. Anatomical definitions of long bone cite the proportional relationship between length and width (length >> width), and it is very clear that metacarpals, metatarsals, and phalanges are included in the anatomical definition of long bone. In order to accommodate these differences between anatomical definitions and the classifications, in SNOMED CT there are anatomical groupers that correspond to the ICD groupings, so that "scapula, humerus, radius or ulna" and "long bone of thigh or lower leg" can be used as the sites for grouper concepts that match various ICD definitions and groupings.

6.2.3.6.1.2.8.7 Sternum – Manubrium, Body, Xiphoid

The sternum is considered a bone organ. The manubrium, body and xiphoid are parts of the sternum classed as zones in FMA.

6.2.3.6.1.2.8.8 Nasal turbinates – Nasal conchae

We have differentiated between the bone underlying the nasal turbinates (118648008 , 122491002 , 122492009 , and 122493004) and the turbinates themselves. The turbinates themselves (6553002 , 60962000 , 65289004 and 33415007) include both bone and overlying mucous membranes and other tissues. The inferior nasal turbinate bone is a facial bone (and skull bone) in its own right. However, parts of the ethmoid bone form the middle, superior and supreme nasal conchae. This means that the bones of the middle, superior and supreme turbinates are not bone organs.

6.2.3.6.1.2.9 Tree structured organs

Arteries, veins, nerves, and the bronchi form tree-like structures that distribute across multiple regions. Because of their extent and their interdigitation with other structures, they require some slightly different thinking and modeling. FMA deals in a very consistent way with organs that are structured as trees, which can either have a cavity or be solid. In the first category (organ with organ cavity) it has a *subtype* hollow tree organ. The hollow tree organs are:

- tracheobronchial tree
- biliary tree
- vascular trees
  - arterial trees
    - the systemic arterial tree
    - the pulmonary arterial tree
  - venous trees
    - the systemic venous trees (superior, inferior, and 4 cardiac trees)
    - the pulmonary venous trees (there are 4: sup L, inf L, sup R, inf R)
    - the portal venous tree
- lymphatic trees (there are two: the right lymphatic duct tree, and the thoracic duct tree)

Among the solid organs, there is one category that is tree-structured. The tree-structured solid organs include:

- neural tree organs
- cranial nerve trees
- spinal nerve trees
- spinal accessory nerve tree (strictly neither a cranial nerve nor a spinal nerve per se)
- peripheral nerve trees
- autonomic nerve trees
- cranial nerve-tract complex trees

Having accepted the idea of tree-structured organs, the next task is to decide what words to use in a systematic way to refer to them and their various parts.

6.2.3.6.1.2.9.1 The word "artery"

There are three potential meanings for the word "artery":

1. an arterial trunk (a single tube)

2. an arterial tree organ, and
3. an arterial trunk plus all its branches.

In modeling *SNOMED* meanings that refer to arteries, it is necessary to decide on a case-by-case basis which of these meanings is intended. However, some general guidance can be given that meaning (1) above is most common in clinical use.

In most cases it is easy to dispense with (2) because there are only two actual complete arterial tree organs (the systemic arterial tree arising at the aortic valve, and the pulmonary arterial tree arising at the pulmonary valve), and these are readily named as arterial tree organs, and seldom referred to by individual disorders or procedures. The remaining difficulties arise in differentiating when a trunk is intended, and when an entire tree (trunk plus branches) is intended.

If one examines clinical usage of the word "artery" for injuries and operations, the meaning is clearly a single tube. the trunk of the named artery, or trunk of the named arterial branch. A puncture wound of the femoral artery affects the femoral arterial trunk. A grafting into the popliteal artery likewise is done into the popliteal arterial trunk. Even occlusions of an artery are located by naming the trunk where the occlusion occurs; and even though the downstream distribution may be affected, collateral circulation often mitigates the effects, so it would be incorrect to interpret artery to mean the entire subtree in any of these clinical usages.

NOTE: This clinical usage is at variance with the definitions of the FMA, which defines artery as a subdivision of an arterial tree (organ) which consists of branching sets of tubes (arterial trunks) that form a tree; together with other arterial trees (organ parts), it constitutes an arterial tree (organ). It would be correct to say that FMA regards artery as an arterial tree organ part which is not just a trunk but also has branches, corresponding to item number 3 above.

6.2.3.6.1.2.9.1.1 Artery: trunk of artery vs arterial tree

6.2.3.6.1.2.9.1.1.1 Pulmonary artery – Artery of lung – Trunk of pulmonary artery

Trunk of pulmonary artery [45341000]: This is the main pulmonary artery, the "great vessel" coming off the right ventricle and splitting into right and left main pulmonary arteries. Some dictionaries make this synonymous with I pulmonary artery I.

Pulmonary artery within lung [128260003]: Any artery of the pulmonary circulation that is regionally within the lung, the boundary being defined by the hilum.

Pulmonary artery [81040000]: Any artery of the pulmonary circulation, i.e. artery(ies) conveying unoxygenated blood from the heart into the lungs, including the trunk, right and left branches of the pulmonary artery, which are within the mediastinum, and all their branches, which tend to occur at or past the hilum and are therefore regionally within the lung.

6.2.3.6.1.2.9.1.1.2 Common carotid artery – Artery of neck

The right common carotid artery usually arises from the brachiocephalic trunk behind the right sternoclavicular joint, and thus has no real thoracic portion. However, the left common carotid arises from the arch of the aorta and does have a short thoracic portion. Should the common carotid artery (not specifiying laterality) be an artery of neck, i.e. an artery that is part of the neck? Strictly speaking, it is not, because of the thoracic portion of the left common carotid. At present, however, the model of anatomy includes common carotid artery as an artery of the neck. This needs to be changed.

6.2.3.6.1.2.9.2 The word "vein"

There are three potential meanings for vein:

1. a venous trunk
2. a venous tree organ, and
3. a venous trunk plus all its branches.

In modeling *SNOMED* meanings that refer to veins, it is necessary to decide which of these meanings is intended. It is easy to dispense with (2) because there are only eleven venous tree organs, and these are readily named as such. The remaining difficulties arise in differentiating when a trunk is intended, and when a venous tree part (trunk plus branches) is intended.

As with the clinical usage of the word "artery", clinical usage of the word "vein" generally refers to the trunk and not the entire tree.

NOTE: This clinical usage is at variance with the interpretations of vein names given in the FMA, which defines vein as a subdivision of a venous tree (organ) which consists of branching sets of tubes (venous trunks) that form a tree; together with other venous trees (organ parts), it constitutes a venous tree (organ). Thus FMA regards vein as a venous tree organ part.

6.2.3.6.1.2.9.2.1 Vein: trunk of vein vs vein as a tree structure

Because trunks of veins, not venous trees, have been used to organize the vein hierarchy, there are implications for regional classes. For example, the internal jugular vein is a vein of the neck, but of course its entire venous tree extends into the head, and the internal jugular vein venous tree is not strictly part of the neck, even though the internal jugular vein venous trunk is strictly part of the neck.

Tributaries are also modeled as direct tributaries of the trunk. A tributary of a named vein is part of the venous tree of the named vein, but not part of the venous trunk of the named vein. Some veins that are part of the venous tree, and therefore might be regarded as indirect tributaries, are not modeled as direct tributaries of the trunk of the vein. Direct tributary is the intended meaning of tributary.

6.2.3.6.1.2.9.2.2 Vein and its tributaries

All codes with the name pattern "vein x and its tributaries" have been retired, with MAYBE-A links to "structure of vein x" and "entire vein x", because there was ambiguity about the meaning of these terms.

6.2.3.6.1.2.9.2.3 Pulmonary vein vs vein of lung

Pulmonary vein great vessel: There are four pulmonary veins that enter the left atrium, two on each side; these are what is intended by the name I pulmonary vein I. The pulmonary veins are "great vessels" (vessels that enter the heart). Common usage sometimes might result in people referring to any vein that is part of the lung as a I pulmonary vein I, but we have a separate code (see pulmonary venous structure) for this meaning.

Pulmonary venous structure [122972007]: This means any vein that drains the lung, and a synonym is vein of lung. Pulmonary veins are kinds of "vein of lung." But I Pulmonary vein I and "vein of lung" are not synonyms.

So far, there is no code for pulmonary vein within lung.

6.2.3.6.1.2.9.2.4 Retinal vein

There is no vein named retinal vein therefore vein of retina and retinal vein are the same.

6.2.3.6.1.2.9.3 The word "nerve"

The word "nerve" potentially has multiple meanings: nerve trunk, nerve organ segment (trunk plus branches), and entire neural tree organ. If we examine the FMA nerve *concepts*, the comment under neural tree organ identifies only two of these meanings:

> The term nerve is conventionally used as a homonym for two concepts: 1. an anatomically distinct nerve trunk (without branches) that is identified in a dissection (e.g. the structure that student identifies when a pin is placed in the trunk of the vagus nerve, for instance located on the arch of the aorta); 2. a larger anatomical entity which supports a related set of functions (e.g. all anatomical components of the vagus nerve that are necessary for it to execute its functions; for instance when a student is asked which nerve is responsible for slowing the heart his answer, the vagus nerve, includes the vagal nucleus, as well as the trunk and branches of the vagus). Neural tree designates the second concept in order to distingiush it from the first which is only a part (subdivision of) the vagal neural tree.

The third meaning is found in the definition of the FMA class labelled "nerve":

> Segment of neural tree organ which has as its parts a nerve trunk and its branches; together with other nerves of the same tree it constitutes a neural tree. Examples: chorda tympani, digastric branch of facial nerve, greater petrosal nerve, posterior cutaneous branch of posterior ramus of cervical nerve, superior lateral cutaneous nerve of arm.

To summarize, the FMA has defined three meanings for nerve:

1. a nerve trunk
2. the entire neural organ including nuclei, ganglia, roots, etc.
3. a nerve trunk plus all its branches (excluding nuclei, ganglia, and roots)

It can create significant confusion we recognize that nerve is commonly used as a homonym for all three meanings. The FMA assigns the third meaning as the one that they adopt for the class labelled nerve.

The trouble in this approach to resolving the problem of what "nerve" means is that when we call the first meaning "nerve trunk" and the second meaning "neural tree", it is difficult to decide how to refer to the third meaning.

One solution is to use the phrase "neural organ" for the second meaning, since it is not really just a tree structure (at least an above-ground tree: the ganglia arent in the trunk or the branches); then the phrase "nerve tree" can be used for the third meaning. This would give us the trio of nerve trunk, neural organ, and nerve tree. I think these phrases have better transparency than the trio of nerve trunk, nerve tree organ and nerve.

Unlike clinical usage for arteries and veins, the clinical usage of the word "nerve" does not reliably refer to one of the three possible meanings, but instead varies much more between the different interpretations, based on context. If one severs the facial nerve, the meaning refers to the trunk. But if one has facial nerve palsy, the meaning refers to the entire distribution of the nerve and the functions served by it.

6.2.3.6.1.2.9.3.1 Nerves – Entire nerve – Nerve and branches – Nerve tissue

There are several codes with the phrase x nerve and its branches which came from *CTV3*, and they are interpreted as meaning the entire nerve and its branches. Therefore, x nerve and its branches would be a duplicate of entire x nerve, when we interpret entire x nerve as being a neural tree organ. [17] For example, entire facial nerve is a neural tree organ, and so there is no need for an additional *concept* called facial nerve and its branches, which would mean the same thing. An entire cranial nerve is a neural tree organ, and structure of cranial nerve is that organ or any part (or branch) thereof. Branches of the cranial and spinal nerves are segments of the neural tree organs that they branch from. [18]

6.2.3.6.1.2.9.3.2 Nerve and its branches

We have retired all the codes named as "nerve x and its branches", with MAYBE-A links to structure of nerve x, and entire nerve x, because of the ambiguity of the terms. To specify the trunk of a nerve requires a specific term.

6.2.3.6.1.2.9.4 Eye structures

The meanings of "subchoroidal space" and "suprachoroidal space" are the same, and refer to a potential space between the choroid and sclera. The term "lamina subchoroidea of choroid" is from SNOMED 2 and had code T-XX320. But this layer is the same as the layer termed the "lamina suprachoroidea". According to Trans Am Ophthalmol Soc. 1993; 91:545-652, "Most of the earlier reports described expulsive hemorrhage starting as a subchoroidal hemorrhage that became large and expelled intraocular contents. The more recent

---

[17] A neural tree organ is defined in FMA as a nonparenchymatous organ which has as its parts an aggregate of neurons (nuclei or ganglia) and their axons which are grouped into fasciculi by connective tissue to form elongated, cable-like structures that are arranged into a tree. The cranial nerves and spinal nerves are considered to be neural tree organs.

[18] A nerve according to FMA is defined as a segment of a neural tree organ which has as its parts a nerve trunk and its branches; together with other nerves of the same tree it constitutes a neural tree.

reports have used the term "suprachoroidal hemorrhage." Both "subchoroidal" and "suprachoroidal" refer to the same potential anatomic space between the choroid and the sclera. In the recent literature, the terms "expulsive hemorrhage" and "subchoroidal hemorrhage" are being replaced by "massive suprachoroidal hemorrhage."

6.2.3.6.1.2.10 Cerebrum – Telencephalon – Supratentorial brain

Cerebrum may refer to the supratentorial brain, which is everything except the midbrain, medulla, pons, and cerebellum. In this interpretation, the telencephalon and diencephalon are in the cerebrum. On the other hand, cerebrum may refer only to the parts derived embryologically from the telencephalon, which are the cerebral hemispheres and the intercerebral commissure (corpus callosum and anterior commissure).

Supratentorial brain is a phrase sometimes used for categorizing tumors (ICD-9 codes 191.0-5 or 191.8-9), and for designating the location of swelling that can result in herniation. The telencephalon and diencephalon (including thalamus, geniculate bodies, pineal body, habenulae, and hypothalamus) are definitely supratentorial. Strictly speaking, the *upper* part of the midbrain (mesencephalon) also is supratentorial. However, the broad categories of ICD codes listed above would exclude any midbrain tumors from the list of supratentorial tumors, so we follow that pattern and exclude all midbrain structures from the supratentorial brain.

6.2.3.6.1.2.11 Regional lymph nodes of the lung

We have retained codes representing the nodes in traditional anatomy (lymph nodes categorized as: pulmonary, bronchopulmonary, tracheobronchial, tracheal, and esophageal) along with codes representing node groups used for clinical staging of lung cancer (lymph nodes categorized into 14 stations). Professional societies concerned with the clinical staging of lung cancer have developed at least three different nomenclatures for "stations" of lung-related lymph nodes. The ATS (Americal Thoracic Society) map, published in 1983, is given in *Am Rev Respir Dis 1983; 127:659-669*. A revised system adopted by the American Joint Committee on Cancer (AJCC) and the International Union against Cancer (UICC) in 1997 is given in *Chest 1997; 111:1718-1723*. Even though the numbering of the stations is very similar, the inter-*Relationships* between the various node groups are complex, particularly in stations 4 and 10, near the carina and hilar regions. For example, we believe that AJCC Station 10, named "hilar lymph node", is a synonym for "bronchial lymph node" and "bronchopulmonary lymph node"; however, ATS Station 10R, named "right tracheobronchial lymph node" is not a *subtype* of "tracheobronchial lymph node" because its definition includes nodes covered by both "lower paratracheal lymph node" (AJCC Station 4) and by "hilar lymph node" (AJCC Station 10). We use "tracheobronchial lymph node" as a supertype of both inferior tracheobronchial (subcarinal) and superior tracheobronchial (a subset of lower paratracheal).

6.2.3.6.1.2.12 Prostate lobes

The "posterior lobe" of the prostate is described in newborns but does not persist into the adult. The three prostate lobes [113295002] are the left and right lateral lobes and the variable middle lobe.

6.2.3.6.1.2.13 Abdominal cavity – Pelvic cavity

The term "abdominal cavity" has two meanings, one including the pelvic cavity, the other excluding it. Abdominal cavity structure is used as inclusive of both. Abdominal cavity proper is used as exclusive of the pelvic cavity.

6.2.3.6.1.2.14 Ear – external ear – pinna/auricle

The ear includes external, middle and inner ear. The external ear has two main parts, the auricle (also called the pinna) and the external auditory canal. The external auditory canal is sometimes also called the external auditory meatus. External auditory meatus is not just the external opening of the canal, but rather is a synonym for the canal extending to the ear drum (tympanic membrane). The internal auditory canal is not part of the ear. It is an opening in the temporal bone, and is primarily a nerve conduit that runs roughly parallel to the external auditory canal.

6.2.3.6.1.2.15 Cardiac valves, normal and malformed

A number of concepts have been added to the anatomy hierarchy to support the representation of congenital cardiac malformations. This content was developed in cooperation with IPCCC (International Pediatric and Congenital Cardiac Code).

The following pairs of cardiac valve terms do not represent the same thing, and are siblings not super- or sub- type in relation to each other:

- Atrioventricular valve (body structure) vs. Atrioventricular (non-mitral, non-tricuspid) valve structure (body structure)
- Mitral valve structure (body structure) vs. Left (non-mitral) atrioventricular valve structure (body structure)
- Tricuspid valve structure (body structure) vs. Right (non-tricuspid) atrioventricular valve structure (body structure)

Atrioventricular (non-mitral, non-tricuspid) valves represent body structures which were anatomically abnormal from the beginning of their development. They are not called mitral/tricuspid valve although they perform the same function as their normal counterpart would. They are also represented using the term "not morphologically mitral/tricuspid valve". For example, 459176007 I Abscess of right atrioventricular (not morphologically tricuspid) valve (disorder) I represents an abscess of the right atrioventricular valve that has been developed abnormally from the beginning vs. 431189009 I Abscess of tricuspid valve (disorder) I which represents an abscess on a normally developed tricuspid valve. For a normally developed mitral/tricuspid valve, the term "left/right atrioventricular valve" can be used interchangeably. They are true synonyms. However, they cannot be used for abnormally developed valves, i.e. left atrioventricular (non-mitral)/right atrioventricular (non-tricuspid) valves.

6.2.3.6.2 Morphologic Abnormalities

The morphologic abnormality hierarchy is found two levels below the body structure hierarchy, with siblings apoptosis and tissue repair:

- *SNOMED CT concept*
  - body structure
    - morphologically altered structure
      - I morphologically abnormal structure I
      - apoptosis
      - tissue repair

The codes in the morphologic abnormality hierarchy represent classes of which the instances are all kinds of abnormal body structure.

6.2.3.6.2.1 Morphologic abnormalities vs. Findings

Codes from the morphologic abnormality hierarchy should not be used in place of codes from the clinical findings hierarchy, even though they appear to refer to similar *clinical situations*.

For example, *mass (morphologic abnormality)* [4147007 I Mass I] is not a finding, but I mass of body structure (finding) I [300848003 I Mass of body structure I] is a finding. Morphologies are used as the values of the defining attributes of findings and procedures. Findings are used to represent the combination of a morphology in a location. For example, I cyst of scalp I [300923002 I Cyst of scalp I] represents cystic type of morphology that is in the location scalp.

Many morphologies have names that could be (mis)-interpreted as implying a process rather than a structure. For example, inflammation might mean the structural-morphologic features of inflammation, such as inflammatory cell infiltrates; or it might mean the process that results in those structural changes. Within the morphologic abnormality hierarchy, the structural interpretation is intended, and the process interpretation is not.

6.2.3.6.2.2 Morphology Hierarchy General Structure

The hierarchy immediately under morphologically abnormal structure is given below, with bold font marking the broad categories that correspond to *SNOMED* 3 morphology sections (see text below):

- abnormal cell

- abnormal cellular component of blood
- abnormal shape
- absence
- **cellular or subcellular abnormality**
- collagen shrinkage
- cutaneous patch
- cutaneous plaque
- damage
    - necrosis
    - **traumatic abnormality**
- **degenerative abnormality**
- depressed structure
- **developmental anomaly**
- effect of surgery
- eruption
- exfoliative lesion
- extracellular alteration
- fibrosis or repair abnormality
- fusion
- **growth alteration**
    - proliferation
        - proliferative mass
            - neoplasm and/or hamartoma
                - | neoplasm |
- hernial opening
- heterotopia
- honeycomb appearance
- **inflammatory morphology**
- macule
- mass
- mast cell abnormality
- **mechanical abnormality**
- minimal lesion
- narrowing
- papule
- pigment alteration
- postmortem change
- pseudomembrane
- pseudotumor
  pseudotumour
- redundant tissue
- therapy-related morphologic change
- tumor-like lesion
  tumor-like lesion
- vegetation

- widening

The classical organization of morphology in *SNOMED* 3 had ten sections, including:

Section 0: General morphologic terms [M-0]
Section 1: Traumatic abnormalities [M-1]
Section 2: Congenital anomalies [M-2]
Section 3: Mechanical abnormalities [M-3]
Section 4: Types of inflammation [M-4]
Section 5: Degenerative abnormalities [M-5]
Section 6: Cellular and subcellular abnormalities [M-6]
Section 7: Growth, maturation and non-neoplastic proliferations [M-7]
Section 8: International classification of neoplasms (ICD-O). [M-8 and M-9]
Section 9: Specific veterinary tumors [M-A and M-B]

Although the sections in *SNOMED* 3 were generally correct, a number of changes in the hierarchy were required to satisfy strict logical subtyping. For example, neoplasms are kinds of proliferation, which is a kind of growth abnormality. But fibrosis is not strictly a growth abnormality, so it is placed outside that hierarchy. Also the phrase general morphologic term names a term, not a morphologic abnormality, and this type of general catch-all phrase should be eliminated from the *SNOMED CT* hierarchies.

6.2.3.6.2.3 Specific policies related to morphology

6.2.3.6.2.3.1 Malignant tumor morphology and ICD-O

The origins of malignant tumor morphology codes can be traced to the Systematized Nomenclature of Pathology (SNOP) which was published in 1965. Subsequently the WHO has published three revisions of the morphology of ICD-O, and all three of these have been tightly coordinated with the M-8 and M-9 sections of *SNOMED*. For tumor morphology codes, ICD-O-2 codes and names were the same as both *SNOMED* 2 and *SNOMED* 3, and ICD-O-3 codes and names are the same as the corresponding *concepts* in *SNOMED RT* and *SNOMED CT*.

6.2.3.6.2.3.1.1 Formatting variations between ICD-O morphology and SNOMED CT tumor morphology

Minor formatting variations occur with these codes. In order to distinguish morphology codes from others in *SNOMED*, the *SNOMED* identifier has always prefixed the 5-character code with an M and a dash (-). ICD-O does not routinely do this. Another minor formatting variation that may be seen in ICD-O coded data is a forward slash (/) before the final character of the code. *SNOMED* does not ever do this. As a result, the code for acidophil carcinoma, for example, might appear in any of the following forms:

Character string	***Origin***
M-82803	*SNOMED*
8280/3	ICD-O
M-8280/3	combined

6.2.3.6.2.3.1.2 Non-synonymous synonyms in ICD-O

ICD-O does not necessarily provide a code for each morphologic variation of a given tumor type. It distinguishes between true synonyms and related terms. *SNOMED CT* often provides a morphology concept for the related terms as a *subtype*, but it never assigns an M-8. or M-9 *legacy SNOMED code* to these concepts. Instead it uses R-.. format codes.

6.2.3.6.2.3.1.3 The use of NOS and No ICD-O Subtype

In ICD-O, a term may end with NOS, meaning not otherwise specified. This means not otherwise specified in the patient record that I am coding. But in the context of *SNOMED CT*, the originator of the code is the pathologist or other health care professional who actually generates the original record. In this context, it makes no sense to reference the record that they are coding, because they are using the terminology to represent their original meaning, not someone elses recorded meaning.

The effect of NOS on a particular phrase in ICD-O is to specialize it to mean none of the other *subtypes* of this. For example, adenocarcinoma NOS means an adenocarcinoma that has not been specified to be any of the other kinds of adenocarcinoma that are available for coding in ICD-O. Therefore in *SNOMED CT*, we revise the *fully specified name* of these *concepts* to be no ICD-O *subtype*, meaning that it is a kind of tumor morphology that does not fit any of the other *subtypes* of this morphology that are named in ICD-O.

6.2.3.6.2.3.2 Congenital anomaly

There is significant doubt about the usefulness of some of the congenital anomaly morphologies. Many disorders that involve congenital anomalies can be defined in terms of the OCCURRENCE attribute, leaving no need for a congenital version of the more general morphology *concept*. For example, 90293002 *congenital stenosis (morphologic abnormality)* could be removed because 415582006 *stenosis (morphologic abnormality)* can be combined with OCCURRENCE = *congenital* to define disorders that involve congenital stenosis.

Because they are currently in the hierarchy, these congenital morphology *concepts* should be properly placed. Congenital anomaly morphology *concepts* should usually have non-congenital parents. For example, *congenital stenosis* needs to be a child of *stenosis*, in addition to being a child of *congenital anomaly*.

6.2.3.6.2.3.3 Degeneration vs. Degenerative Abnormality

A distinction should be made between 33359002 I Degeneration I, and 107669003 I Degenerative abnormality I. Degenerative abnormalities, the broad group of *concepts*, are those morphologies characterized by retrogressive pathologic structural changes. Examples of these include degeneration proper as well as lyses, vascular scleroses, necroses and infarcts, depositions, dystrophies, pigmentations, atrophies and depletions. In other words, 107669003 *degenerative abnormality* is a grouping *concept* to put together this set of things that have in common the fact of retrogressive structural degeneration.

Morphologies under degeneration also show retrogressive structural changes, but they are not necessarily any of the following: atrophy, depletion, deposition, dystrophy, lysis, resorption, malacia, necrosis, obliteration, opacity, plaque, vascular sclerosis or postmortem change. This does seem to be definition by exclusion.

Necrosis is a degenerative abnormality, but not a degeneration. Necrosis can **follow** degeneration.

Atrophy is a degenerative abnormality, but only atrophic degeneration is also a degeneration.

As a general rule, we do not assume that diseases called "degenerative" necessarily have I Associated morphology I=33359002 I Degeneration I since the word "degenerative" sometimes refers to loss of function, rather than structural degeneration.

For those disorders called "degenerative" that have a specific structural degeneration, it is preferable to use a more specific value instead of the generic 33359002 I Degeneration I, or even worse, the more general 107669003 I Degenerative abnormality I.

107669003 I Degenerative abnormality I should rarely, if ever, be used as the value of associated-morphology of a particular disorder; rather, a more specific *subtype* should be used as the value. It *might* be used as the value of I Associated morphology I for a broad category of degenerative disorders where the degeneration is always and necessarily structural. It will then be inherited by all the *subtypes*, unless specialized by assigning a particular *subtype* of 33359002 I degeneration I as the value of I Associated morphology I for that disorder.

6.2.3.6.2.3.4 Abscess

Although most abscesses are infectious, there are some sterile abscesses. If a code has a meaning (based on its FSN and text definition) that does not specify whether the abscess is sterile or infectious, then the code should not be modeled as infectious. In these general cases, the code's logic definition uses the attribute| Associated morphology | with the value | abscess morphology | [44132006 | Abscess morphology |].

6.2.3.6.2.3.5 Fracture

Although most fractures are traumatic, there are some pathological fractures. In the vast majority of cases, fractures are traumatic.

6.2.3.7 Pharmaceuticals and biologic products

6.2.3.7.1 Pharmaceutical/biologic product

The | Pharmaceutical / biologic product | *hierarchy* is separate from the | Substance | *hierarchy*. This *hierarchy* was introduced as a top-level *hierarchy* in order to clearly distinguish drug products (products) from their chemical constituents (substances).

It contains *concepts* that represent the multiple levels of granularity required to support a variety of uses cases such as computerized provider *order* entry (CPOE), e-prescribing, decision support and formulary management. The levels of drug products represented in the *International Release* include Virtual Medicinal Product (VMP), Virtual Therapeutic Moiety (VTM), and Product Category. Additionally, US and UK drug *extensions* have been developed, which represent Actual Medicinal Products (AMPs).

Note: See also *Attributes used to define Pharmaceutical/Biologic Product concepts*.

6.2.3.7.2 Attributes used to define Pharmaceutical/Biologic Product concepts

Table 181: Approved Pharmaceutical/Biologic Product attributes summary

Defining Attribute	Allowable Values
| HAS ACTIVE INGREDIENT |	| Substance | 105590001 (<<)
| HAS DOSE FORM |	| Type of drug preparation | 105904009 (<<)

Note:

Permissible values for these attributes include the *concepts* listed and their *descendants*.

Note: See also *Pharmaceutical/biologic product*.

6.2.3.7.2.1 HAS ACTIVE INGREDIENT

This attribute indicates the *active* ingredient of a drug product, linking the | Pharmaceutical / biologic product | *hierarchy* to the | Substance | *hierarchy*.

Table 182: Permissible values for HAS ACTIVE INGREDIENT

Attribute Values	Examples
| Substance | 105590001 (<<)	| Naproxen 500mg tablet (product) | • | HAS ACTIVE INGREDIENT | | Naproxen (substance) |

6.2.3.7.2.2 HAS DOSE FORM

This attribute specifies the dose form of a product.

Table 183: Permissible values for HAS DOSE FORM

Attribute Values	**Examples**
I Type of drug preparation I 105904009 (<<)	I Digoxin 0.1mg capsule (product) I • I HAS DOSE FORM I I Oral capsule (qualifier value) I

6.2.3.7.3 Pharmaceutical/Biologic Product Hierarchy

The following diagram depicts the current structure of the pharmaceutical/biologic product hierarchy.

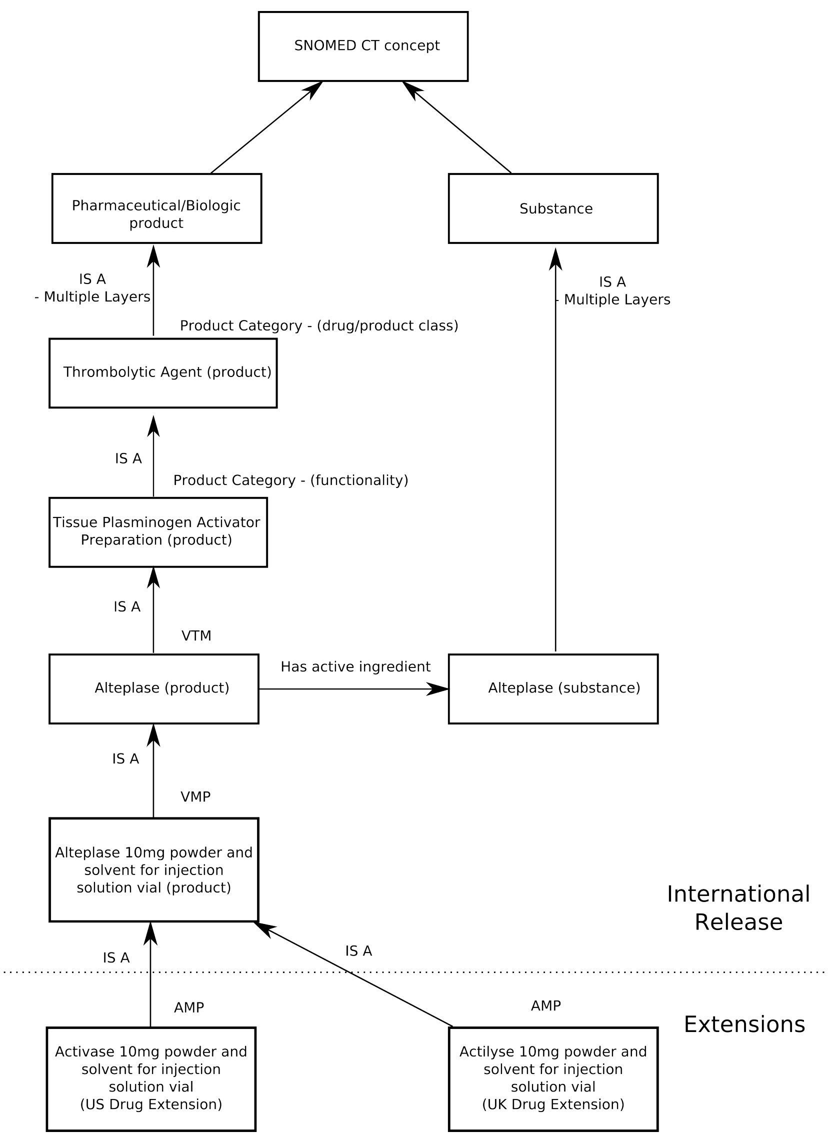


Figure 61: Structural Diagram

6.2.3.7.4 International Release Pharmaceutical/Biologic Model: Draft IHTSDO Standard

In response to a need for improvements to the drug product hierarchy in order to meet user requirements, a revised model has been developed. This model is now a Draft *IHTSDO* Standard. The content of *SNOMED CT* has not yet been revised to conform to the draft model, however.

6.2.3.7.4.1 Concept Classes in Revised Draft Standard International Pharmacy Model

6.2.3.7.4.1.1 Medicinal Entity (ME)

A Medicinal Entity (ME) is the abstract representation of the set of active ingredient(s) (devoid of strength and form), which when formulated as a medicinal product, is intended for use in the treatment of a patient. i.e. For any one Medicinal Entity, one or more NPMP *concepts* must exist, reflecting actual products that are or have been available in one or more *IHTSDO* member nations. *A Relationship may exist between two medicinal entities to support the Relationships between Medicinal Entities that are "moieties" and Medicinal Entities that represent the "moieties with their modifiers" (such as base and salt).*

- A Medicinal Entity is created for a moiety and for the precise ingredient(s) (moiety and modifier) where the active ingredient differs from the active moiety.
- A Medicinal Entity defines a group of products as represented by a set of one or more NPMPs, which contain substances with the same combination of active moieties.
- Medicinal Entities *concepts* will have a *Relationship* to all of their active ingredients, using one or more "has intended active ingredient" *Relationship*(s).

Examples of Medicinal Entities *Fully Specified Name* (FSN) and *Preferred Term* (PT) [1]include:

Type of product	**ME *Fully Specified Name***	**ME *Preferred Term***
Single Ingredient- moiety	amoxicillin (product)	amoxicillin
Single Ingredient – Moiety with modifier	haloperidol decanoate (product)	haloperidol decanoate
Multi-Ingredient – Moiety	codeine + paracetamol (product)	codeine + paracetamol
Multi-Ingredient – "Moiety" and Moiety with modifier	codeine phosphate + paracetamol (product)	codeine phosphate + paracetamol

Points to note

- Medicinal Entity can be a moiety, or a moiety plus modifier where the modifier is required to identify therapeutically relevant modifications or modifications that are the basis of strength substance (BoSS) in one or more *IHTSDO* member nations. In instances where a moiety plus modifier is required for BoSS considerations, it should be for the member nation to identify the requirement on a case by case basis. This means that the BoSS will always be the same *concept* as the active ingredient of the immediate ME parent.
- Therapeutic combinations of separate chemical entities/moieties e.g. "paracetamol and codeine" will be modeled as a single Medicinal Entity.
- Recursion is supported to allow the representation of base and base specialization. This recursion is unrestricted however active review, approval and communication of instances where more than one layer of recursion must be a routine quality assurance activity.
- Pro-drugs will be modeled showing the pro-drug as the ingredient.

[1] RF2 contains no *Description* type value " *Preferred Term*", only types of " *Fully specified name* " and "Synonym", where the latter may be refined either to a "*Preferred term* " or to a "Synonym" within a language *reference set*. As a result of this change from RF1 to RF2, the preference for particular *Descriptions* in a language or *dialect* will be represented using a *reference set*.

6.2.3.7.4.1.2 Non Proprietary Medicinal Preparation (NPMP)

A Non Proprietary Medicinal Preparation (NPMP) is an abstract concept representing the properties of one or more clinically equivalent proprietary product units of use (from (existing or past) National extensions).

A NPMP is the abstract representation of the set of active ingredient(s) and their strength(s) and Dose form

Where strength is a ratio (concentration) example 500 mg per unit dose for discrete dosage forms or 50 mg/mL for continuous represented at a single unit level e.g. per mL or per g

A new NPMP will be created for each different strength of a licensed medicinal product. If an existing product has a change of ingredient, such that it does not conform to the ingredients of the original NPMP, then a new NPMP will be created for the new product.

Please note that all *Non Proprietary Medicinal Preparation concepts* will inherit *Relationships* to all of their active ingredients, as identified by the 'has active ingredient' *Relationship* from their parent ME *concept*.

Examples of *Non Proprietary Medicinal Preparation Fully Specified Name* and *Preferred Term* include:

Type of Product	***Fully Specified Name***	***Preferred Term***
Single ingredient product - (moiety)	amoxicillin 500 mg capsule (product)	amoxicillin 500 mg capsule
Single ingredient product (moiety)	diclofenac 46.54 mg tablet (product)	diclofenac 46.54 mg tablet
Single ingredient product (moiety with modifier)	diclofenac sodium 50 mg tablet (product)	diclofenac sodium 50 mg tablet
Multi-Ingredient (moiety)	codeine 23.43 mg + paracetamol 500 mg tablet (product)	codeine 23.43 mg + paracetamol 500 mg tablet
Multi-Ingredient (moiety with modifier)	codeine phosphate 30 mg + paracetamol 500 mg tablet (product)	codeine phosphate 30 mg + paracetamol 500 mg tablet

Points to note

- *Non Proprietary Medicinal Preparation* can identify a moiety, or a moiety plus modifier where the modifier is required to identify therapeutically relevant modifications or modifications that are the basis of strength substance (BoSS) in one or more *IHTSDO Member* nations. In instances where it is required as the BoSS it should be for the member nation to identify the requirement on a case by case basis.
- Recursion is supported to allow the representation of base and base plus specialization. This recursion is unrestricted however active review, approval and communication of instances where more than one layer of recursion must be a routine quality assurance activity.
- Pro-drugs will be modeled showing the pro-drug as the ingredient.
- Member nations may create an NPMP *Preferred term* in their own extension where an alternative *Description* is required. Member nations may also create Basis of Strength *Relationships* and associated values in their own extension to support this alternative *Description*.

6.2.3.7.4.2 Expression of Strength

6.2.3.7.4.2.1 Representation of numeric values

It is recognized that there is a requirement for the *expression* of pharmaceutical strength in *SNOMED CT* to allow the full definition of medicinal product *concepts*, however the method utilized to achieve this is not yet determined. The representation of strength using *reference sets* in this document is therefore illustrative based upon the current view of best practice.

6.2.3.7.4.2.2 Active ingredient and reference BoSS association

As part of the *expression* of strength for full definition of NPMP *concepts* and to support the automatic generation of FSN and PT *Descriptions* for NPMPs where the active ingredient differs from the BoSS there is a requirement that each 'has reference BoSS' *Relationship* be associated with the appropriate 'has active ingredient' *Relationship*. Without this it may not always be possible to automatically identify the active ingredient for which the Basis of Strength Substance is measuring the strength. This difficulty most commonly occurs when there is more than one active ingredient in the product, which share the same base (but with different modifications or level of modification). For example, 'Sodium Citrate + Citric Acid' (ME). If the BOSS strength (e.g. Citric Acid 66.8 mg/mL) is associated with the incorrect 'active ingredient', then the NPMP actually represents a different product.

To address this issue, it is proposed that '*Relationship groups* ' are used to group each 'has reference BoSS' *Relationship* together with the appropriate 'has active ingredient' *Relationship*. The exact nature of these requirements are to be addressed in a proposal to support machine processable strengths in *SNOMED CT*.

6.2.3.7.4.3 New Attributes in International Pharmacy Model

The model as proposed will require a number of new attributes to be created in *SNOMED CT* to support associations between *concepts*. These are:

6.2.3.7.4.3.1 Plays therapeutic role

Because one of the main consequences of the new draft standard is a revision of the upper levels of the Pharmaceutical and Biological product hierarchy (any *concepts* above the Medicinal Entity or Medicinal Entity with Modifier level) to eliminate inappropriate inheritance caused by *IS_A* overloading, there is a requirement to replace the removed inappropriate data.

Grouping *concepts* indicating mode of action or structure are to be relocated in a separate part of the Pharmaceutical and Biological product hierarchy under a single parent *concept* of Therapeutic product group (product).

These relocated *concepts* are to be associated with the medicine *concepts* in the main pharmaceutical and biological product hierarchy by a role based *Relationship* of PLAYS_THERAPEUTIC_ROLE rather than the current *IS_A Relationship*.

Example:

I propranolol (product) I

PLAYS_THERAPEUTIC_ROLE I non-selective beta-blocking agent (product) I

Permissible values include the *descendants* of the *concept* Therapeutic product group (product).

This would not be a defining *Relationship* if not always necessarily true of all *subtypes*; a non-defining *Relationship* would be created instead. This *Relationship* will be used to associate Medicinal Entity (or Medicinal Entity with modifier) *concepts* from the Pharmaceutical and Biological product hierarchy to *concepts* within the same hierarchy that identify the mode of action.

One Medicinal Entity (or Medicinal Entity with modifier) *concept* can have zero to many PLAYS_THERAPEUTIC_ROLE *Relationships*, depending on whether it has multiple modes of action or the mode of action is unknown or currently unspecified.

The level of specificity for the grouping *concepts* would be in response to the user community requests or ATC groupings.

See also *Revision of upper levels* on page 254

6.2.3.7.4.3.2 Has reference BoSS

This attribute identifies the substance used to describe the strength of the NPMP drug product, and provides a *Relationship* between *concepts* in the Pharmaceutical/Biologic product hierarchy and *concepts* in the Substance hierarchy.

Domain and *concept* modeling details for this attribute will be further specified as part of the *expression* of strength work.

N.B.The existing HAS_ACTIVE_INGREDIENT attribute is also used to provide a *Relationship* between the ME drug product *concept* and *concepts* in the Substance hierarchy.

When the proposals for the *expression* of numbers in *SNOMED CT* are finalized there may be further attributes required to support the full definition of pharmaceutical *concepts*. This will be the subject of a further document.

6.2.3.7.4.4 Diagram of International Pharmacy Model

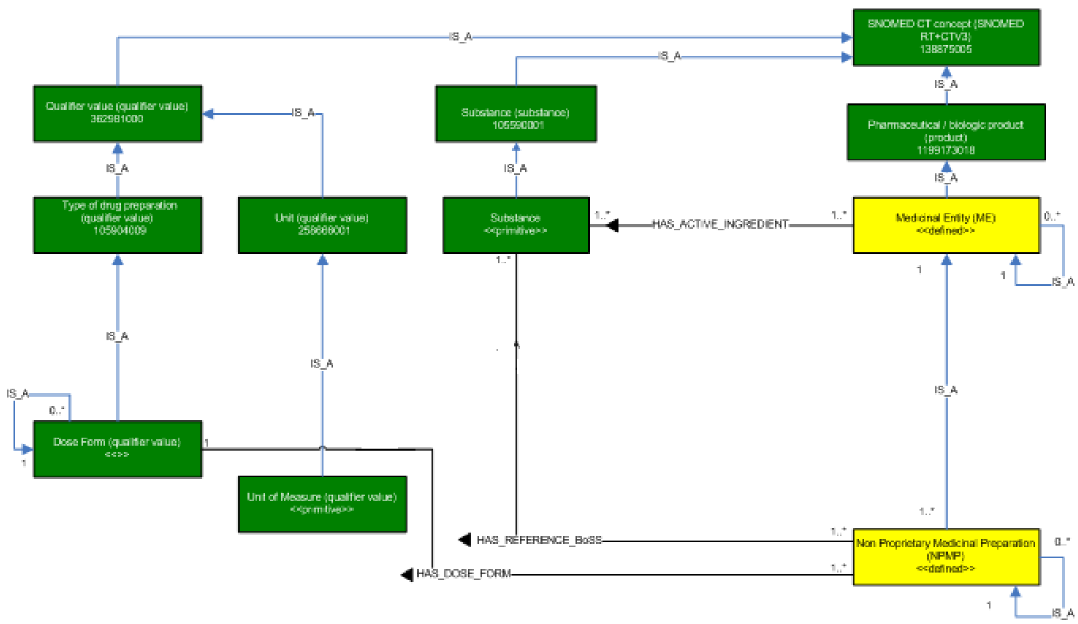


Figure 62: The International Pharmacy Model

6.2.3.7.4.5 Impact of Release Format 2 (RF2)

The model as proposed in this and the other associated documents is able to be expressed in the current *SNOMED CT Release Format* and in the new RF2 format. However neither the current *Release Format* nor RF2 as it is currently proposed support the *expression* of numeric values in a machine processable manner. The requirement for machine processable numeric values is necessary to support the *expression* of pharmaceutical strength in *SNOMED CT* and to allow the full definition of medicinal product (NPMP) *concepts*.

Options for the *expression* of pharmaceutical strength have been requested from the Implementation and Innovation committee and proposals for an amendment to the RF2 *Release Format* are being be circulated in a separate document by that committee for comment. These proposals will permit the representation of numeric values by the use of concrete domains.

Since the proposals for the representation of numeric values are currently undergoing the *IHTSDO* consultation process any reference in this, or the other associated documents, on representation of strength is therefore illustrative based upon the current view of best practice.

6.2.3.7.4.6 International Pharmacy Model - noted

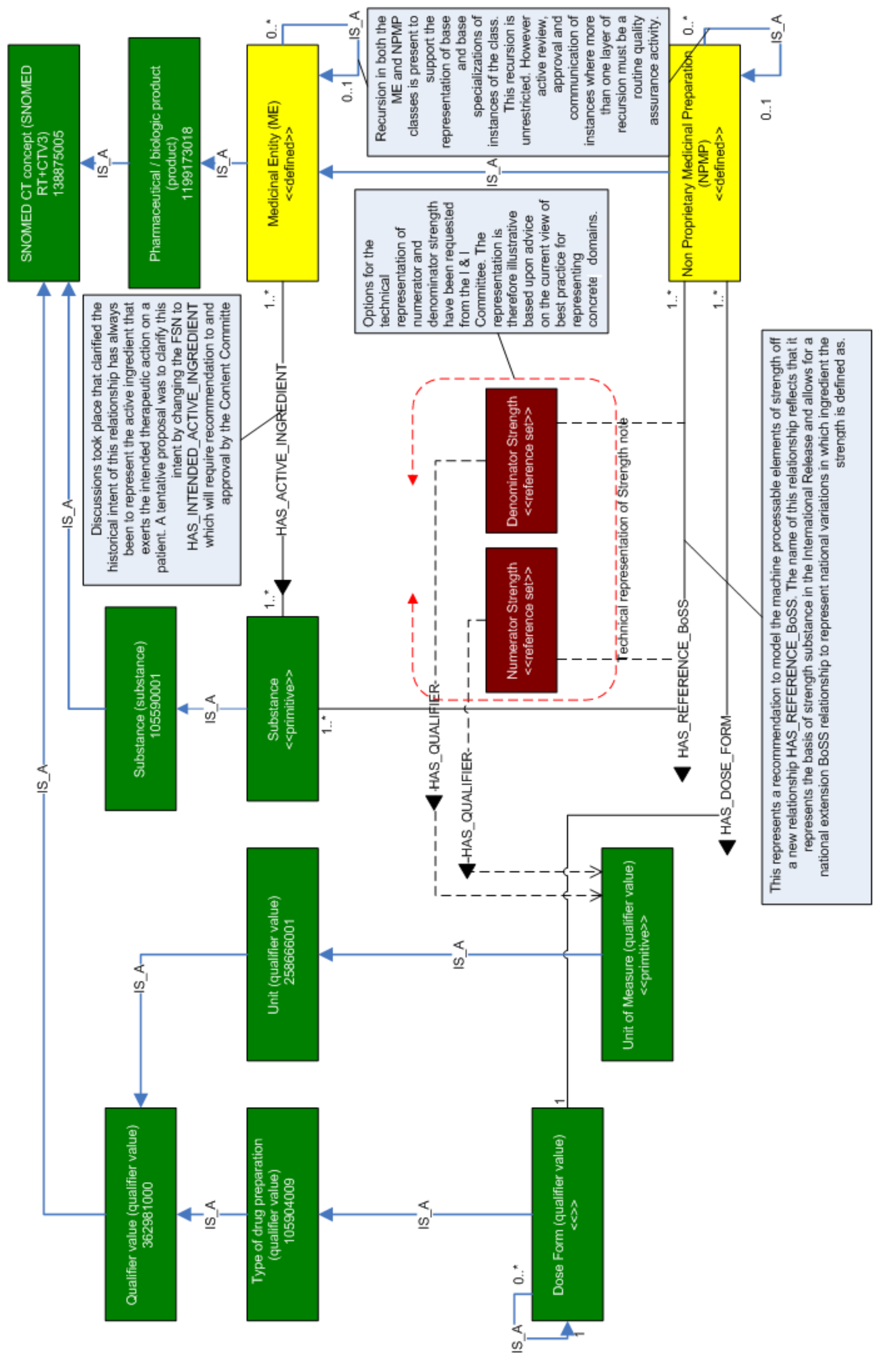


Figure 63: International Pharmacy Model - Noted

6.2.3.7.4.7 International Pharmacy Model - expanded

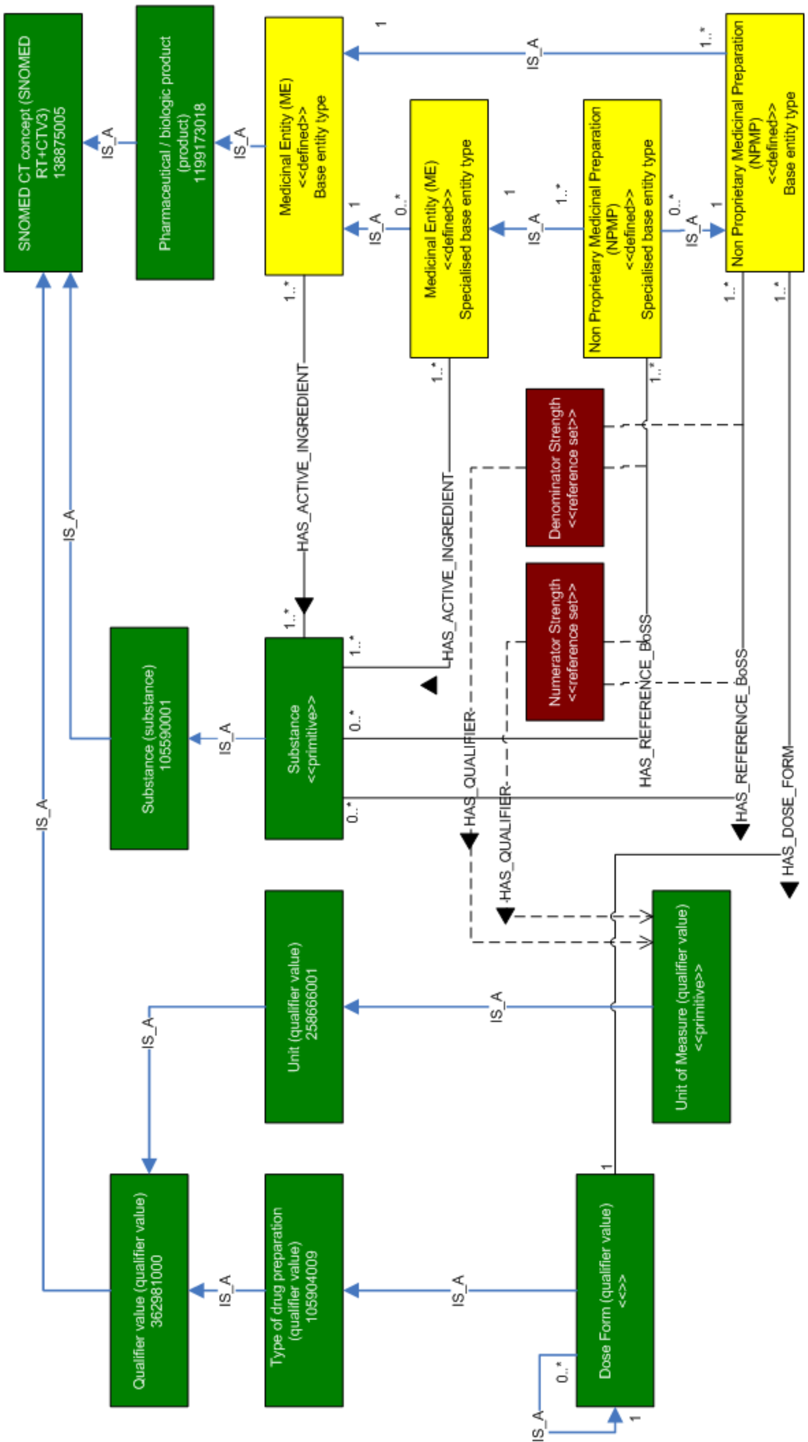


Figure 64: International Pharmacy Model - Expanded

6.2.3.7.4.8 Populated examples

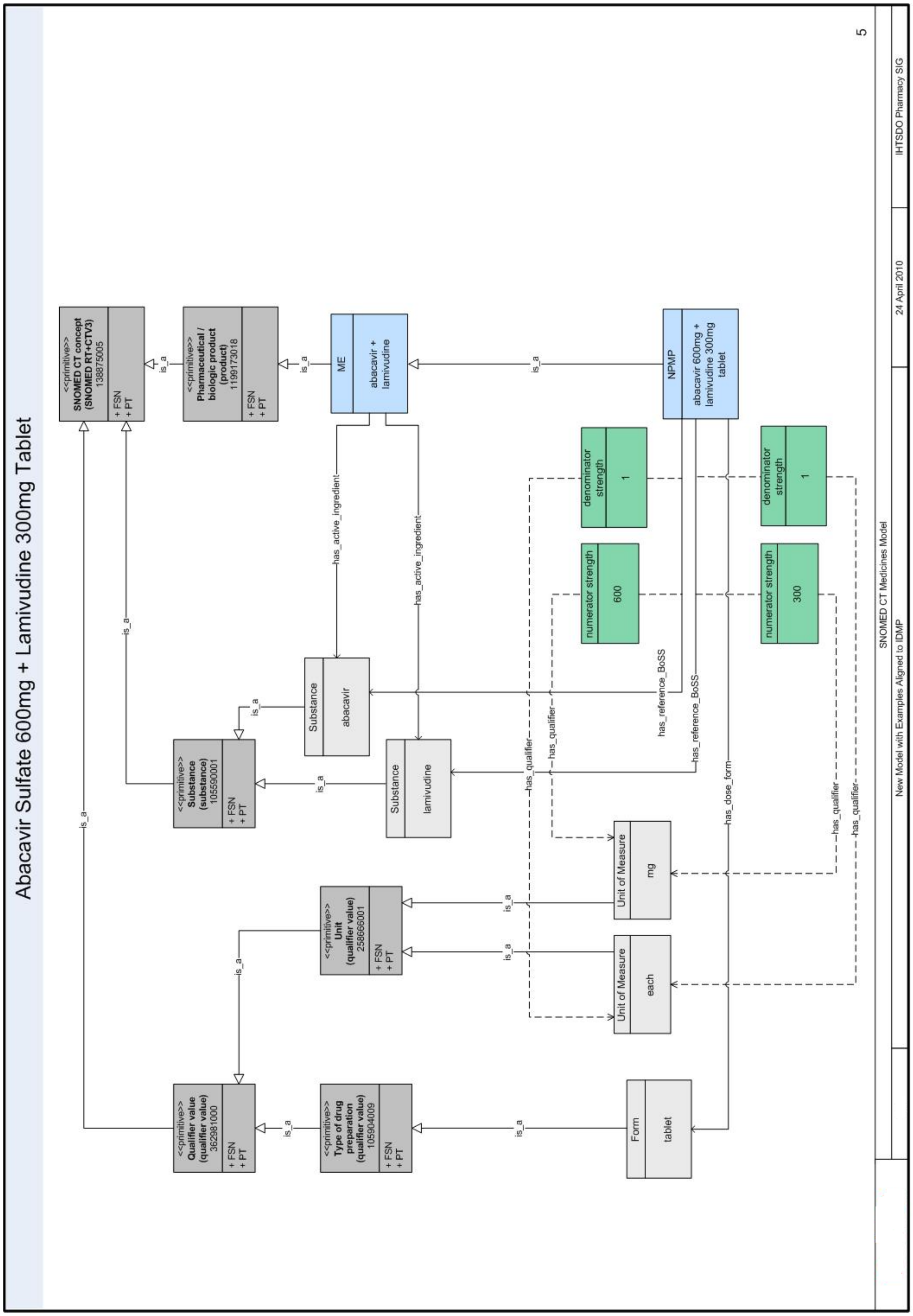


Figure 65: Abacavir Sulfate 600mg + Lamivudine 300mg Tablet

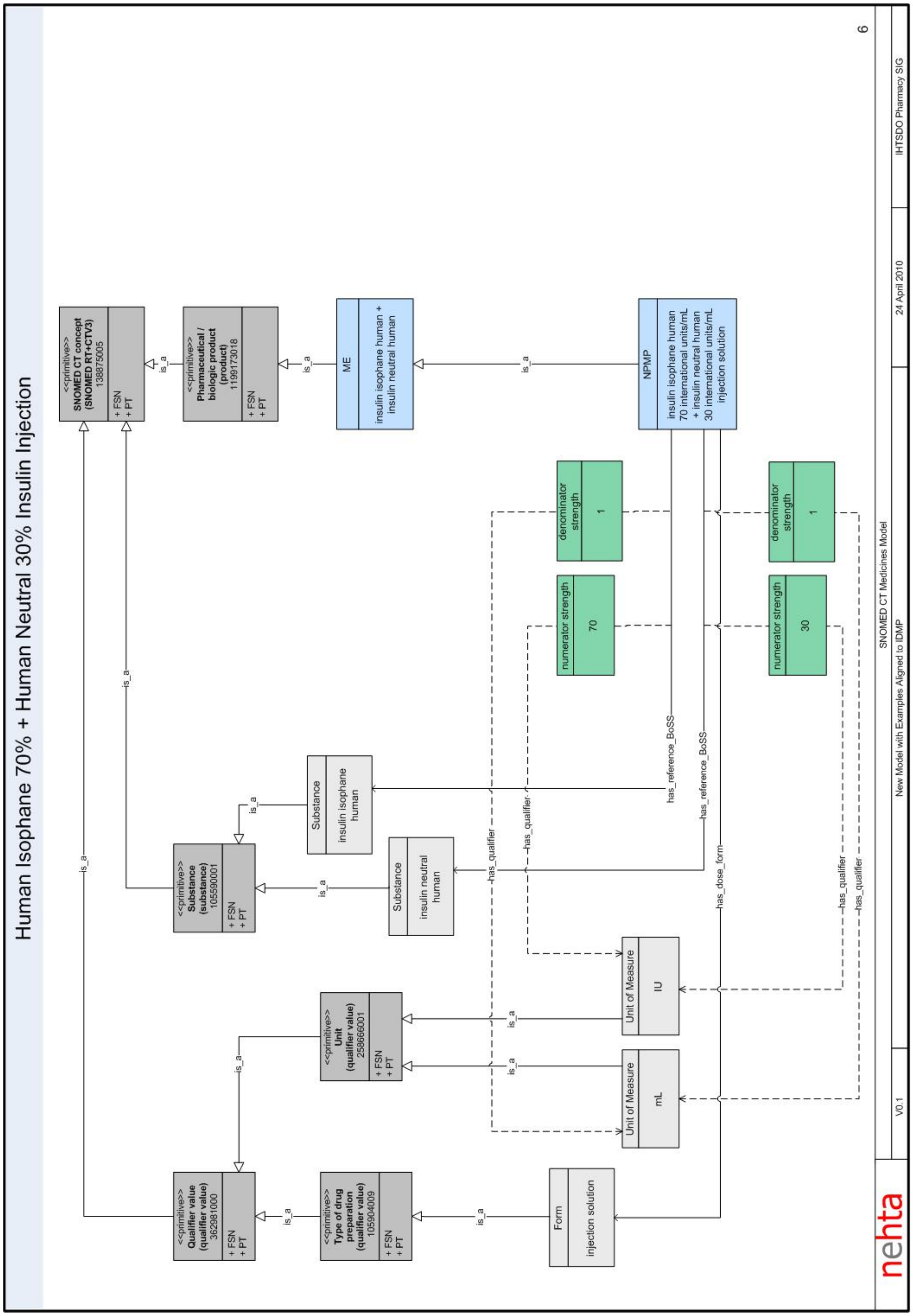


Figure 66: Human Isophane 70% + Human Neutral 30% Insulin Injection

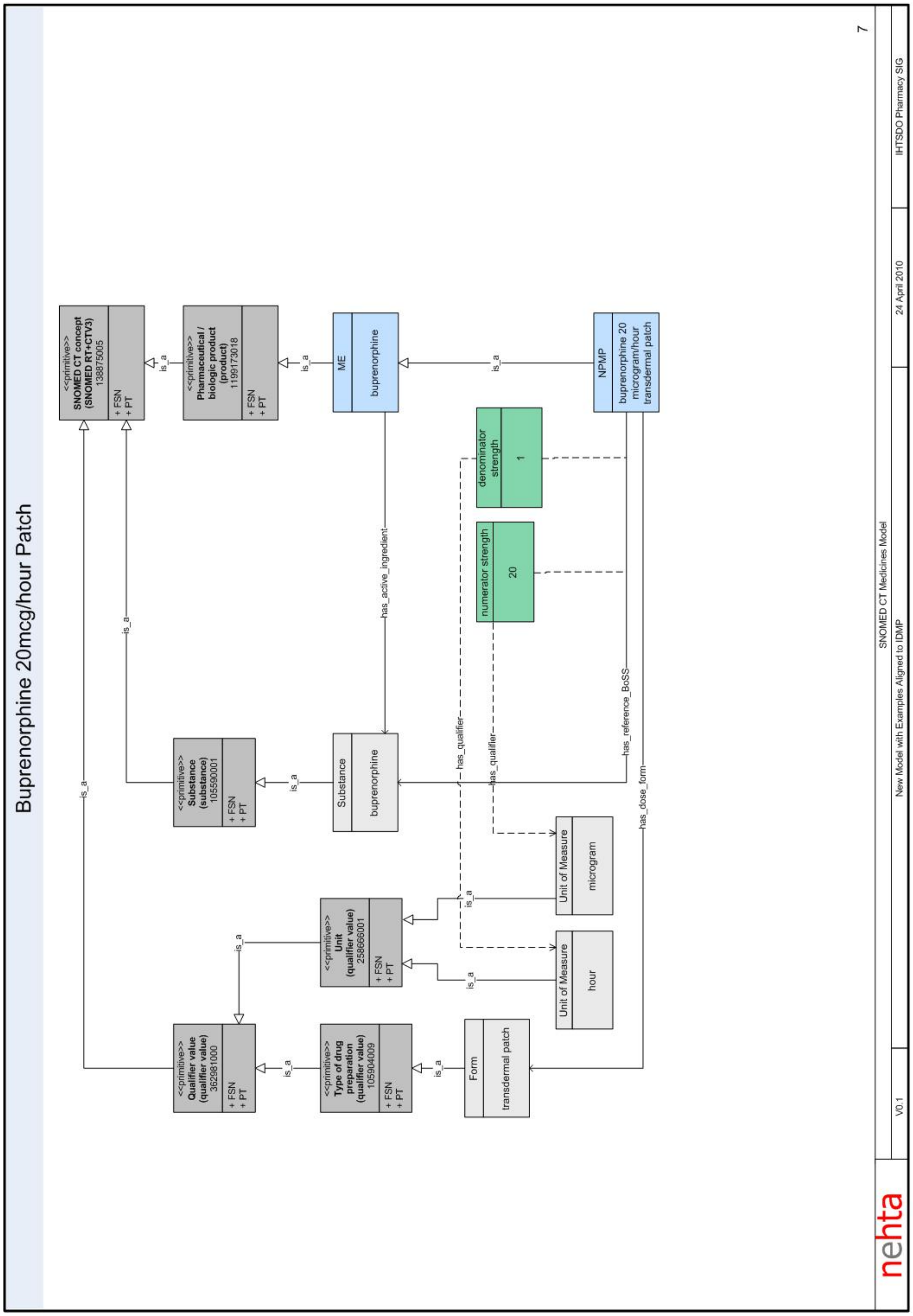


Figure 67: Buprenorphine 20mcg/hour Patch

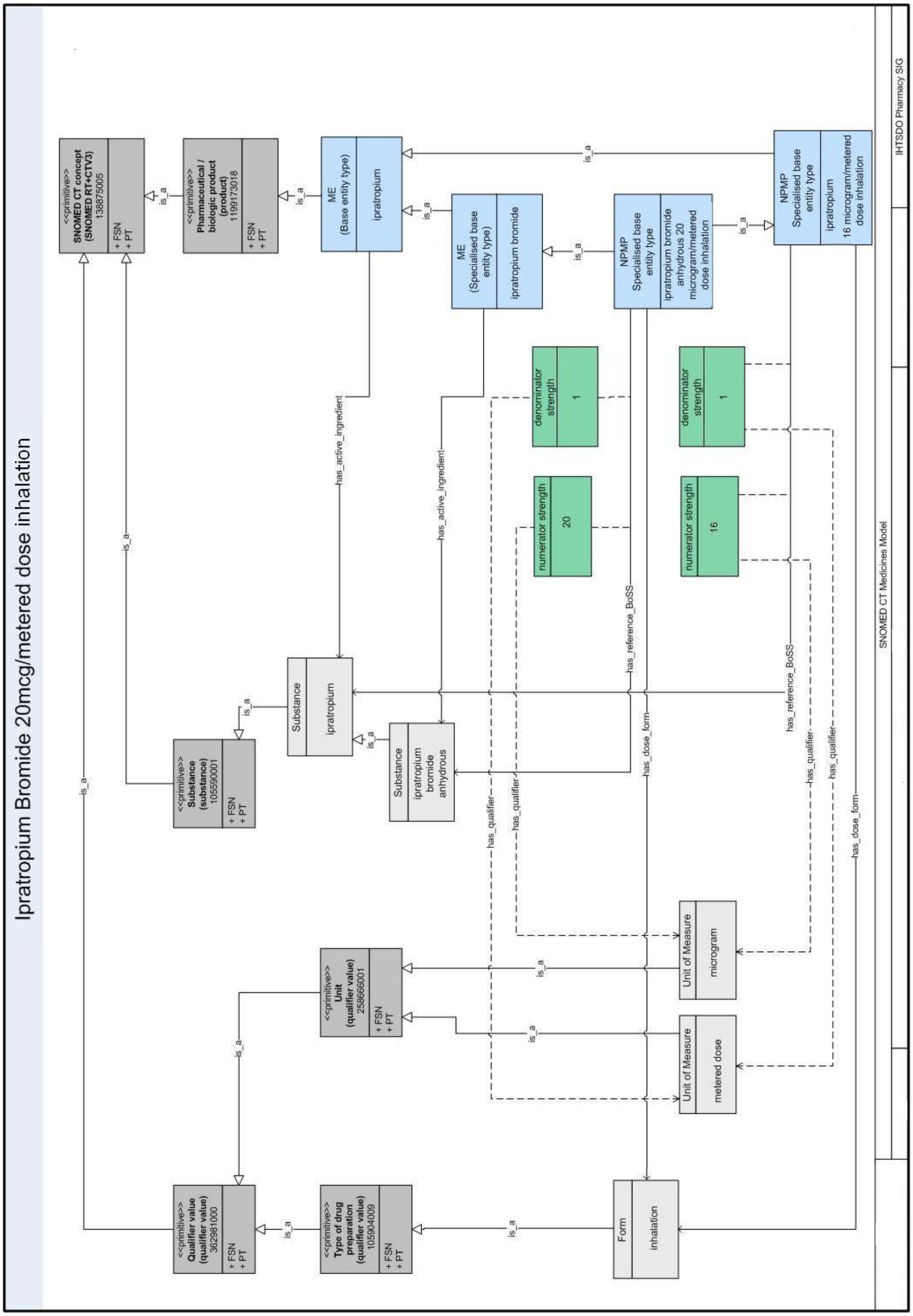


Figure 68: Ipratropium Bromide 20mcg/metered dose inhalation

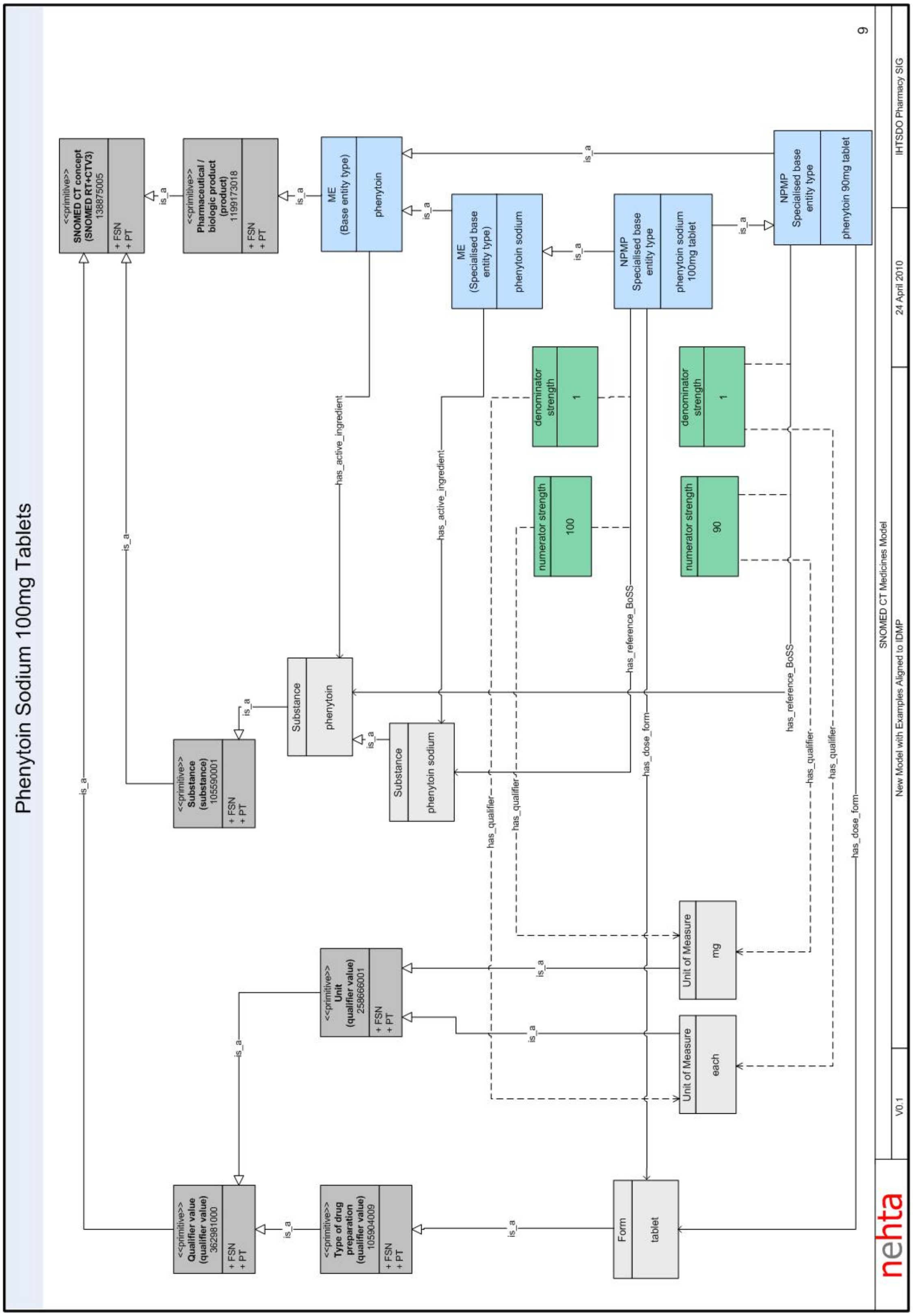


Figure 69: Phenytoin Sodium 100mg Tablets

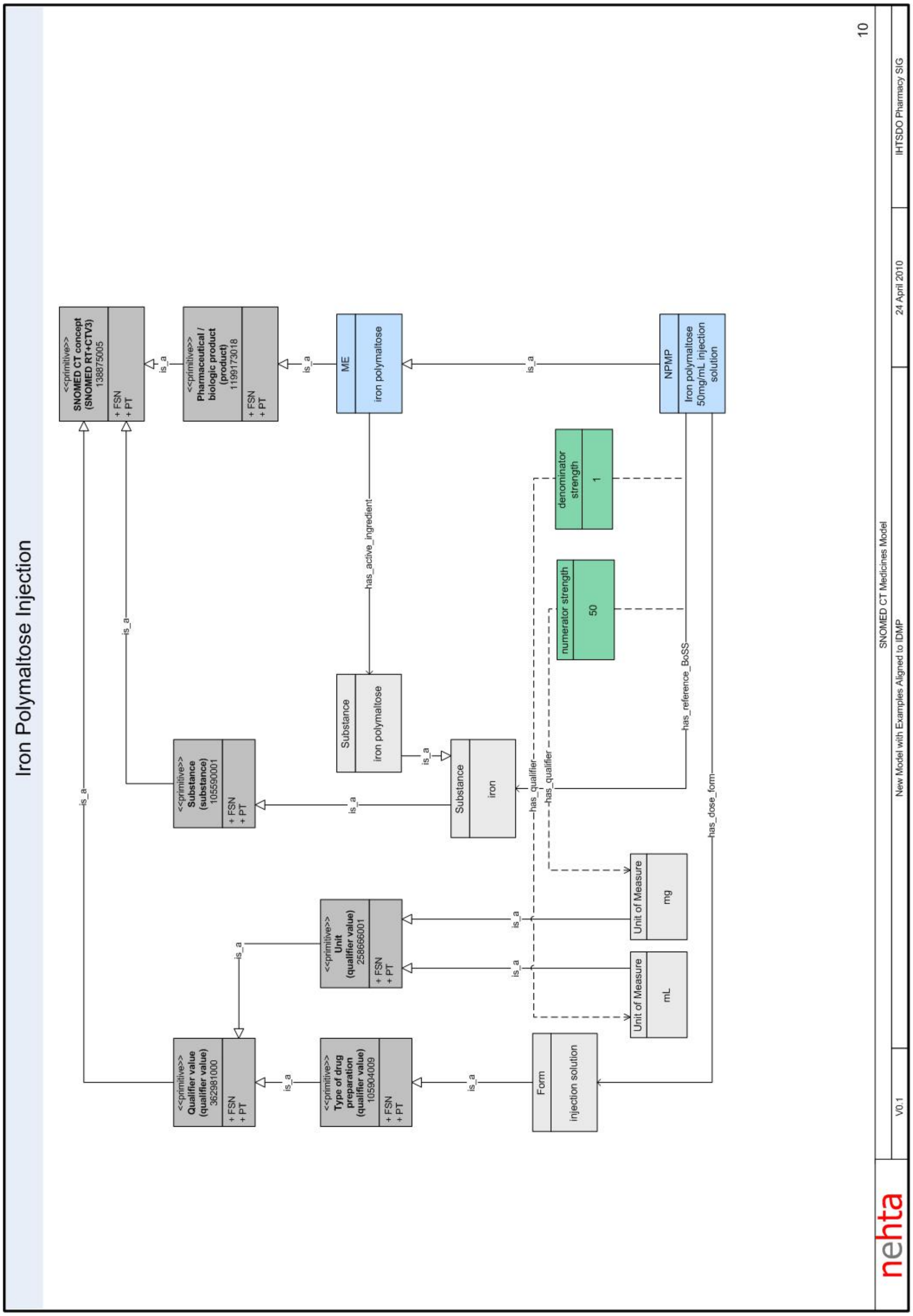


Figure 70: Iron Polymaltose Injection

6.2.3.7.5 Definitions relating to pharmaceutical / biologic product concepts

6.2.3.7.5.1 Virtual Medicinal Product (or clinical drug) VMP

The Virtual Medicinal Product (VMP) is the conceptual representation of one or more clinically equivalent Actual Medicinal Products (AMP) the purpose of which is to support the representation of the fundamental reality of the *concept*. Its core *Description* requires product name, strength, dosage form and unit dose where appropriate (see *Unit Dose Representation* on page 442 for guidance on the applicability of unit dose representation) but is devoid of explicit or implicit information attributable to the manufacturer or pack.

Example:

Clemastine	1mg	tablet
Name	**Strength**	**Dose form**

The Virtual Medicinal Product *concept* type is the most granular level of *concept* representation expressed in the core of *SNOMED CT*.

In instances where the AMP contains a salt weight as a representation of product dose strength, then the salt form must be represented in the VMP.

Example (product with salt form):

- Synflex 275mg tablet (AMP)

The 275mg strength depicts the amount of Naproxen sodium present and not the amount of base Naproxen.

In this instance, the VMP would be Naproxen sodium 275mg tablet.

6.2.3.7.5.2 Virtual Therapeutic Moiety VTM

A VTM is the abstract conceptual representation of the material (i.e. the product) defining the prescribers therapeutic intent, separate from formulation, dose or strength. The HAS_ACTIVE_INGREDIENT attribute will be assigned at this level. Their *descendants* will inherit this attribute.

Examples:

- Atenolol (product)
- Chlorthalidone (product)

6.2.3.7.5.3 Virtual Therapeutic Moiety Subtype VTM Subtype

A VTM *Subtype* is a VTM that contains dose form information and should appear as a child of a VTM. It does not contain the product's strength, and is therefore not a VMP.

Example:

Parenteral epinephrine (product).

Rule : A VTM sub-type will always be included in the hierarchical representation of any new *concepts*, regardless of the number of sibling terms. For instance, the hierarchy for the drug *concept* Olmesartan will be as follows:

1. VTM: Olmesartan
2. VTM sub-type Oral Olmesartan
3. VMP Olmesartan 10mg tablet

6.2.3.7.5.4 Combination Virtual Therapeutic Moiety - Combination VTM

A Combination VTM is a VTM that refers to a combination of ingredients and/or product categories.

Examples:

- Pindolol+diuretic
- Piperacillin+tazobactam
- Piperazine+sennosides
- Rifampicin+isoniazid+pyrazinamide

These *concepts* may be useful to aggregate common classes of VMPs.

Combination VTMs will have as their parent *concepts* the single consituent VTMs. For example the parent *concepts* of Pindolol+diuretic could be both Pindolol and Diuretic.

Combination VTMs are appropriate when there is a single delivery agent that contains both agents.

Example:

- Septra tablet *IS A* Sulfamethoxazole + trimethoprim

Multiple ingredient substances such as Co-Amilofruse should not be used as the value of the HAS_ACTIVE_INGREDIENT attribute for these *concepts*. Instead, multiple single ingredient substances should be applied as the value of the HAS_ACTIVE_INGREDIENT attribute, in this case Frusemide and Amiloride.

6.2.3.7.5.5 Actual Medicinal Product AMP

The Actual Medicinal Product (AMP) is the representation of a single unit dose of a medicinal product that is (or has been) made or marketed by a specific manufacturer or supplier. Its core *Description* requires product name, strength, dosage form, flavor (where applicable) and manufacturer/supplier, but is devoid of explicit information attributable to pack size.

Because they represent domain specific facts Actual Medicinal Product *concepts* will not be expressed within the core of *SNOMED CT*, but exist wholly within any suitably identified domain extension.

Example:

- Zyrtec 10mg tablet

6.2.3.7.5.6 Product Category

A Product Category *concept* supports a group of Pharmaceutical / biologic products related by their functionality or drug/product class. A specific ingredient does not define these *concepts*.

Examples:

- Sex hormone product (product)
- Mineralocorticoid preparation (product)
- beta-Blocking agent (product)
- Tissue plasminogen activator preparation (product)

It is proposed that future authoring will see all Product Categories standardized as preparations as opposed to using a combination of descriptors e.g. products, agents, etc.

6.2.3.7.5.7 Active Ingredient

Any drug component that is intended to furnish pharmacological activity or other direct effect in the diagnosis, cure, mitigation, treatment, or prevention of disease, or to affect the structure or any function of the body of man or other animals.[19]

6.2.3.7.5.8 Substance

The chemical ingredients of products, reside in Substance hierarchy in the *SNOMED CT core*

Values of Has_active ingredient, Direct Substance, Component, Causative agent

6.2.3.7.5.9 Biologic Product

Any virus, therapeutic serum, toxin, antitoxin or analogous product applicable to the prevention, treatment or cure of diseases or injuries in man. [20]

6.2.3.7.5.10 Drug

Any article intended for use in the diagnosis, cure, mitigation, treatment or prevention of disease in man.

6.2.3.7.5.11 Drug Product

A finished dosage form (tablet, capsule, solution, etc.) that contains an active drug ingredient generally, but not necessarily, associated with inactive ingredients. This also includes a finished dosage form that does not contain an active ingredient but is intended to be used as a placebo.[21]

6.2.3.8 Organism

This *hierarchy* includes organisms of significance in human and animal medicine. Organisms are also used in modeling the causes of diseases in *SNOMED CT*. They are important for public health reporting of the causes of notifiable conditions and for use in evidence-based infectious disease protocols in clinical decision support systems. Sub-hierarchies of organism include, but are not limited to: | Animal (organism) |, | Microorganism (organism) |, | Kingdom Plantae (organism) |.

Examples of Organism concepts:

- | Streptococcus pyogenes (organism) | ;
- | Texon cattle breed (organism) | ;
- | Bacillus anthracis (organism) | ;
- | Lichen (plant) (organism) | .

6.2.3.9 Substance

The | Substance | *hierarchy* contains *concepts* that can be used for recording *active* chemical constituents of drug products, food and chemical allergens, adverse reactions, toxicity or poisoning information, and physicians and nursing *orders*. *Concepts* from this *hierarchy* represent general substances and chemical constituents of | Pharmaceutical / biologic product (product) | which are in a separate *hierarchy*. However, sub-hierarchies of | Substance | also include but are not limited to: | Body substance (substance) | (*concepts* to represent body substances); | Dietary substance (substance) |; | Diagnostic substance (substance) |.

Examples of Substance concepts:

- | Insulin (substance) | ;
- | Methane (substance) | ;

---

19 *Fundamentals of Regulatory Affairs*, Regulatory Affairs Professionals Society, September 2002, ISBN 0-9673115-1-9, p. 265

20 *Fundamentals of Regulatory Affairs*, Regulatory Affairs Professionals Society, September 2002, ISBN 0-9673115-1-9, p. 265

21 *Fundamentals of Regulatory Affairs*, Regulatory Affairs Professionals Society, September 2002, ISBN 0-9673115-1-9, p. 266

- | Chromatin (substance) | ;
- | Dental porcelain material (substance) | ;
- | Albumin (substance) | ;
- | Endorphin (substance) | ;
- | Acetaminophen (substance) | .

Combined substances added to the Substance hierarchy that are intended to be used as ingredients for pharmaceuticals must meet the following criteria:

- The physiologic or biologic action of the combination must be enhanced or synergistic
- Combinations that do not have an enhanced or synergistic effect (e.g. combinations created for convenience) are out of scope.

Examples of combined substances that are in scope include:

- ampicillin + sulbactam
- piperacillin + tazobactam

Combined substances are in scope when useful to represent the results of laboratory tests.

Examples of combined substances that are in scope because the value reported by a laboratory measurement is a single combined total of multiple substances include:

- kynurenate + xanthurenate
- trypsin + trypsinogen

6.2.3.10 Specimen

The | Specimen | *hierarchy* contains *concepts* representing entities that are obtained (usually from a patient) for examination or analysis. | Specimen | *concepts* can be defined by attributes which specify: the normal or abnormal body structure from which they are obtained; the procedure used to collect the specimen; the source from which it was collected; and the substance of which it is comprised.

Examples of Specimen concepts:

- | Specimen from prostate obtained by needle biopsy (specimen) | ;
- | Urine specimen obtained by clean catch procedure (specimen) | ;
- | Calculus specimen (specimen) | ;
- | Cerebroventricular fluid cytologic material (specimen) | .

Note: See also *Attributes used to define Specimen concepts*.

6.2.3.11 Physical object

Concepts in the | Physical object | *hierarchy* include natural and man-made objects. One use for these *concepts* is modeling procedures that use devices (e.g. catheterization).

Examples of Physical object concepts:

- | Military vehicle (physical object) | ;
- | Implant, device (physical object) | ;
- | Artificial kidney, device (physical object) | ;
- | Latex rubber gloves (physical object) | ;
- | Book (physical object) | ;
- | Pressure support ventilator (physical object) | ;
- | Vena cava filter (physical object) | .

Note: See also *Attributes used to define Physical Object concepts*.

6.2.3.12 Physical force

The *concepts* in the | Physical force | *hierarchy* are directed primarily at representing physical forces that can play a role as mechanisms of injury.

Examples of Physical force concepts:

- | Spontaneous combustion (physical force) | ;
- | Alternating current (physical force) | ;
- | Friction (physical force) | .

6.2.3.13 Event

The | Event | *hierarchy* includes *concepts* that represent occurrences (excluding procedures and interventions).

Examples of Event concepts:

- | Flood (event) | ;
- | Bioterrorist attack (event) | ;
- | Earthquake (event) | .

Note: See also *Attributes used to define Event concepts*.

6.2.3.14 Environments and geographic locations

The | Environment or geographical location | *hierarchy* includes types of environments as well as named locations such as countries, states, and regions.

Examples of Environments and geographic locations concepts:

- | Canary islands (geographic location) | ;
- | California (geographic location) | ;
- | Rehabilitation department (environment) | ;
- | Intensive care unit (environment) | .

6.2.3.15 Social context

The | Social context | *hierarchy* contains social conditions and circumstances significant to healthcare. Content includes such areas as family *status*, economic *status*, ethnic and religious heritage, life style, and occupations. These *concepts* represent social aspects affecting patient health and treatment. Some sub-hierarchies of | Social context | and *concepts* typical of those sub-hierarchies are shown in the following examples.

Examples:

- | Ethnic group (ethnic group) |:
  - | Afro-Caribbean (ethnic group) | ;
  - | Estonians (ethnic group) | .
- | Occupation (occupation) |:
  - | Bank clerk (occupation) | ;
  - | Carpenter, general (occupation) | .
- | Person (person) |:
  - | Employer (person) | ;
  - | Boyfriend (person) | ;

    - | Caregiver (person) | .

- | Religion / philosophy (religion/philosophy) |:
    - | Hinduism (religion/philosophy) | ;
    - | Orthodox Christian religion (religion/philosophy) | .
- | Economic status (social concept) |:
    - | Middle class economic status (social concept) | .

6.2.3.16 Staging and scales

This *hierarchy* contains such sub-hierarchies as | Assessment scales (assessment scale) |, which names assessment scales; and | Tumor staging (tumor staging) | , which names tumor staging systems.

Examples of Assessment scales (assessment scale) concepts:

- | Glasgow coma scale (assessment scale) | ;
- | Stanford Binet intelligence scale (assessment scale) | .

Examples of Tumor staging (tumor staging) concepts:

- | International Federation of Gynecology and Obstetrics (FIGO) staging system of gynecological malignancy (tumor staging) | ;
- | Dukes staging system (tumor staging) | .

6.2.3.17 Qualifier value

The | Qualifier value | *hierarchy* contains some of the *concepts* used as values for *SNOMED CT* attributes that are not contained elsewhere in *SNOMED CT*. Such a code may be used as the value of an attribute in a defining *Relationship* in *precoordinated* definitions, and/or as the value of an attribute in a *qualifier* in a *postcoordinated expression*. However, the values for attributes are not limited to this *hierarchy* and are also found in hierarchies other than | Qualifier value |.

For example, the value for the attribute | LATERALITY | in the *concept* shown below is taken from the | Qualifier value | *hierarchy*:

- | Left kidney structure | | LATERALITY | | Left | .

However, the value for the attribute | FINDING SITE | in the *concept* shown below is taken from the | Body structure | *hierarchy*, not the | Qualifier value | *hierarchy*.

- | Pneumonia | | FINDING SITE | | Lung structure | .

Examples of Qualifier value concepts:

- | Unilateral | ;
- | Left | ;
- | Puncture - action | .

6.2.3.18 Special concept

The Top Level *Concept* Code | Special concept | and its subclass codes provide a place for *concept* codes that are no longer *active* in the terminology.

The subclasses of | Special concept | are:

- | Navigational concept | ;
- | Inactive concept | .

6.2.3.18.1 Navigational concept

These *concept* codes are to be used only as nodes in alternative navigation structures. Navigational *concepts* are distributed as active *concepts*, with an inactive *Is-a Relationship* to the *concept* "Navigational *concept*". These *concepts* should have no *subtypes*.

6.2.3.18.2 Inactive concept (RF2)

When a *concept* is no longer intended for active use, the *concept* and its *Relationships* are turned inactive. Previously, in RF1, the *concept* was moved into an special "Inactive *concept*" hierarchy, this is not done anymore in RF2, the *concept* is inactivated "in place", with its last location described in the history of its inactive *Relationships*.

6.2.3.18.3 Namespace concept

The *concept* 370136006 | Namespace concept (namespace concept) | is a *subtype* of | SNOMED CT model component |. Each of its *subtype concepts* has an integer term which is an assigned *Extension namespace identifier*.

6.2.3.19 Record artifact

A | Record artifact | is an entity that is created by a person or persons for the purpose of providing other people with information about events or states of affairs. In general, a record is virtual, that is, it is independent of its particular physical instantiation(s), and consists of its information elements (usually words, phrases and sentences, but also numbers, graphs, and other information elements). | Record artifact | need not be complete reports or complete records. They can be parts of larger | Record artifact |. For example, a complete health record is a | Record artifact | that also may contain other | Record artifact | in the form of individual documents or reports, which in turn may contain more finely granular | Record artifact | such as sections and even section headers.

6.2.3.20 Core metadata concept

Subtypes of | Core metadata concept | provide structural information required to support *International Release* data. This supporting information includes sets of enumerated values that apply to attributes of *concepts*, *descriptions* and *relationships*.

6.2.3.21 Foundation metadata concept

Subtypes of the | Foundation metadata concept | provide supporting metadata and structural information for *derivative* release structures including *Reference Sets*.

6.2.3.22 Linkage concept

Linkage *concept* codes are intended to link two or more other codes to each other to express compositional meanings. All *concept* codes that can be used as a *Relationship Type* are included under | Linkage concept |. The ones approved for use are the *Concept Model Attributes*. Implementation guidance is as yet quite limited for the other Linkage *concept* codes. Use of them should be regarded as non-standard, tentative and experimental, requiring extra care.

| Linkage concept | is a subclass of | SNOMED CT model component |, and the | Linkage concept | *hierarchy* contains the sub-hierarchies:

- | Link assertion |;
- | Attribute |.

Note: In RF1, | Linkage concept | was a top level hierarchy.

6.2.3.22.1 Link assertion

The Link assertion *sub-hierarchy* enables the use of *SNOMED CT concepts* in *HL7* statements that assert *relationships* between statements. Currently this content supports the *UK NHS* Connecting for

Health requirements for encoding of Statement *relationships* for the implementation of *HL7* Version 3 messaging in the UK *realm*.

Examples of Link assertion concepts:

- | Has reason | ;
- | Has explanation | .

6.2.3.22.2 Attribute

Concepts that descend from this *sub-hierarchy* are used to construct *relationships* between two *SNOMED CT concepts*, since they indicate the *relationship type* between those *concepts*. Some attributes (*relationship types*) can be used to logically define a *concept* (defining attributes). This *sub-hierarchy* also includes non-defining attributes or attributes that may be useful to model *concept* definitions but which have not yet been used in modeling *precoordinated concepts* in *SNOMED CT*.

Examples of Defining attributes:

- | is a | .
- | Concept model attribute |:
  - | Laterality | ;
  - | Procedure site | ;
  - | Finding site | ;
  - | Associated morphology | .

Examples of Non-defining attributes:

- | Unapproved attribute | :
  - | Relieved by | ;
  - | Has assessment | .

6.2.4 Terming and Naming Conventions

This Part of the Editorial Guide provides chapters that explain and specify the conventions for construction and formation of terms and names, including word order, spelling, punctuation, verb tense, abbreviations, acronyms, and related issues.

6.2.4.1 Introduction to Terms and Descriptions

Terms are character strings that consist of words, phrases and other human-readable representations that convey the meanings of *concepts*. A term in connection to a particular *concept* is called a *Description*. *Descriptions* may be of several different types. Refer to the *SNOMED CT* Technical Implementation Guide for specific background on the different types of *Descriptions* that are recognized in the *International Release*. This document refers to two commonly used *Description* types, *Fully Specified Name* (FSN) and *Synonym*, and it also refers to *Preferred Terms*. *Preferred Term* is not a *Description* types but is the name given to a *Synonym* which is marked as *preferred* in a particular language or dialect.

6.2.4.1.1 Fully Specified Name

Each *concept* has at lease one *Fully Specified Name* (FSN) intended to provide an unambiguous way to name a *concept*. The purpose of the FSN is to uniquely describe a *concept* and clarify its meaning. The FSN is not a commonly used term or natural phrase and would not be expected to appear in the human-readable representation of a clinical record.

A *concept* may have more than one FSN, but only one of these may be marked as *preferred* in a given language. A Language *Reference Set* is used to specify which FSN *descriptions* is *preferred* in each language or *dialects*. The original *fully specified name* (the first FSN created for a *concept*) is the ultimate source of reference, if FSNs in different languages have conflicting meanings. Most original FSNs are in US English and, as many translators choose not translate FSNs, the original FSN is preferred by default.

Note: The term in each FSN is unique across the entire active content of a *SNOMED CT release*.

Each FSN term ends with a “semantic tag” in parentheses. The semantic tag indicates the semantic category to which the *concept* belongs (e.g. clinical finding, disorder, procedure, organism, person, etc.). The “semantic tag” helps to disambiguate different *concepts* which may be referred to by the same commonly used word or phrase.

Example: I Hematoma (morphologic abnormality) I is the FSN of the *concept* that represents the “hematoma” that a pathologist sees at the tissue level. In contrast, I Hematoma (disorder) I is the FSN of the *concept* that represents the clinical diagnosis that a clinician makes when they decide that a person has a “hematoma”.

6.2.4.1.2 Synonym

A *synonym* represents a *term*, other than the FSN, that can be used to represent a *concept* in a particular language or *dialect*.

Each *concept* one or more *descriptions* of type synonym in each language. A *description* of type *synonym* contains a term that represents a word or phrase, other than the term in the *fully specified name* that can be used to represent a *concept*. One synonym for each *concept* is marked as *preferred* in each *dialect* and the associated term is called the *preferred term* for that *concept*.

The use of a *description* can vary between different languages, *dialects* and contexts, so a *description* may be *preferred* in some *dialects*, *acceptable* for use in other *dialects* and may not used in some *dialects*. A Language *Reference Set* is used to specify the *descriptions* that are *acceptable* or *preferred* in each language or *dialect*.

Example: *Synonyms* of the *concept* 22298006 I myocardial infarction (disorder) I in English include:

- I cardiac infarction I (*Description*.id: 37442013);
- I heart attack I (*Description*.id: 37443015);
- I infarction of heart I (*Description*.id: 37441018);
- I myocardial infarction I (*Description*.id: 37436014).

The *synonym* I myocardial infarction I (*Description*.id: 37436014) is marked as *preferred* in the US English Language *Reference Set*. Thus in US English this is the *preferred term*.

Note: Unlike *fully specified names*, synonyms are not required to be unique.

6.2.4.1.3 Preferred Term

The *preferred term* is the preferred common word or phrase used by clinicians to name that *concept* in a particular language, *dialect* or context. Each *concept* has one to more *descriptions* of type *synonym* in each language. In each language or *dialect* one of these *description* is marked as *preferred* and is the *preferred term* for that *concept*.

The use of a *description* can vary between different languages, *dialects* and contexts, so a *description* may be *preferred* in some *dialects*, *acceptable* for use in other *dialects* and may not used in some *dialects*. A Language *Reference Set* is used to specify the *descriptions* that are *acceptable* or *preferred* in each language or dialect.

Example: The *concept* 54987000 I repair of common bile duct (procedure) I has a *description* of type *synonym* I choledochoplasty I. This is marked as *preferred* in the US English Language *Reference Set*. Therefore, I choledochoplasty I is the *preferred term* for this *concept* in US English.

Note: Unlike the *fully specified name* (FSN) the *preferred terms* need not be unique. Occasionally, the *preferred term* for one *concept* may also be a *synonym* for a different *concept*. Interpretation in these cases will depend on context of use.

Example:

- | Cold sensation quality (qualifier value) | has a *preferred term* of “Cold”;
- | Common cold (disorder) | also has a *synonym* of “Cold”.

In both cases, “cold” represents a common clinical phrase used to capture the meaning of the *concept*.

Note: Selection of one term over another as "preferred" in a given language *dialect* depends entirely on whose preferences are being expressed. Different users are likely to have different preferences, and implementers are encouraged to select terms that properly represent the *concept* and meet the preferences of users. There is no expectation that the *preferred term* distributed with a given language *dialect* will meet all use cases; nor is there anything sacrosanct about the term. The US English *preferred term* has no special status relative to other terms. Rather, it is merely one term that properly represents the *concept* and can be used as a starting point.

6.2.4.2 General criteria - all term types, all hierarchies

6.2.4.2.1 Plurals

In general, terms are represented in the singular rather than the plural.

For example:

- | breast procedure | instead of | Breasts procedure |
- | Disorder of lung | instead of | Disorder of lungs |
- | Adrenal imaging | instead of | Adrenals imaging |

6.2.4.2.1.1 Exceptions

Organizational nodes, also called "grouper" *concepts*, may have a *synonym* that is plural. There is a special term type for these plurals.

For example:

- | Procedures for splenic lesions |
- | Diseases of mitral and aortic valves |

Fully-specified names should not be given in plural form unless the *concept* necessarily involves multiples. Unintended plurals might mislead data analysts into incorrectly inferring there were multiples when there was in fact only one.

An unintended plural is the use of a plural in a name for a code that might be attached to a case where there is only one of the entity being coded.

For example, consider "trochlear lesion" versus "trochlear lesions": Since users will want to use this code to refer to a single trochlear lesion, the singular form of "lesion" is correct, and the plural form would be incorrect.

As another example, consider "multiple cranial nerve palsies": In this case, the word "multiple" indicates that there can never be just one, so a plural form of "palsies" is correct, and the singular would be incorrect.

Note that these rules apply to the FSN, and should apply to preferred display names to be used in coding. Exceptions to these rules might be allowed for special synonyms used for navigation, where the broad category is more naturally named using a plural. However, it would be advisable to keep track of these exceptions in a separate subset or using a special term type so that they can be excluded when the singular/plural distinction is important for coding.

6.2.4.2.2 Punctuation

6.2.4.2.2.1 Apostrophes

Eponymous terms should ideally not include an apostrophe or final "s" (unless the name normally ends in "s"). With rare exceptions, *concepts* with any eponymous terms should have at least one term that follows this rule.

For example:

- | Down syndrome |
- | Sjogren syndrome |
- | Meigs syndrome |

On the other hand, in common usage the preferred name frequently does include the apostrophe "s". Where common usage requires it, there should be at least one term that has the apostrophe “s”. Existing eponymous terms with the possessive "s" but no apostrophe need not be retired, but newly added terms should either have no "s", or else include the apostrophe. For terms with a possessive apostrophe where the name normally ends in "s", the apostrophe should of course follow the "s". There should be no hyphen in between the two words.

For example:

- | Alzheimer's disease |
- | Bowen's disease |
- | Reiter's disease |
- | Meigs' syndrome |

6.2.4.2.2.2 Special characters

The special characters &, %, $, @, # are not permitted in FSNs. All instances of FSNs containing any of these characters need to be spelled out in full text. E.g. "FD&C Yellow #2" should be "FD and C Yellow Number Two".

The characters @ and $ are not used in any active term, regardless of term type, and new descriptions containing these characters should be avoided.

The characters &, %, and # are legitimately used in some synonyms including preferred terms.

6.2.4.2.2.3 Hyphens and dashes

A *hyphen* is a punctuation mark used to join words and to separate syllables. There should be no spaces either before or after the hyphen. Hyphens should follow rules of style for the *dialect* and language in which the terms are used.

For example:

- | intra-articular |
- | Zollinger-Ellison syndrome |
- | Zellweger's-like syndrome |
- | tick-borne hemorrhagic fever |
- | phospho-2-dehydro-3-deoxygluconate aldolase |

A *dash* is also a punctuation mark, but is used differently from a hyphen. It may be used to separate two phrases or names, to contrast values, to show a *Relationship* between two things, or to separate ranges of values.

Dashes (as opposed to hyphens) should not be used in FSNs (with rare exceptions) because they may obscure the exact meaning of the term. The dash should be replaced with words that clarify the meaning.

Table 184: Examples of dashes that should not be used in FSNs, and a replacement phrase:

Not acceptable for FSN:	Use instead:
Y90 - Yttrium 90	Yttrium 90
Aeronautical engineer - feasibility studies	Aeronautical engineer involved in feasibility studies
On examination - breath urinose - uremic	on examination, breath urinose implying uremic subject

Not acceptable for FSN:	Use instead:
O/E - bowel sounds exaggerated	On examination, bowel sounds exaggerated
Disability - all limbs	Disability of all limbs

Exceptions to the use of dashes: Where there is a need to distinguish categories from more specific subtypes that have the same name, it is sometimes expedient to use a dash followed by the word "category". For example, "glioma - category" distinguishes the general category of all gliomas from those neoplasms that are called simply "glioma". The neoplasm specifically called "glioma" is one of several subtypes of the glioma category, and does not have the same meaning as the category itself. Many classifications distinguish the category from the subtype of the same name by using a plural for the category. For example, ChEBI has a category called "amphetamines" and a specific molecular entity called "amphetamine".

6.2.4.2.2.4 Colons

Colons should not be used in FSNs.

Exceptions: Colons are allowed in the FSNs of organisms, substances, or products where the colon is a proper part of the name. They are also allowed in ratios and in tumor stages.

For example:

- | Salmonella II 43:g,t:[1,5] (organism) |
- | lidocaine hydrochloride 1.5%/epinephrine 1:200,000 injection solution vial (product) |
- | pT3: tumor invades adventitia (esophagus) |

Colons are allowed in designations (non-FSN terms) in a variety of contexts. Some common examples of use are to separate acronyms from the rest of a name, and to separate a specimen from the finding identified in that specimen.

For example:

- | FH: Metabolic disorder |
- | H/O: breast problem |
- | Urine: red - blood |

6.2.4.2.2.5 Forward slashes

The forward slash should not be used in FSNs.

Exceptions: A forward slash may be used for representing units of measure, as required in the pharmaceutical products hierarchy, and in laboratory test results and units of measure hierarchies. They may also be used in the construct “and/or” in FSNs. There should be no space either before or after the slash.

For example:

- | Nitroglycerin 0.3mg/hr disc (product) |
- | Mesoridazine besylate 25mg/mL injection solution ampule (product) |
- | Ibuprofen 5% / Levomenthol 3% gel (product) |
- | Bone structure of head and/or neck (body structure) |

A forward slash may be allowed in designations (non-FSN terms) in a variety of contexts. Some common examples of use are in acronyms with findings, and as an abbreviation meaning “and/or" *concepts*.

For example:

- | O/E - abdominal mass palpated |

6.2.4.2.2.6 Plus signs (+)

For combination drug products, a "+" sign is allowed and should be used.

6.2.4.2.2.7 Caret symbols (^)

As a workaround for the lack of markup that would allow proper representation of superscript characters, a pair of caret symbols is used to enclose character strings that properly should be displayed as superscript.

For example:

- | Technetium Tc^99c^ medronate (substance) |
- | Blood group antigen Sd^a^ (substance) |

Representing exponents: In alignment with the Unified Code for Units of Measure (UCUM) guidance on the use of powers of ten, the single caret is used to represent exponents, i.e. "powers of". For example, "10^3" for the third power of ten is acceptable.

6.2.4.2.3 Eponyms

Eponyms are names that are derived from a proper name, usually the name of a person who discovered or described the thing originally. They are commonly found in a wide variety of names in health terminology, ranging across diverse areas such as anatomic structures, morphologic abnormalities, blood groups, diseases, findings, and procedures.

For example:

- | Rutherford Morrison's pouch |
- | vein of Galen |
- | Aschoff body |
- | Kell blood group |
- | Down syndrome |
- | Moro reflex |
- | Whipple procedure |

It is neither desirable nor indeed possible to completely avoid the use of eponyms in a health terminology. Nevertheless, FSNs should avoid including eponyms wherever possible in order to improve clarity of meaning and to facilitate translation to other languages. The full *Description* should be used as the FSN, and the eponymous term can be added as a synonym. For example, the FSN for “Moro reflex” should use the phrase “infant startle reflex.”

Exceptions are allowed when the full *Description* is exceptionally long and unwieldy. An example of allowed exception is “Hemi-Fontan operation (procedure).” This operation is defined as a "bidirectional Glenn shunt with end-to-side anastomosis of proximal superior vena cava to right pulmonary artery with isolation from right atrium". The resulting FSN would be too long and unwieldy, so the eponym is allowed in the FSN in this case. Such exceptions require careful attention to the possibility that the acronym’s meaning may change over time.

Exceptions are also allowed for *concepts* where the eponym is the only precise clinically relevant name available, and where an artificially constructed non-eponymous name would necessarily be vague or subject to significant misinterpretation. Examples include “Hodgkin lymphoma” and “Burkitt lymphoma.”

It is permitted and encouraged to include eponyms as designations (non-FSN terms) whenever they are understandable, reproducible and useful in a given context. For example, the *preferred term* for “infant startle reflex” may be “Moro reflex.”

6.2.4.2.4 Abbreviations

Abbreviations are prohibited in FSNs and synonyms, with specified exceptions. Where there are exceptions, the specific policy allowing the exceptions must be spelled out in detail in the Editorial Guide, and otherwise terms containing abbreviations are not to be accepted into the *International Release*.

6.2.4.2.4.1 Acronyms

Acronyms are an abbreviation formed from the initial letters of other words and pronounced as a word (e.g. ASCII, AIDS). Here we reiterate the prohibition of acronyms in *fully specified names* (see *Acronyms* *in FSNs* on page 396. Acronyms can be misinterpreted because they are not fully spelled out. It is a mistake to assume that everyone will know what an acronym means. Therefore acronyms may not be used in *fully specified names* when the fully spelled out name is available. An exception may be where a sequence of letters started as an acronym but has now become a word in its own right, understood without expansion to its original full form. A common example would be “laser”. Evidence that it is a word in its own right is that it is included in dictionaries in lower case, and the fully spelled-out meaning has become a trivia question. An example of an acronym that may *not* be included in an FSN is “CT” for “computed tomography”. While those involved in imaging and radiology may regard “CT” as a word (pronounced “see tee”), it does not pass the test of being unambiguous, of appearing in a dictionary in lower case, or of its component words being a trivia question.

Acronyms are however allowed in synonyms or preferred terms when accompanied by the full expansion of the abbreviation. Expansions should be enclosed in parenthesis to reduce the technical implementation burden when indexing and searching e.g. CT (Computed tomography).

6.2.4.2.5 Verbal nouns

Where possible, the form for the action word describing the clinical action within a FSN and *preferred term* is the verbal noun ending in "-ion" rather than "-ing".

For example:

- “incision” instead of “incising”l
- “destruction” instead of“destroying”

Exceptions:

When no other verbal form is available, the "-ing" form may be used. Also, when common usage sanctions the noun form, it may be used.

For example:

- “suturing” of tricuspid valve
- “cautery” of wart

6.2.4.2.6 Past tense verbal forms and sentence function types

Past tense verbal phrases must not be used to name procedures, since they invoke a temporal context that indicates that the procedure was done. Any existing terms containing past tense verbs should be moved to the situation hierarchy.

Terms for procedures should also be examined for the presence of a phrase that can be categorized as a sentence function type - i.e. imperative, declarative, interrogative, or exclamatory. These should be disallowed in procedure terms. A procedure term should be a noun phrase that names the procedure, and should not contain information that it was done, or is to be ordered, carried out, or planned.

For example:

- | Hand tendon ganglion excised | indicates the procedure was done, as a past tense declarative statement. This is a situation with explicit context, not a procedure.
- | Excision of ganglion of tendon sheath of hand | is a noun phrase giving the proper term for the procedure.

6.2.4.2.7 Language dialects

All *fully specified names* should be represented in the “US” *dialect*. When there is a difference in spelling or *dialect* between the US and UK, the *concept* should be given a US *preferred term* and/or synonym and a UK *preferred term* and/or synonym.

For example:

- | Tumor of endocrine pancreas (disorder) | - FSN
- | Tumor of endocrine pancreas | Preferred - GB *dialect*, Synonym - US *dialect*
- | Tumor of endocrine pancreas | Preferred - US *dialect*, Synonym - GB *dialect*

6.2.4.2.7.1 Reference sources for English spelling variants

For general principles and references, Wikipedia has a summary for editors that is good quality, addresses a large number of US-GB differences, provides reference sources, and appears consistent with what SNOMED CT users have requested over the years: *http://en.wikipedia.org/wiki/User:SpNeo/Spelling_Guide*. This guide should be the first point of reference. In addition, specific references for each dialect can be consulted.

For GB Medical English, references include:

- The BMJ: *www.bmj.com/about-bmj/resources-authors/house-style*
  - “foetus and fetus are both acceptable in English: the BMJ uses fetus"
- BMJ in turn references the following two sources:
  - Dorland’s Medical Dictionary as a preferred dictionary for medical terms
  - Chambers 21st Century Dictionary for general usage

Note: Oxford English Dictionary spelling (en-GB-oed) is different from British English (en-GB). A summary of the points of difference can be found at *http://en.wikipedia.org/wiki/Oxford_spelling*. In those cases where en-GB and en-GB-oed differ, SNOMED CT preferred terms in the British English dialect should follow the en-GB spelling and not en-GB-oed. Addition of an en-GB-oed term is allowed but not required; when added it should be marked as acceptable in the en-GB dialect (and in some cases it will also be either acceptable or preferred in the en-US dialect).

For US Medical English, references include:

- Stedman’s Medical Dictionary
- Merriam-Webster Online Dictionary
- AMA Manual of Style

6.2.4.2.7.2 Principles for selecting preferred spelling variants

1. SNOMED CT may include (or add) more than one description, each with a different spelling for a given term, if the references cited above provide evidence of acceptability in the dialect(s) for which they are being added.
2. For spelling of preferred terms in a dialect, where the reference sources provide multiple options, a judgment may need to be made about the most commonly adopted spelling. This can sometimes be determined by examining a few articles containing the word in question that are from highly cited journals such as BMJ (for British English) or NEJM or JAMA (for US English). If the term is a non-clinical term, appropriate scientific journals should be consulted, such as Science (published by a U.S. publisher) or Nature (published by a UK publisher).
3. The choice may be arbitrary. In difficult cases it is advisable to just make a choice and move on. It is unlikely that a strict rule covering all borderline cases can be found, and besides, preferences change over time.

Capitalization

Capitalize the first word in a term unless this would change its meaning. The rest of the string should be in lower case, except for proper nouns, adjectives derived from proper nouns, and acronyms.

For example:

- | Angiokeratoma of Fordyce |
- | Neonatal jaundice with Dubin-Johnson syndrome |
- | CMS supervisory note |
- | WBC enzyme determination |
- | Family Canidae |
- | pH measurement |

- | mm |

6.2.4.2.9 Articles

Terms should omit the unnecessary inclusion of articles like “an” and “the”.

For example:

- | Neoplasm of respiratory tract | instead of | Neoplasm of the respiratory tract |

6.2.4.2.10 Multiple meanings

A single term can refer to one or more than one meaning, and therefore the *fully specified name* should be examined in order to resolve any ambiguity.

For example:

- | immunosuppression | may mean the state of being immunosuppressed, or it may mean the procedure of applying immunosuppressive therapy.

6.2.4.2.11 Patient vs Subject

Terms containing the word "patient" should instead use the word "subject" which is more general. For example, "swab taken by patient" should instead be "swab taken by subject". The word "subject" here refers to the subject of the record, who may in some circumstances not actually be considered a patient per se.

6.2.4.3 General criteria - Fully Specified Names

These guidelines should be applied to new content, and the process of applying them to existing content will be carried out as time and resources permit, according to priorities for usage. There are many terms in existing content that are not in compiance with current guidance. This is partly a result of historical practice in the source terminologies, and in some cases a result of multiple competing rules for determining term composition or word order that cannot all be simultaneously satisfied.

It is unrealistic to expect the phrasing pattern or the word order of the *Fully Specified Name* to ever be able to satisfy the wide variety of sorting and display requirements. Word order consistency for the sake of display or sorting should be achieved by using descriptions other than the *Fully Specified Name*, and in many or most cases will require the creation of descriptions in an extension and maintained as reference sets designed for this purpose by the end user, vendor, or national release center.

6.2.4.3.1 Minor changes in the Fully Specified Name

This rule specifies that the *Concept* does not need to be retired in cases where the *Fully Specified Name* (FSN) undergoes minor changes. Minor changes in the FSN are those changes that do not alter its meaning. They may include changed capitalization, punctuation, spelling, acronym expansion or word order revision. Such changes are sometimes necessary in order to achieve a consistent and predictable presentation style, but are allowed only if they do not change the *Concept*'s meaning.

A change to the semantic type shown in parentheses at the end of the FSN may sometimes be considered a minor change if it occurs within a single top-level hierarchy (e.g. a change from a finding tag to a disorder tag, or a change from a procedure tag to a regime/therapy tag), but a move to a completely different top-level hierarchy is regarded as a significant change to the *Concept*'s meaning and is prohibited.

Unlike the *Concept*, the *Description* cannot remain unchanged even with minor changes to the term string. If *DescriptionType* is *Fully Specified Name* then any change to any character of the *term* string requires a new *Description* with a new unique identifier.

6.2.4.3.2 Unique

The FSN is unique among active *concept* codes.

6.2.4.3.3 Unambiguous

The FSN should provide a linguistic representation of the *concept* in an unambiguous way. It is considered an anchor for the representation of meaning of a *concept*, to which modelers can refer when assigning a logic-based definition. The FSN does not necessarily follow the usual phrasing used in clinical practice; it may be phrased differently and may be longer and more fully spelled out in order to represent the meaning as clearly as possible and globally communicate the intended meaning of the *concept*.

Ambiguity Examples:

The following FSNs are ambiguous because the *concept* clearly has additional meaning that is not spelled out by the FSN. These FSNs would have to be retired:

| Standing in water side toward (finding) |

Note: this FSN does not indicate which side of what is toward what.

| Lumbar ache - renal (finding) |

Note: this FSN does not clearly convey whether the lumbar ache is must be of a specifically renal etiology or whether the lumbar ache is merely located over the renal area

Examples of clear FSNs:

- | Benign neoplasm of clavicle (disorder) |
- | Excision of cyst of spleen (procedure) |

6.2.4.3.4 Hierarchy tag

The FSN must end with a hierarchy tag in parentheses that identifies the hierarchy into which the *concept* is placed via its *Relationships*.

Examples:

| Neoplasm of lung (disorder) |

Note: the '(disorder)' hierarchy tag indicates that this *concept* is in the | disorder | subhierarchy of the *clinical finding hierarchy*.

| Appendectomy (procedure) |

Note: The '(procedure)' hierarchy tag indicates that this *concept* is in the | procedure | hierarchy.

Sometimes the hierarchy tag is the only disambiguating factor for the meaning of the FSN.

Examples:

| Carcinomatosis (disorder) | versus | Carcinomatosis (morphologic abnormality) |

6.2.4.3.5 US English spelling and dialect for the International FSN

The FSN assigned to a *concept* when the *concept* is accepted for incorporation into the *International Release* is designated as the International FSN. The International FSN is considered the gold standard for interpretation of the meaning of the *concept*, from a linguistic standpoint. Obviously the logical definitions represented using the *concept model* should represent the same meaning. Spelling of the International FSN follows United States (American) English spelling conventions. Other English language spelling and conventions, such as Great Britain (UK) English, may be represented in *Preferred Terms* and Other Terms that are appropriately tagged using the language *reference set* mechanism. (In this document, US English *Descriptions* are preceded by US, and Great Britain English *Descriptions* are preceded by GB).

Examples:

FSN: Chronic anemia (disorder)

US PT: Chronic anemia

GB PT: Chronic anaemia

FSN: Ischemic heart disease (disorder)

US PT: Ischemic heart disease

GB PT: Ischaemic heart disease

6.2.4.3.6 FSN for procedures

When possible, the FSN for a procedure should name the action of the procedure (the METHOD) first, and then the object that the action acts directly upon. For some of the examples below, the modeling of the *concepts* is included to help the user understand the suggested naming conventions.

1. **When the direct object of the action is an anatomical site:** In this case the word(s) naming the site should follow the word(s) naming the action.

   Example:

   | Repair of artery (procedure) |

   In this case, the action is named by "repair" and the site is named by "artery".

2. **When the direct object of the action is a device:** In this case the word(s) naming the device should follow the word(s) naming the action. If there is a site that is not the direct object of the action, the word(s) naming it should come after the word(s) naming the device.

   Example:

   | Insertion of catheter into artery (procedure) |

   In this case, the action is named by "insertion," the direct object is named by "catheter," and the indirect site is named by "artery.".

3. **When the direct object of the action is a substance:**In this case, the word(s) that name the substance should follow the words that name the action. If there is a site that is not the direct object of the action, the word(s) naming it should follow the word(s) naming the substance.

   Example:

   | Injection of hormone into subcutaneous tissue (procedure) |

   In this case, "injection" is the action, "hormone" is the direct object, and "subcutaneous tissue" is the indirect site.

4. **When the direct object of the action is a morphologic abnormality:**In this case, the morphology term should follow the action term. If there is a site mentioned, it should follow the morphology term.

   Examples:

   | Excision of cyst of breast (procedure) |

   In this example, "excision" is the action, "cyst" is the morphology, and "breast" is the site. The "cyst" is the direct object of the action.

   | Operation on aneurysm of carotid artery (procedure) |

   Here the action is "operation," the direct object is the morphologic abnormality "aneurysm," and the site is "carotid artery."

6.2.4.3.6.1 Past tense verbal forms and sentence function types

Past tense verbal phrases must not be used to name procedures, since they invoke a temporal context that indicates that the procedure was done. Any existing terms containing past tense verbs should be moved to the situation hierarchy.

Terms for procedures should also be examined for the presence of a phrase that can be categorized as a sentence function type - i.e. imperative, declarative, interrogative, or exclamatory. These should be disallowed in procedure terms. A procedure term should be a noun phrase that names the procedure, and should not contain information that it was done, or is to be ordered, carried out, or planned.

For example:

- | Hand tendon ganglion excised | indicates the procedure was done, as a past tense declarative statement. This is a situation with explicit context, not a procedure.
- | Excision of ganglion of tendon sheath of hand | is a noun phrase giving the proper term for the procedure.

6.2.4.3.7 FSN for clinical findings

When possible, the FSN for a Clinical finding should name the morphologic abnormality before naming the site.

Examples:

I Inflammation of ampulla of Vater (disorder) I

Here the morphologic abnormality is inflammation and the site is “ampulla of Vater.”

I Edema of hand (finding) I

In this case “edema” names the morphologic abnormality and “hand” names the site.

6.2.4.3.8 Acronyms in FSNs

Acronyms are easily misinterpreted and therefore an FSN is not fully specified if an acronym is not spelled out in full. For this reason, all acronyms are banned from FSNs. In those cases where it is found that there is an acronym in an FSN, the FSN *DescriptionId* will be retired and a new FSN will be created. This is to occur whether the acronym was enclosed in parentheses with an expansion or not. In the replacment FSN term, the acronym will be entirely removed and replaced by its expansion. This does not necessarily require the retirement of the *ConceptId*, unless the meaning of the FSN had significant ambiguities before expanding the acronym.

Example:

I Computed tomography of chest (procedure) I instead of I CT of chest (procedure) I

6.2.4.4 General criteria

6.2.4.4.1 The Preferred Term

The *Preferred Term* (PT) represents a common word or phrase used by clinicians to name a *concept* in clinical practice or in the literature. Only one of the available *Descriptions* can be designated as preferred. Depending on the computer application, the PT may be the *Description* selected for display, although each application is assumed to be allowed to independently determine the most appropriate display term for a given situation.

Examples:

FSN:	I Pharyngectomy (procedure) I
PT:	I Pharyngectomy I
Synonym:	I Excision of pharynx I

Note: This *concept* existed prior to the naming convention guidelines in this document. According to these guidelines, if this *concept* did not exist and was to be proposed as a new addition, its FSN should be “Excision of pharynx (procedure)”. The choice of PT (“pharyngectomy” or “excision of pharynx”) depends on the preference of the submitter and/or the reviewers. Where there is no clear preference, or where preferences vary, the choice of PT is currently arbitrary and unsystematic.

FSN:	I Blepharitis (disorder) I
PT:	I Blepharitis I
Synonym:	I Inflammation of eyelid I

Note: According to naming conventions in this document, this *concept* would be given an FSN of “Inflammation of eyelid (disorder).” Either of the non-FSN *Descriptions* could be chosen as the PT.

FSN:	I Catheterization (procedure) I
PT:	I Catheterization I

Synonym:	| Insertion of catheter |

Note: Following the naming conventions, the FSN of this *concept* would be "Insertion of catheter (procedure)." Either of the non-FSN *Descriptions* could be chosen as the PT.

FSN:	| Fracture of coccyx (disorder) |
PT:	| Fracture of coccyx |

Note: In this case the PT is the same as the FSN without the hierarchy tag "(disorder)." The term "coccygeal fracture" would be an equally acceptable PT.

6.2.4.4.2 Synonyms

A synonym is a term with the same meaning as another term. Terms that are synonyms are said to be synonymous, and the state of being a synonym is called synonymy.

Addition of the same term as a synonym to more than one concept will not usually be acceptable, but of course some terms have more than one meaning. When a term can take more than one meaning, the term may be attached to multiple concepts without violating this rule. Where multiple concepts have an identical term as a synonym, a review will be undertaken to ascertain if the concepts involved are duplicates. Where the concepts are duplicates, one concept will be retired with a SAME_AS relationship to the concept it duplicated.

Where a term with a single meaning is associated with more than one concept (which are not duplicates), the term should be retained on only one of the concepts and retired on the other(s).

In some cases a general term may be acceptable as a synonym for two or more specific concepts, depending on the context of use to derive proper specificity (e.g. "fundus" in the context of obstetrics vs ophthalmology). It is expected that these occasions will be unusual, and should be highlighted as exceptions, to assist with proper usage. Editorial guidance for broader synonyms should be taken into account.

6.2.4.4.3 Narrower synonyms

If a synonym is more specific than the FSN this is an error, since it does not have the same meaning. *Concepts* with synonyms more specific than the FSN are considered ambiguous, since they could have the general meaning of the FSN or the specific meaning of the erroneously attached synonym.

Example:

FSN: | Removal of device (procedure) |

SYN: | Removal and replacement of prosthetic device |

6.2.4.4.4 Broader synonyms

When a synonym is more general than the FSN, and there is no common context in which it has the same meaning as the FSN, the *concept* should be retired as ambiguous.

Example:

FSN: | Sprain (morphologic abnormality) |

SYN: | Joint injury |

However, a more general synonym is considered valid when there is a context where the more general synonym has the same meaning as the FSN.

Example:

FSN: | Entire fundus uteri (body structure) |

SYN: | Fundus | in the context of obstetrics.

These valid broader synonym terms should be marked using the "degree of synonymy" field in the language *reference set*, with value "near synonymous (depending on context of use)".

6.2.4.5 General rules for naming conventions for specific hierarchies

Generally desired features of naming conventions:

1. The convention should be consistent and reproducible.
2. The convention should follow "natural" language when possible.
3. The meaning of phrases generated by the convention should be unambiguous to users and across linguistic barriers for translation purposes.
4. Conventions should not attempt to address needs for variations based on word order preferences (e.g. for searching or display). The creation of multiple word order variants for these purposes is considered outside the scope of the *International Release* of *SNOMED CT*.

6.2.4.6 Naming conventions for the disorder hierarchy

In the disorder hierarchy, the word 'disorder' in singular should be used. When the *concept* is a general grouper of disorders of a body system, body site, or other broad general category, the word 'disorder' should be used in preference to 'disease' for the FSN. This rule in favor of 'disorder' over 'disease' applies only to broad groupers, and is not applied at 'leaf' level. Plurals such as 'disorders' or 'diseases' may only be used in synonym terms for grouper *concepts*. Strings should avoid the article 'the' in the string.

Table 185: Examples of use of the words 'disorder' and 'disease' for general groupers:

At a general level, use:	Instead of:
Disorder of nose	Diseases of the nose
Disorder of reproductive system	Disease of reproductive system
Disorder of lower respiratory system	Disease of lower respiratory system

6.2.4.7 Naming conventions for measurement procedures

6.2.4.7.1 Convention for the FSN for measurement procedures:

General naming pattern: Action, Analyte, Specimen

- First component: Action

  This is consistent with the general rule followed for the FSN for new I procedure I *concepts*. The action (the value of I METHOD I) is named first when possible.

  - Modifier of the first component: Scale Method

    Scale Method refines the Action, and therefore will precede the action in the naming order. (Scale Method, Action)

  - Naming pattern: (Scale Method, Action), Analyte, Specimen

- Second component: Analyte

  This is also consistent with the general rule followed for FSNs for new procedure *concepts*. Where possible, the action (the METHOD) is named first followed by the object acted directly upon. Applying this convention to measurement procedures, the object being acted directly upon (measured) is the Analyte.

- Third component: Specimen

  - Modifier of third component: Timing

    Timing provides information about the specimen, and will therefore precede it in the naming order. (Timing, Specimen)

- Word order for additional properties

Additional properties such as ratio, concentration, percentage, and count will follow the action.

Measurements done by screening should be specified with “by screening method” added at the end of the term

Example:

Measurement of substance X in Y specimen by screening method

6.2.4.8 Naming conventions for clinical imaging procedures

Almost all imaging procedures can be unambiguously expressed in a number of ways and there is a balance to be struck between allowing flexibility in language use and efficient terminology maintenance so all variants for all modalities will not routinely be included in SNOMED CT – any over and above those optimal descriptions following must be justified explicitly.

As a minimum, procedures will ordinarily be expressed in terms of the modality and body site (represented as ‘X’ in the following illustrations).

There are a number of inconsistencies in existing content, but it is not required that these naming conventions necessarily be applied wholesale to that content. It is definitely intended for new content. Changes to existing content to conform to these guidlines will be considered based on value to users.

6.2.4.8.1 X-ray imaging

Two possible approaches have been discussed, and final resolution has not yet been reached.

Option 1:

Descriptions:

FSN: Radiography of X (procedure)

PT: Radiography of X

Option 2:

Descriptions:

FSN: X-ray of X (procedure)

PT: X-ray of X

Discussion:

There is a need to differentiate a grouper for all imaging procedures that utilize X-rays, from plain X-rays. The goal for this guidance is plain X-rays of a site. Given the apparent lack of consensus, the solution probably will involve a text definition and a naming convention specific to SNOMED CT, with advice to implementers.

6.2.4.8.2 Computed tomography

The convention of describing these as Computerized Axial Tomography (CAT) has ceased to be accurate because of the "axial" part of the phrase; several alternative techniques may be used to create images in multiple planes or axes.

Computed tomography descriptions will **not** routinely include:

- Computed tomography scan of X

  The word "scan" will not be systematically added in new descriptions, and should not be included in the preferred description. However, if a specific request is received to add a description with the word "scan", it will not be denied.

- Computerized axial tomography scan of X

  This term and its abbreviation "CAT scan" are considered obsolete, and will not be added as new descriptions.

To avoid retirement of existing computed tomography concepts, existing FSNs that do not follow this terming guide will not be changed. A batch change of the FSNs without retirement of the concepts would require Committee approval.

Descriptions:

FSN: Computed tomography of ***X*** (procedure)

PT: CT of ***X***

Synonym: Computed tomography of ***X***

- Computed tomography angiography

  Descriptions:

  FSN: Computed tomography angiography of ***X*** (procedure)

  PT: CT angiography of ***X***

  Synonym: CT angiogram of ***X***

  Synonym: Computed tomography angiography of ***X***
- Computed tomography arthrography

  Descriptions:

  FSN: Computed tomography arthrography of ***X*** (procedure)

  PT: CT arthrography of ***X***

  Synonym: CT arthrogram of ***X***

  Synonym: Computed tomography arthrography of ***X***
- Computed tomography venography

  Descriptions:

  FSN: Computed tomography venography of ***X*** (procedure)

  PT: CT venography of ***X***

  Synonym: CT venogram of ***X***

  Synonym: Computed tomography venography of ***X***

  Venography may simply be a timing phase of angiography. It is agreed that venography can be a useful term in the FSN and that there is a meaningful technique difference between simple angiography and purposeful venography.

6.2.4.8.3 Ultrasonography

Descriptions:

FSN: Ultrasonography of ***X*** (procedure)

PT: Ultrasonography of ***X***

Synonym: Ultrasound scan of ***X***

Synonym: Ultrasound of ***X***

- Doppler ultrasonography

  Descriptions:

  FSN: Doppler ultrasonography of ***X*** (procedure)

  PT: Doppler ultrasonography of ***X***

  Synonym: Doppler ultrasound scan of ***X***

Synonym: Doppler ultrasound of ***X***

- Obstetric ultrasonography

  Note - Obstetric ultrasound scans are more complex to describe and often much additional clinical information needs to be described. However in terms of the modality the same rules are applied as follows:

  Descriptions:

  FSN: Obstetric ultrasonography of ***X*** (procedure)

  PT: Obstetric ultrasonography of ***X***

  Synonym: Obstetric ultrasound scan of ***X***

  Synonym: Obstetric ultrasound of ***X***

6.2.4.8.4 Magnetic resonance imaging

It has been agreed that MRI and MR are exceptions to the rule that all abbreviations should be accompanied in a description by their expanded form in parentheses.

It was agreed that the use of near synonyms is acceptable for these procedures (i.e. angiogram/arteriogram).

Descriptions:

FSN: Magnetic resonance imaging of X (procedure)

PT: MRI of X

Synonym: Magnetic resonance imaging of X

- Magnetic resonance angiography

  Descriptions:

  FSN: Magnetic resonance angiography of ***X*** (procedure)

  PT: Magnetic resonance angiography of ***X***

  Synonym: Magnetic resonance angiogram of ***X***

  Synonym: Magnetic resonance angiography of ***X***

  Synonym: MR angiography of ***X***

- Magnetic resonance venography

  Descriptions:

  FSN: Magnetic resonance venography of ***X*** (procedure)

  PT: Magnetic resonance venography of ***X***

  Synonym: Magnetic resonance venogram of ***X***

  Synonym: Magnetic resonance venography of ***X***

  Synonym: MR venography of ***X***

- Magnetic resonance arthrography

  Descriptions:

  FSN: Magnetic resonance arthrography of ***X*** (procedure)

  PT: Magnetic resonance arthrography of ***X***

  Synonym: Magnetic resonance arthrogram of ***X***

  Synonym: Magnetic resonance arthrography of ***X***

  Synonym: MR arthrography of ***X***

6.2.4.8.5 Radionuclide imaging (Nuclear medicine)

Descriptions:

FSN: Radionuclide scan of ***X*** (procedure)

PT: Radionuclide scan of ***X***

Synonym: Radioisotope scan of ***X***

Note - Where it is important to represent a particular isotope and associated agents in the concept then the descriptions will be as follows:

Descriptions:

FSN: Radionuclide scan of ***X*** using **Y** (procedure)

PT: Radionuclide scan of ***X*** using **Y**

Synonym: Radioisotope scan of ***X*** using **Y**

6.2.4.8.6 Positron emission tomography

Descriptions:

FSN: Positron emission tomography of ***X*** (procedure)

PT: Positron emission tomography of ***X***

Synonym: PET of ***X***

6.2.4.8.7 Single photon emission computed tomography (SPECT)

Descriptions:

FSN: Single photon emission computed tomography of ***X*** (procedure)

PT: Single photon emission computed tomography of ***X***

Synonym: SPECT of ***X***

6.2.4.8.8 Fluoroscopy imaging

Note: It was agreed that the use of near synonyms was acceptable for these procedures (ie angiogram/arteriogram).

It was also agreed that the word fluoroscopic was unnecessary in any description other than the FSN.

Unresolved: <u>There was a suggestion that imaging intensifier procedure of X is more correct.</u>

Unresolved: <u>There was a suggestion that GI fluoroscopy needs route of administration and nature of contrast agent identifying. There might also need to be exception to the naming convention, eg barium enema (as opposed to fluoroscopy of colon with barium contrast medium).</u>

Descriptions:

FSN: Fluoroscopy of ***X*** (procedure)

PT: Fluoroscopy of ***X***

Note - Simple Fluoroscopy is the real time imaging, usually on TV monitors/image intensifiers, of a body part or system and is only rarely undertaken as an imaging process alone (without use of contrast or some interventional procedure), most commonly fluoroscopy is used to guide or direct the primary procedure or purpose. The convention in clinical practice is largely to ignore the fluoroscopic element and refer to a procedure entirely by the primary component, eg angiography, however this would be unacceptable in SNOMED CT where the imaging component must be explicitly described, at least in the FSN. The convention is to use the adjectival form rather than the noun.

- Fluoroscopic arteriography

  Descriptions:

FSN: Fluoroscopic arteriography of ***X*** (procedure)

PT: Arteriography of ***X***

Synonym: Arteriogram of ***X***

Synonym: Angiography of ***X***

Synonym: Angiogram of ***X***

- Fluoroscopic venography

  Descriptions:

  FSN: Fluoroscopic venography of ***X***

  PT: Venography of ***X***

  Synonym: Venogram of ***X***

- Fluoroscopic arthrography

  Descriptions:

  FSN: Fluoroscopic arthrography of ***X*** (procedure)

  PT: Fluoroscopic arthrography of ***X***

  Synonym: Fluoroscopic arthrogram of ***X***

- Dual energy X-ray photon absorptiometry

  Descriptions:

  FSN: Dual energy X-ray photon absorptiometry of **X** (procedure)

  PT: Dual energy X-ray photon absorptiometry of **X**

  Synonym: DXA of ***X***

  Synonym: DEXA of **X**

6.2.4.8.9 Multiple modality imaging

There are very few imaging procedures which could be truly considered to be multi-modality procedures; on most occasions two procedures can be considered to be conducted in parallel rather than as one. There are, however, a small number of instances where this is the case, and the images are produced by one piece of equipment by (possibly) a single operator, albeit with multiple imaging energies: these are PET/CT and SPECT/CT.

These will be represented as follows:

- Single photon emission computed tomography with computerized tomography

  Descriptions:

  FSN: Single photon emission computed tomography with computerized tomography of ***X*** (procedure)

  PT: Single photon emission computed tomography with computerized tomography of ***X***

  Synonym: SPECT CT of ***X***

- Positron emission tomography with computerized tomography

  Descriptions:

  FSN: Positron emission tomography with computerized tomography of ***X*** (procedure)

  PT: Positron emission tomography with computerized tomography of ***X*** Synonym: PET CT of ***X***

6.2.4.8.10 Imaging guided procedures

There are numerous procedures where the imaging component can be considered to be a supplemental or secondary technique to assist with the accomplishment of the primary goal.

The primary goal can be either the procedure itself or the imaging guidance, and this difference will be reflected in the pattern of term construction for the fully specified name. In a term that is constructed as "*Y (procedure)*using*DI Modality* guidance", the primary goal is the procedure; but in a term that is constructed as "*DI Modality* guidance for *Y (procedure)*", the primary goal is the guidance.

Where the primary goal is the procedure, and the imaging is secondarily used for guidance, the pattern for all modalities would be:

*Y (procedure)*using*DI Modality* guidance (for FSN)

A synonym should be created that follows the FSN pattern, and in addition a preferred term should be constructed according to the pattern:

DI Modality guided *Y (procedure)*

E.g.

"Biopsy of liver using computed tomography guidance (procedure)" is the FSN, and "CT guided biopsy of liver" is the preferred term.

Specifically, for each 'guidance' modality in turn, the patterns are:

- Computed tomography guided procedure

  Descriptions:

  FSN: ***Y*** using computed tomography guidance (procedure)

  PT: CT guided ***Y***

- Fluoroscopy guided procedure

  Descriptions:

  FSN: ***Y*** using *fluoroscopic* guidance (procedure)

  PT: Fluoroscopy guided ***Y***

  The term "Fluoroscopic ***Y***" will be interpreted as meaning "***Y*** using fluoroscopic guidance (procedure)". Procedures such as I Biopsy of wrist using fluoroscopic guidance (procedure) I will be subtypes of I Fluoroscopy (procedure) I.

- Magnetic resonance imaging guided procedure

  Descriptions:

  FSN: ***Y*** using magnetic resonance imaging guidance (procedure)

  PT: MRI guided ***Y*** (procedure)

- Ultrasonography guided procedure

  Descriptions:

  FSN: ***Y*** using ultrasonographic guidance (procedure)

  PT: Ultrasonography guided ***Y***

  Syn: Ultrasound guided ***Y***

- X-ray guided procedure

  Descriptions:

  FSN: ***Y*** using X-ray guidance (procedure)

  PT: X-ray guided ***Y***

6.2.4.8.11 Contrast for imaging

It is essential to express when contrast is part of a procedure and that descriptions are constructed consistently, for example:

Descriptions:

FSN: Computerized axial tomography of brain with radiopaque contrast (procedure)

PT: Computerized axial tomography of brain with radiopaque contrast

CT brain with contrast

Synonym: CT of brain with contrast

Exceptions to this rule include all fluoroscopic angiography and fluoroscopic guided angiography interventions.

Although vascular contrast and other contrast are regularly used in imaging procedures, it is agreed that there is no need to specify the vascular nature of contrast when this is used, e.g. CT brain with contrast is the accepted pattern, not CT brain with vascular contrast.

It is agreed that it is unnecessary to add the word 'media' to make the phrase 'contrast media'.

It is also agreed that the link word to associate the contrast use with the procedure would be 'with', not 'for' or anything similar.

There is a case for explicitly adding a term qualification when naming procedures that are explicitly to be performed without contrast. In the UK and Australia, it was reported that there are no procedures that specifiy "without contrast" pre-coordinated in the national subset and this has the benefit of being able to provide implementation guidance to that effect.

There is also a suggestion that additional term detail is required when it is necessary to know the more precise nature of contrast (e.g. iodinated, with various osmolalities, barium, gaseous).

6.2.4.8.12 Imaging adjustments for view, projection or technique

It can be important both clinically and from an administrative perspective that variations in technique from the norm are captured, not least so that images can be both acquired and interpreted correctly. Examples of modifications include:

- Axial
- Skyline
- Decubitus
- Weight-bearing
- Penetrated
- Stress views
- Soft tissue
- Pediatric

Though these exist as potential qualifying values within SNOMED CT, they are not presently allowable values for post-coordination of DI procedures.

6.2.4.9 Naming conventions for test observable entities

Naming conventions are described for the fully-specified name (FSN) for observable entities, and for naming evaluation procedures or observable entities that are submitted with names from the *LOINC* or *IFCC-IUPAC* NPU systems.

6.2.4.9.1 Convention for the FSN for test observable entities:

General naming pattern: Property, Toward, System

- First component: Property

  The property (the | PROPERTY TYPE | of the observable) is named first when possible.

  - Modifier of the first component: Scale Method

    Scale Method refines the Property, and therefore will precede the action in the naming order. (Scale Method, Property)
  - Naming pattern: (Scale Method, Property), Toward, System

- Second component: Toward

  Where possible, the property is named first followed by the entity that is the value of I TOWARDS I.
- Third component: System
  - Modifier of third component: Timing

    Timing provides information about the specimen, and will therefore precede it in the naming order. (Timing, System)

Measurements done by screening should be specified with “by screening method” added at the end of the term

Example:

Level of substance X in Y specimen by screening method

6.2.4.9.2 Guidance for content submitted using LOINC or IFCC-IUPAC names

Naming pattern for *LOINC* parts: (Scale Method, **Action**), **Analyte**, (Timing, **Specimen**)

Naming pattern for *IFCC-IUPAC* parts:

- Mandatory terms: System (similar to specimen), Component, Kind-of-property
- Order of terms does not seem to matter due to the multilingual origin and use.

  Example: Substance concentration of glucose in blood plasma

  Substance concentration (= Kind-of-property) of glucose (= Component) in blood plasma (= System)

Examples of *SNOMED CT* FSNs for content submitted as *LOINC* names. For the patterns and the resulting FSNs below, (a) is for the observable, and (b) is for the procedure:

1. Ethylene glycol:MCnc:Pt:Urine:Qn

LOINC name:	Ethylene glycol:MCnc:Pt:Urine:Qn
Long common name:	Ethylene glycol [Mass/volume] in Urine

(a) Pattern:	Property X of analyte Y in system Z (observable entity)
(a) FSN:	Mass concentration of ethylene glycol in urine (observable entity)

(b) Pattern:	Measurement of property X of analyte Y in specimen Z (procedure)
(b) FSN:	Measurement of mass concentration of ethylene glycol in urine specimen (procedure)

Comments:

Observables do not necessarily specify the type of specimen obtained for making an observation, but rather may only specify the “system”, in this case the body substance in which the property inheres. It is therefore the default pattern in the observables hierarchy not to mention specimens (they are mentioned when necessary of course). However, it has been the pattern in the procedure axis to name the specimen, and it may be argued that each measurement action necessarily takes place on a specimen. It is therefore the default pattern in the procedure hierarchy to mention specimens.

2. Hemoglobin F:ACnc:Pt:Amnio fld:Ord

LOINC name:	Hemoglobin F:ACnc:Pt:Amnio fld:Ord
Long common name:	Hemoglobin F [Presence] in Amniotic Fluid

(a) Pattern:	Ordinal level of analyte X in system Y (observable)
(a) FSN:	Ordinal level of hemoglobin F in amniotic fluid (observable)

(b) Pattern:	Ordinal measurement of analyte X in specimen Y (procedure)
(b) FSN:	Ordinal measurement of hemoglobin F in amniotic fluid specimen (procedure)

Comments:

Both *LOINC* and NPU use a property type of arbitrary concentration (ACnc) combined with a scale type of ordinal (Ord) to indicate tests that are reported as either the presence or absence of an entity. Absence/presence is an ordinal scale with only two levels (e.g. absent=0 and present=1), but the combination of ACnc with Ord also allows ordinal scales with more levels (e.g absent=0, small amount=1, large amount=2). As a result, we do not follow the *LOINC* long common name pattern of calling this [Presence], but instead call it an “ordinal level” (in the observable), and an “ordinal measurement” (in the procedure).

3. Tricyclic antidepressants:MCnc:Pt:Ser/Plas:Qn

LOINC name:	Tricyclic antidepressants:MCnc:Pt:Ser/Plas:Qn
Long common name:	Tricyclic antidepressants [Mass/volume] in Serum or Plasma

(a) Pattern:	Property of analyte in system (observable)
(a) FSN:	Mass concentration of tricyclic antidepressant in plasma (observable)

(b) Pattern:	Measurement of property of analyte in specimen (procedure)
(b) FSN:	Measurement of mass concentration of tricyclic antidepressants in serum or plasma specimen (procedure)

Comments:

The names of analytes are given in singular tense. The system is plasma when the specimen is serum or plasma.

4. Creatinine:MRat:24H:Urine:Qn

LOINC name:	Creatinine:MRat:24H:Urine:Qn
Long common name:	Creatinine [Mass/time] in 24 hour Urine

(a) Pattern:	Rate property X of process Y over duration Z period (observable)
(a) FSN:	Mass rate of creatinine excretion in urine over 24 hour period (observable)

(b) Pattern:	Measurement of rate property of process Y in duration Z specimen (procedure)
(b) FSN:	Measurement of mass rate of creatinine excretion in 24 hour urine specimen (procedure)

Comments:

Rates are given per unit time, and describe processes.

5. Creatinine:MCnc:XXX:Urine:Qn

LOINC name:	Creatinine:MCnc:XXX:Urine:Qn
Long common name:	Creatinine [Mass/volume] in unspecified time Urine

(a) Pattern:	Property X of analyte Y in specimen Z collected over a time period (observable)
(a) FSN:	Mass concentration of creatinine in urine specimen collected over a time period (observable)

(b) Pattern:	Measurement of property X of analyte Y in specimen Z collected over a time period (procedure)
(b) FSN:	Measurement of mass concentration of creatinine in urine specimen collected over a time period (procedure)

Comments:

Since the property is not a rate, the observable needs to mention the urine specimen and the time period of its collection.

6. Cortisol^6 AM specimen:MCnc:Pt:Ser/Plas:Qn

LOINC name:	Cortisol^6 AM specimen:MCnc:Pt:Ser/Plas:Qn
Long common name:	Cortisol [Mass/volume] in Serum or Plasma --6 AM specimen

(a) Pattern:	Property X of analyte Y in system Z at time W (observable)
(a) FSN:	Mass concentration of cortisol in plasma at 6 A.M. (observable)

(b) Pattern:	Measurement of property X of analyte Y in specimen Z obtained at time W (procedure)
(b) FSN:	Measurement of mass concentration of cortisol in serum or plasma specimen obtained at 6 A.M. (procedure)

7. Serum ascites albumin gradient

LOINC name:	Serum ascites albumin gradient
Long common name:	? not found in *LOINC*

(a) Pattern:	Analyte X property Y difference between system Z1 and system Z2 (observable)
(a) FSN:	Albumin mass concentration difference between serum and peritoneal fluid (observable)

(b) Pattern:	Measurement of analyte X property Y difference between system Z1 and system Z2 (procedure)
(b) FSN:	Measurement of albumin mass concentration difference between serum specimen and ascitic fluid specimen (procedure)

8. Apoliporotein A-I/Apolipoprotein B:MCrto:Pt:Ser/Plas:Qn

LOINC name:	Apoliporotein A-I/Apolipoprotein B:MCrto:Pt:Ser/Plas:Qn
Long common name:	Apolipoprotein A-I/Apolipoprotein B [Mass ratio] in Serum or Plasma

(a) Pattern:	Mass ratio of substance X to substance Y in specimen Z (observable)
(a) FSN:	Mass ratio of apolipoprotein A to apolipoprotein B in plasma (observable)

(b) Pattern:	Measurement of mass ratio of substance X to substance Y in specimen Z (procedure)
(b) FSN:	Measurement of mass ratio of apolipoprotein A to apolipoprotein B in serum or plasma specimen (procedure)

9. Amprenavir^peak:MCnc:Pt:Ser/Plas:Qn

LOINC name:	Amprenavir^peak:MCnc:Pt:Ser/Plas:Qn
Long common name:	Amprenavir [Mass/volume] in Serum or Plasma--peak

(a) Pattern:	Peak property of analyte X in system Y (observable)
(a) FSN:	Peak mass concentration of amprenavir in plasma (observable)

(b) Pattern:	Measurement of peak mass concentration of analyte X in specimen Y (procedure)
(b) FSN:	Measurement of peak mass concentration of amprenavir in serum or plasma specimen (procedure)

10. Amino acids:Imp:Pt:Urine:Nar

LOINC name:	Amino acids:Imp:Pt:Urine:Nar
Long common name:	Amino acids [interpretation] in Urine Narrative

(a) Pattern:	Pattern of analyte X in system Y (observable)
(a) FSN:	Pattern of amino acids in urine (observable)

(b) Pattern:	Interpretation of pattern of analyte X in specimen Y (procedure)
(b) FSN:	Interpretation of pattern of amino acids in urine (procedure)

Comments:

In this case the pattern necessarily involves multiple amino acids, so singular would not be correct. An observable that is modeled by *LOINC* with an impression property and a narrative scale type may need creative naming.

11. Lutropin^baseline:MCnc:Pt:Ser/Plas:Qn

LOINC name:	Lutropin^baseline:MCnc:Pt:Ser/Plas:Qn
Long common name:	Lutropin [Mass/volume] in Serum or Plasma --baseline

(a) Pattern:	Property X of analyte Y in baseline specimen Z (observable)
(a) FSN:	Mass concentration of lutropin in baseline plasma (observable)

(b) Pattern:	Measurement of property X of analyte Y in baseline specimen Z (procedure)
(b) FSN:	Measurement of mass concentration of lutropin in baseline serum or plasma specimen (procedure)

12. Erythrocyte Ab:ACnc:Pt:Ser:Qn

LOINC name:	Erythrocyte Ab:ACnc:Pt:Ser:Qn
Long common name:	Erythrocyte Ab [Units/volume] in Serum

(a) Pattern:	Property of analyte X in system Y (observable)
(a) FSN:	Concentration of erythrocyte antibody in plasma (observable)

(b) Pattern:	Measurement of concentration of analyte X in specimen Y (procedure)
(b) FSN:	Measurement of concentration of erythrocyte antibody in serum specimen (procedure)

Comments:

In the case of red cell antibodies, serum is routinely used, but the system is still plasma. When the *LOINC* property type is ACnc and the scale type is Qn, we use only the word “concentration”, rather than spelling out “arbitrary concentration”.

6.2.4.10 Naming conventions for techniques

Techniques will include the word "technique" in their FSN; for example, "Microbial culture technique (qualifier value)".

6.2.4.11 Naming conventions for substances

- **Ribosomal ribonucleic acid of X organism**

  FSN: |Ribosomal ribonucleic acid of X organism (substance)|

  Preferred Term (US/GB), with Initial letter case sensitive: |X organism rRNA|

  Synonyms: A synonym that matches FSN

  Parent concept: Most distal appropriate descendant of 118251005 | Microbial ribosomal ribonucleic acid (substance)

  Example:

  FSN: Ribosomal ribonucleic acid of Candida (substance)

  Preferred Term, with Initial letter case sensitive: Candida rRNA

  Synonym: Ribosomal ribonucleic acid of Candida

  Parent concept: Microbial ribosomal ribonucleic acid (substance)

- **Ribonucleic acid of X organism**

  FSN: |Ribonucleic acid of X organism (substance)|

  Preferred Term (US/GB), with Initial letter case sensitive: |X organism RNA|

  Synonyms: A synonym that matches FSN

  Parent concept: Most distal appropriate descendant of |118248003|Microbial ribonucleic acid (substance)

  Example:

  FSN: Ribonucleic acid of Norovirus genogroup I (substance)

  Preferred Term, with Initial letter case sensitive: Norovirus genogroup I RNA

  Parent concept: Ribonucleic acid of Norovirus (substance)

  Synonym: Ribonucleic acid of Norovirus genogroup I

- **Deoxyribonucleic acid of X organism**

  FSN: |Deoxyribonucleic acid of X organism (substance)|

  Preferred Term (US/GB), with Initial letter case sensitive: |X organism DNA|

  Synonyms: A synonym that matches FSN

  Parent concept: Most distal appropriate descendant of |118249006|Microbial deoxyribonucleic acid (substance)|

  Example:

  FSN: Deoxyribonucleic acid of Aspergillus terreus (substance)

  Preferred Term, with Initial letter case sensitive: Aspergillus terreus DNA

  Synonym: Deoxyribonucleic acid of Aspergillus terreus

  Parent concept: Deoxyribonucleic acid of Aspergillus (substance)
- **Antigen of X organism**

  FSN: |Antigen of X organism (substance)|

  Preferred Term (US/GB), with Initial letter case sensitive: |X organism Ag|

  Synonyms: A synonym that matches FSN

  Synonyms with Initial letter case sensitive: |X organism antigen|

  Parent concept: Most distal appropriate descendant of |116633006|Microbial antigen (substance)|
- **Antibody to X organism**

  FSN: |Antibody to X organism (substance)|

  Preferred Term (US/GB), with Initial letter case sensitive: |X organism Ab|

  Synonyms: A synonym that matches FSN

  Synonyms: Anti-X antibody

  Synonyms with Initial letter case sensitive: |X organism antibody|

  Parent concept: Most distal appropriate descendant of |116642004|Antimicrobial antibody (substance)|
- **Immunoglobulin G, M, A, E, D antibody to X organism**

  FSN: |Immunoglobulin G, M, A, E, D antibody to X organism (substance)|

  Preferred Term (US/GB), with Initial letter case sensitive: |X organism IgG, M, A, E, D |

  Synonyms: A synonym that matches FSN

  Synonyms: Anti-X organism IgG, M, A, E, D

  Parent concept: Most distal appropriate descendant of |399812006|Antibody class (substance)|

  Parent concept: |X species antibody (substance)| if in SNOMED CT
- Descriptions, including FSNs, Preferred terms, and/or Synonyms, in the Substance hierarchy should not contain the word "total". Terms containing the word "total" (e.g. "total cholesterol") should not be added to the Substance hierarchy. Existing terms were retired as part of the January 2015 SNOMED CT release due to the inability to differentiate the definition between the “base” substance and “total” substance (e.g. “cholesterol” versus “total cholesterol”) within the context of the Substance hierarchy as well as the inability to create an appropriate relationship between the “base” and “total” concepts.

6.2.4.11.1 Naming conventions for antivenom

Antivenin and antivenom concepts in the Substance Hierarchy should be descendants of | Antivenin (substance) |. No new sub-grouper concepts should be added until further notice. The value of and need for sub-grouper concepts will be evaluated as part of the Substance Hierarchy Redesign Project.

Use antivenom, not antivenin, for FSNs and Preferred Terms. Synonyms containing antivenin will not be created routinely, but may be created upon request. Existing concepts that are not consistent with this naming convention will be cleaned up as a batch at a later date.

FSNs should be based on the scientific name, if there is a one-to-one correspondence. Naming conventions for polyvalent antivenoms (effective against multiple organisms) will not comply with this naming convention, and will be evaluated on a case-by-case basis. Preferred Terms should be based on the common name. Synonyms based on the scientific name should be created in most cases.

Useful reference: Current version of “WHO Guidelines for the Production, Control and Regulation of Snake Antivenom Immunoglobulins”.

6.2.4.12 Naming conventions for pharmaceuticals and biologic products

6.2.4.12.1 Acronyms and Notations

Table 186: Table of Acronyms

Acronym	Explanation
BoSS	Basis Of Strength Substance
EBNF	Extended Backus-Naur
FSN	*Fully Specified Name*
ID	Identifier
INN	International Non-proprietary Name
mINN	Modified International Non proprietary Name
ME	Medicinal Entity
NPMP	Non Proprietary Medicinal Preparation
PT	*Preferred Term*
RF2	*SNOMED CT Release Format 2*
SNOMED CT®	*SNOMED Clinical Terms*

Notation:

The definitions are written using a notation for describing formal languages, called Extended Backus-Naur Form (EBNF). EBNF has been standardized by the ISO under the code ISO/IEC 14977:1996(E), and this document uses the following characters:

Character	Name	*Description*
;	Terminating character	This identifies the end of a rule (called a “production rule”).
[…]	Option	Encloses optional items
{ … }	Optional repetition	Encloses optional items that can be repeated zero or more times.
(…)	Arrangement in groups	Encloses items that need to be grouped together.
“ … ”	Double quotation	A terminal *expression* (i.e. characters that appear exactly as shown).

6.2.4.12.2 General Style Rules

This document deals with rules that are used to describe *concepts* within the *SNOMED CT® International release*. The *SNOMED CT® International release* does not describe branded products, differing container sizes of a medicine, unit dose types or medicines presented as multicomponent packs. These will be part of a national extension where required and it is seen as the responsibility of the National *Extension* to map to the correct *SNOMED CT® International release concept*.

Other than those attributes used in the model for medicinal products it is assumed that all other attributes will be retired to avoid confusion.

It is anticipated that the terms required to describe *concepts* for use in interface systems may need to be created at the national level as national *Preferred Terms* [1]; therefore use of the *International Release Preferred Term* may occur to give an indication of how a term may be described within the national extension or to be used to assist translation.

Not all *concepts* in the model are defined: for example *qualifier Relationships* and data representation *concepts*; these are assumed to have standard *SNOMED CT®* definitions.

The following rules apply to all names below:

- Names will not include commas
- “mL” – will include a uppercase L to be consistent with the abbreviation for liter “L”
- Representation of per one mL will be “/mL” rather than “/1mL”
- A space will be placed in front of an opening parenthesis and after a closing parenthesis (unless it is at the end of the product’s name), but not within parentheses e.g. aaaa (bbbb) cccc.
- Where a name includes a + symbol a space will be placed either side of the plus symbol e.g. ibuprofen + oxycodone (product)
- Ratio representations will use a colon for the FSN and the PT
- Where a name includes the word “pre-filled” change to prefilled
- All strength representations with microgram or nanogram as the unit of strength should be written out in full.
- The units used in strength representation should be singular e.g. unit, microgram, mg. mL
- All names which include the term product or preparation other than as part of the semantic tag will be retired and the new name will not include product or preparation. E.g. formaldehyde product (product) to formaldehyde (product); acetylcholine preparation (product) to acetylcholine (product)

- Capitalization will only exist when significant as defined in the section *Capitalization of Pharmaceuticals and Biologic Products* on page 420. E.g. ibuprofen + oxycodone not Ibuprofen + Oxycodone

[1] RF2 will no longer contain a *Description* type value " *Preferred Term*", only types of " *Fully Specified Name* " and "Synonym", Synonyms may be refined either to "Preferred" or "Acceptable" using the acceptability attribute within a language *reference set*. As a result of this change in RF2 the preference for particular *Descriptions* in a language or *dialect* but it will be represented using a *reference set*.

6.2.4.12.3 Medicinal Entity (ME) Editorial Rules

6.2.4.12.3.1 Medicinal Entity Definition

A Medicinal Entity (ME) is the abstract representation of the set of active ingredient(s) (devoid of strength and form), which when formulated as a medicinal product, is intended for use in the treatment of a patient by at least one member nation. i.e. a NPMP must exist somewhere internationally. Where a member nation requests the addition of an ME *concept* that is within the scope of the Pharmaceutical and Biological product hierarchy, as defined in the Boundary and Scope standard, it will be added to the *International release*.

The medicinal entity class is recursive: an instance medicinal entity can have a child instance of a related medicinal entity to support the Relationship between a "moiety" medicinal entity and a "moiety with modifier" medicinal entity (e.g. two medicinal entities representing the moiety and the moiety + modifier).

Note that the Medicinal Entity Name is derived from the active ingredient of substance *concepts*, with the following knowledge or rules incorporated:

· The active moiety or the moiety with modifier is specified.

· A Medicinal Entity defines a group of products as represented by a set of one or more NPMPs, which contain substances with the same combination of active moieties.

· Medicinal Entities *concepts* will have a *Relationship* to all of their active ingredients, using one or more "has intended active ingredient" *Relationship*(s).

· Multicomponent packs are out of scope of the *International release* and so Medicinal Entity *concepts* will not be created to represent the components of such packs. However it is noted that the *International Release* must represent the individual active components of these packs and Medicinal Entity *concepts* will be provided to support this.

· Where there is a Medicinal Entity that is a moiety with modifier a corresponding moiety *concept* will be created.

· Some circumstances (for example when it is a multi-ingredient product) require more than two levels to the hierarchy may be required

6.2.4.12.3.2 Medicinal Entity Descriptions

6.2.4.12.3.2.1 Medicinal Entity "Fully Specified Name" Definition

The default FSN of Medicinal Entities can be defined as follows:

ME FSN: = (Ingredient_Details) "(product)"

***Description* of Component**	**Definition**
Medicinal Entity FSN	(Ingredient_Details) " (product)"

***Description* of Component**	**Definition**
Ingredient_Details	Ingredient_Name { " + " Ingredient_Name } **1.** An Ingredient_Name is included for each ingredient in the given Medicinal Entity. **2.** Ingredient_Names are ordered alphabetically within each Medicinal Entity.
Ingredient_Name	Ingredient_Name {" + " Ingredient_Name} **1.** One Ingredient_Name is included for each 'ME has active Ingredient' *Relationship*. **2.** The Ingredient_Names are ordered alphabetically based on the *Preferred Term* where ME has active Ingredient. **3.** The name for different ingredients are separated by a " + ".
(product)	The semantic tag used in the FSN of all Medicinal Entity *concepts*.

6.2.4.12.3.2.2 Medicinal Entity "Fully Specified Name" Rules

Rule ID	***Description***
ME-FSN-1	All rules in FSN Definition and Rules apply. Capitalization rules apply, as defined in *Capitalization of Pharmaceuticals and Biologic Products* on page 420.
ME-FSN-2	The Medicinal Entity FSN will be derived from the INN if active ingredient is a "moiety" or mINN if active ingredient is a "moiety with modifier". Where an International name is not available from The World Health Organization, national pharmaceutical naming conventions should be used e.g. USAN or BAN to derive the Medicinal Entity FSN.
ME-FSN-3	The Medicinal Entity FSN will be derived from the set of active ingredient(s) that is the destination of one or more "has active ingredient" *Relationship* from ME to Substance *concepts*.
ME-FSN-4	The Medicinal Entity FSN will include all active ingredients for each multi-ingredient preparation.
ME-FSN-5	The FSN may be required to be greater than 255 characters.

6.2.4.12.3.2.3 Medicinal Entity "Preferred Term" Definition

The default *SNOMED CT® International Release Preferred Term* of a Medicinal Entity can be more *fully defined* as follows:

ME PT: = (Ingredient_Details)

***Description* Component**	**Definition**
Medicinal Entity PT	(Ingredient_Details)

***Description* Component**	**Definition**
Ingredient_Details	Ingredient_Name { " + " Ingredient_Name } **1.** An Ingredient_Name is included for each ingredient in the given Medicinal Entity. **2.** Ingredient_Names are ordered alphabetically within each Medicinal Entity.
Ingredient_Name	ME has active ingredient *Preferred Term* where ME has Ingredient.

6.2.4.12.3.2.4 Medicinal Entity "Preferred Term" Rules

Rule ID	***Description***
ME-PT-1	All rules defined in *Preferred Term* Definition and Rules apply. Capitalization rules apply, as defined in *Capitalization of Pharmaceuticals and Biologic Products* on page 420.
ME-PT-2	The Medicinal Entity PT will be derived from the INN if active ingredient is a "moiety", or mINN if active ingredient is a "moiety with modifier". Where an International name is not available from The World Health Organization, national pharmaceutical naming conventions should be used e.g. USAN or BAN to derive the Medicinal Entity PT.
ME-PT-3	The sequence of ingredients in the Medicinal Entity *Preferred Term* will, by default, be based on the alphabetic order of the ingredient names. EXCEPTION: currently no exceptions exist. NOTE *National releases* may choose a different order or term where local clinical practice justifies this, for example: ticarcillin + clavulanic acid.

6.2.4.12.3.2.5 Examples of Medicinal Entity Terms

Type of product	**ME *Fully Specified Name***	**ME *Preferred Term***
Single Ingredient- moiety	amoxicillin (product)	amoxicillin
Single Ingredient – moiety with modifier	haloperidol decanoate (product)	haloperidol decanoate
Multi-Ingredient – moiety	codeine + paracetamol (product)	codeine + paracetamol
Multi-Ingredient – "moiety" and moiety with modifier	codeine phosphate + paracetamol (product)	codeine phosphate + paracetamol

6.2.4.12.4 Non Proprietary Medicinal Preparation (NPMP) Editorial Rules

6.2.4.12.4.1 Non Proprietary Medicinal Preparation Definition

A Non Proprietary Medicinal Preparation (NPMP) is an abstract *concept* representing the properties of one or more clinically equivalent Proprietary Product Unit of Uses (from (existing or past) National extensions).

A NPMP is the abstract representation of the set of active ingredient(s) and their strength(s) and dose form

Where strength is a ratio (concentration) example 500 mg per unit dose for discrete dosage forms or 50 mg/mL for continuous dosage forms, strength is represented at a single unit level e.g. per mL or per g.

A new NPMP will be created for each different strength and dosage form of a licensed medicinal product. If an existing product has a change of ingredient, such that it does not conform to the ingredients of the original NPMP, then a new NPMP will be created for the new product.

6.2.4.12.4.2 Non Proprietary Medicinal Preparation Descriptions

6.2.4.12.4.2.1 Non Proprietary Medicinal Preparation "Fully Specified Name" Full Definition

The *Fully Specified Name* of a Non Proprietary Medicinal Preparation can be more *fully defined* as follows:

NPMP FSN: = (Ingredients_With_Strength) “ ” Form “(product)”

***Description* Component**	**Definition**
Non Proprietary Medicinal Preparation FSN	(Ingredients_With_Strength) “ ” Form “ (product)”
Ingredients_With_Strength	Ingredient_Strength {“ + ” Ingredient_Strength} **1.** One Ingredient_Strength is included for each ME 'has active ingredient' *Relationship* that exists on the parent ME. **2.** The Ingredient_Strengths are ordered alphabetically based on the *Preferred Term* of the NPMP’s reference BoSS (followed by the *Preferred Term* of the associated ME’s active ingredient).The ingredient-strength pairs are separated from each other by a “ + ”.
Ingredient_Strength	If the parent ME’s ‘has active ingredient’ *Relationship* has an associated ‘has reference BoSS’ *Relationship* on the NPMP: BoSS [“ (as ” Active Ingredient “) ”] “ ” Strength where BoSS is the PT of the Ingredient from the *Relationship* ‘NPMP has reference BoSS’ and Strength is the strength associated with this *Relationship*. The Active_Ingredient is the PT of the Ingredient from the *Relationship* ‘ME has active ingredient’ on the NPMP’s parent ME. The “ (as ” Active_Ingredient “) ” is only included where the BoSS is different from the Active_Ingredient. If the parent ME’s ‘has active ingredient’ *Relationship* is not grouped to any ‘has reference’ *Relationship* on the NPMP: Active Ingredient “ ” Strength where Active Ingredient is the PT of the Ingredient from the *Relationship* ‘ME has active ingredient’ on the NPMP’s parent ME.
Form	The dose formulation of the NPMP, defined in a non-proprietary way. This is the PT of the Form (F) from the *Relationship* ‘NPMP has dose form F’.

Description **Component**	**Definition**
(product)	The semantic tag used in the FSN of all Non Proprietary Medicinal Preparation *concepts*.

6.2.4.12.4.2.2 Non Proprietary Medicinal Preparation "Fully Specified Name" Rules

Rule ID	***Description***
NPMPFSN-1	All rules in FSN Definition and Rules apply. Capitalization rules apply, as defined in *Capitalization of Pharmaceuticals and Biologic Products* on page 420.
NPMPFSN-2	The Non Proprietary Medicinal Preparation will be derived from the "moiety" or "moiety with modifier" of the BoSS, as defined for NPMP FSN.
NPMPFSN-3	Strength *expression*: All NPMP FSN will include strength *expression* (if available). The strength *expression* general rules and application to specific medication forms is outlined in *General Strength Formats* on page 422. There are occasions when *expression* of strength is not appropriate. e.g. Aqueous cream. Exceptions list will be reviewed on a case by case basis. See *EXCEPTIONS that do not require a strength to be specified* on page 425.
NPMPFSN-4	The form will be expressed as a singular form, e.g. tablet, capsule

6.2.4.12.4.2.3 Non Proprietary Medicinal Preparation "Preferred Term" Definition

The default *SNOMED CT® International Release Preferred Term* of a *Non Proprietary Medicinal Preparation* can be defined as follows:

NPMP PT: = (Ingredients_With_Strength) " " Form

Description **Component**	**Definition**
Non Proprietary Medicinal Preparation PT	(Ingredients_With_Strength) " " Form
Ingredients_With_Strength	Ingredient_Strength {" + " Ingredient_Strength} **1.** One Ingredient_Strength is included for each 'ME has active ingredient' *Relationship* that exists on the parent ME. **2.** The Ingredient_Strengths are ordered alphabetically based on the *Preferred Term* of the NPMP's reference BoSS (followed by the *Preferred Term* of the associated ME's active ingredient). **3.** The ingredient-strength pairs are separated from each other by a " + ".

***Description* Component**	**Definition**
Ingredient_ Strength	If the parent ME's 'has active ingredient' has an associated strength then: BoSS [" (as " Active_Ingredient_Minus_BoSS ") "] " " Strength where BoSS is the PT of the Ingredient from the *Relationship* 'NPMP has reference BoSS' and Strength is the strength associated with this *Relationship*. The Active_Ingredient_Minus_BoSS is the PT of the Ingredient from the *Relationship* 'ME has active ingredient' on the NPMP's parent ME, except with the PT of the BoSS removed from this string (if it is present in full). The " (as " Active_Ingredient_Minus_BoSS ") " is only included where the BoSS is different to the Active_Ingredient. If the parent ME's 'has active ingredient' *Relationship* does not have an associated strength) then: Active Ingredient " " Strength where Active Ingredient is the PT of the Ingredient from the *Relationship* 'ME has active ingredient' on the NPMP's parent ME.
Form	The dose formulation of the NPMP, defined in a non-proprietary way. This is the PT of the Form (F) from the *Relationship* 'NPMP has dose form F'. NPMP has dose form.PT

6.2.4.12.4.2.4 Non Proprietary Medicinal Preparation "Preferred Term" Rules

Rule ID	***Description***
NPMP-PT-1	All rules defined in *Preferred Term* definition and rules apply. Capitalization rules apply, as defined in *Capitalization of Pharmaceuticals and Biologic Products* on page 420.
NPMP-PT-2	The Non Proprietary Medicinal Preparation will be derived from the "moiety" or "moiety with modifier" of the BoSS, as defined for NPMP PT.
NPMP-PT-3	Where the associated active ingredient is different from the BoSS (e.g. it may be a modification of the BoSS), then the name of the active ingredient (with the BoSS name removed if present in full) is included in brackets following the word "as" in the NPMP's *Preferred Term*. e.g. Erythromycin (as ethylsuccinate) 500 mg tablet e.g. Clavulanic Acid (as Potassium Clavulanate) 125 mg tablet e.g. Diclofenac Sodium (as Diclofenac Diethylammonium) 10 mg/g Gel
NPMP-PT-4	Strength *expression* All NPMP PT will include a strength *expression* (if available). The strength *expression* general rules and application to specific medication forms is outlined in *General Strength Formats* on page 422. There are occasions when *expression* of strength is not appropriate. e.g. Aqueous cream, coal tar, glucose powder. Exceptions list will be reviewed on a case by case basis. See *EXCEPTIONS that do not require a strength to be specified* on page 425.
NPMP-PT-5	The form will be expressed as a singular form, e.g. tablet, ampoule.

6.2.4.12.4.2.5 Examples of Non Proprietary Medicinal Preparation Terms

Type of Product	***Fully Specified Name***	***Preferred Term***
Single ingredient product - (moiety)	amoxicillin 500 mg capsule (product)	amoxicillin 500 mg capsule
Single ingredient product (moiety and moiety with modifier)	diclofenac 46.54 mg tablet (product) diclofenac sodium 50 mg tablet (product)	diclofenac 46.54 mg tablet diclofenac sodium 50 mg tablet
Multi-Ingredient (moiety and moiety with modifier as BoSS)	codeine 23.43 mg + paracetamol 500 mg tablet (product) codeine phosphate 30 mg + paracetamol 500 mg tablet (product)	codeine 23.43 mg + paracetamol 500 mg tablet codeine phosphate 30 mg + paracetamol 500 mg tablet
Single ingredient product (moiety and moiety with modifier)	fluphenazine 9.24 mg/mL solution for injection (product) fluphenazine decanoate 12.5 mg/mL solution for injection (product)	fluphenazine 9.24 mg/mL solution for injection fluphenazine decanoate 12.5 mg/mL solution for injection

Type of product	***Fully Specified Name***	***Preferred Term***
applications, creams and ointments, ear preparations, enemas, gels, eye preparations, intravenous infusions, injection solutions, lotions, mouthwashes, dusting powders, sachets	terbinafine 8.89 mg/g cream (product) terbinafine hydrochloride 10 mg/g cream (product)	terbinafine 8.89 mg/g cream terbinafine hydrochloride 10 mg/g cream
solid oral dose forms, sunscreens, inhalations, powders for injection, nasal drops, oral liquids, spray solutions and spray suspensions(moiety and moiety with modifier as BoSS)	diclofenac 46.54 mg tablet (product) diclofenac sodium 50 mg tablet (product)	diclofenac 46.54 mg tablet diclofenac sodium 50 mg tablet

6.2.4.12.5 Capitalization of Pharmaceuticals and Biologic Products

The first character of a *Description* will either be lowercase or an integer unless listed as exception.

6.2.4.12.5.1 Capitalization of amino acid isomers

The isomeric prefix D or L will be indicated using a capital letter. The name of the entity itself will be entirely in lower case if prefixed by anything other than a number. Where a name is broken up using descriptors the entity names are in lower case.

- N-acetyl
- 2-methyl
- N-acetyl-L-cysteine
- Note: L-lysine will only be lysine as D-Lysine is not available

6.2.4.12.6 Ingredient/Basis of Strength Substance Naming Conventions

Note: Ingredient and Basis of Strength Substance names will be derived from the INN and Martindale: The Complete Drug Reference unless listed as an exception.

Ingredient and Basis of Strength Substance that end in "ate" when available as a salt shall be changed so that the moiety is represented by ending in "ic acid" where appropriate so as to match the WHO International Non-proprietary Names.

Additional examples will be added when required.

Examples:

Name	INN Name
alendronate	alendronic acid
clodronate	clodronic acid
cromoglycate	cromoglycic acid
etidronate	etidronic acid
folate	folic acid
folinate	folinic acid
fusidate	fusidic acid
mycophenolate	mycophenolic acid
pamidronate	pamidronic acid
risedronate	risedronic acid
tiludronate	tiludronic acid
valproate	valproic acid

Ingredient and Basis of Strength Substance shall have the order of their name changed where necessary, so that the clinically significant part of the salt name is represented first.

Examples:

Name	*SNOMED CT® International Release* Name
calcium folinate	folinate calcium
disodium etidronate	etidronate disodium
disodium pamidronate	pamidronate disodium

Name	***SNOMED CT® International Release*** **Name**
potassium clavulanate	clavulanate potassium
sodium citrate	citrate sodium
disodium clodronate	clodronate disodium
sodium cromoglicate	cromoglicate sodium
sodium fusidate	fusidate sodium
sodium valproate	valproate sodium

6.2.4.12.6.1 Waters of Hydration

Waters of hydration shall be expressed for each ingredient where hydration is clinically significant. Where an ingredient or BoSS is found to be anhydrous, this shall not be expressed.

6.2.4.12.6.2 Insulins

The name for insulins will be modified to show the type of insulin for discussion:

- insulin aspart
- insulin aspart protamine
- insulin detemir
- insulin glargine
- insulin glulisine
- insulin lispro
- insulin lispro protamine
- isophane insulin bovine
- isophane insulin human
- neutral insulin bovine
- neutral insulin human

6.2.4.12.7 General Strength Formats

6.2.4.12.7.1 General Rules

Rule ID	*Description*
APP-STR-1	The strength units will be consistent with the Unit of Measure *reference set*
APP-STR-2	Note that any overage contained in the product to allow the formulated amount to be administered is not specified.

Rule ID	***Description***
APP-STR-3	In general, the strength of a BoSS should be expressed by a number between 1 and 999 metric units. That is, if the number of units is less than 1 (for example 0.5), the next lower unit level should be used (e.g. 500 microgram should be used in preference to 0.5 mg). If the value is equal to or greater than 1000 milligram ("mg"), convert to and display as gram ("g"). If the value is less than 1 milligram ("mg") convert to and display as microgram.If the value is equal to or greater than 1000 millilitre ("mL"), convert to and display as liter ("L"). Note: for large volume parenteral injections, irrigation solutions, haemodialysis and peritoneal dialysis solutions, display as millilitres ("mL") Where the strength unit of measure will vary for the actual ingredient, BoSS and base ingredient according to this rule, all units of measure for the substance will standardized according to the unit of measure for the BoSS.
APP-STR-4	A space will be inserted between the strength value and strength unit of measure.
APP-STR-5	The full term "unit" will be used rather than the abbreviated "U".
APP-STR-6	The percentage strength will be qualified with the appropriate w/w or w/v.

6.2.4.12.7.2 Strength Expression Rules

The rules for the *expression* of strength to be used for various dose forms is as follows.

For safety reasons, Member Nations may require items to have an alternate representation of the strength or dual representation of strength. This will be achieved by the use of a synonym contained in the *National release*.

Rule ID	***Description***
APP-STR-7	The strength is expressed per one each (e.g. x mg per one mL, not per < 1 or > 1 mL)
APP-STR-8	Unit dose form, solid and liquid: the strength is expressed per unit dose. The package type is not identifying for the *concept*, and should be managed by the *National release* if required to be identified. If there are different package types, the *concept* is the same. Unit dose means that each package contains the amount intended to be taken as a single measured dose. Medicinal products in a packages intended for single use but that are not completely administered as a dose (e.g. Minims and ampoules) are handled as continuous dose forms.
APP-STR-9	Continuous dose forms, (semi)solid and liquid: the strength is expressed per gram or per mL. The size of the package (vial 10 mL or 100 mL) is not defining for the *concept* and should be managed by the *National release*; If the package size of two products is different, but the reference BoSS, strength and dosage form are the same, then the *concept* associated with those two products is the same
APP-STR-10	Nasal/respiratory/sublingual products that are used per dose: the strength is expressed per dose.

Rule ID	***Description***
APP-STR-11	Where delivered dose and metered dose differ (e.g. for some metered dose inhalers) strength should be expressed as the metered dose.
APP-STR-12	Patches: the strength is expressed as the amount delivered per unit of time. The time duration over which the patch is intended to be used (e.g. 3 days or a week) is not defining for the *concept* and should be managed by the *National release*; if the Patch duration is different, the *concept* is the same.
APP-STR-13	Where strength is commonly expressed as a percentage, for example in case of certain eye drops and saline. This would be added as a synonym in the *SNOMED CT® International Release* with the FSN being defined as above.

Medication Form	**Rules**
Solid unit dose forms –tablets, capsules, pessaries, suppositories, urethral stick, lozenge, pastille, chewing gum, granules or powder in a package for unit dose	Strength is to be expressed as the amount per unit dose, for example:amoxicillin 500 mg capsulefentanyl 400 microgram lozengemesalazine 500 mg modified release granules per unit dose
Liquid unit dose forms – liquids in a package for unit dose	lactulose 10 g syrup per unit dosesalbutamol 2.5mg solution for nebulisation unit dose
Solid continuous dose forms – granules, powders	psyllium husk 535 mg/g powder
Semi solid continuous dose forms – creams, gels, ointments	aciclovir 50 mg/g cream Strength is to be expressed as amount per unit.
Liquid continuous dose forms	timolol 10 mg/mL eye drops xylometazoline 1 mg/mL nasal drops ipratropium bromide 500 microgram/mL solution for inhalation
Liquid parenteral dose forms	Will always be considered as continuous dose forms rather than unit doses where there is a known concentration per volume. e.g. furosemide 10 mg/mL solution for injection carboplatin 10 mg/mL solution for injection insulin 100 international units/mL solution for injection
Patches	estradiol 4.17 microgram/hour patch
Powder for suspension, oral	amoxicillin 50 mg/mL oral suspension
Powder for solution, parenteral	amoxicllin 500 mg powder for injection

Medication Form	**Rules**
Nasal/respiratory/sublingual sprays/powder – metered dose inhalers, pressurised inhalers, dry powders, nasal/sublingual sprays	salbutamol 100 microgram/dose powder for inhalation salbutamol 100 microgram/metered dose pressurised inhalation

6.2.4.12.7.3 EXCEPTIONS that do not require a strength to be specified

There are occasions when *expression* of strength is not appropriate. An exceptions list will be reviewed on a case by case basis.

Examples:

Aqueous cream

Calamine lotion

6.2.4.12.8 Representation of Dose Forms

These dose forms should be selected from a standard reference of dose forms. (An appropriate source to provide a defined list of pharmaceutical dose forms is still under discussion). The following rules will apply to display of dose forms in the *SNOMED CT International Release*.

Rule ID	***Description***
APP-FOR-1	The dose form of a *concept*.
APP-FOR-2	Products dispensed as powders for solution, powder for parenteral solution or powder for parenteral suspension: the dose form and strength is as the powder itself, because the final concentration is not known.
APP-FOR-3	A pack containing powder plus solvent is to be managed as a multicomponent pack and should be contained in the *National release*.
APP-FOR-4	For respiratory products, whether the presentation is a metered dose pressurised inhalation or a dry powder for inhalation, will be represented in the *International Release*, but the type of inhalator (breath-actuated, CFC free etc) will not be defined any more specifically than that in the *International release*. If a more granular representation is required it will be maintained in the appropriate *National release*.
APP-FOR-5	Powder for oral suspension that is intended to be dispensed in a reconstituted state of a standard concentration should be represented as oral suspensions.

6.2.4.12.8.1 Rules and examples relating to specific dose forms

6.2.4.12.8.1.1 Solid Oral Dose Forms

Syntactic *Normal Form*: **Name – Strength – Dose Form**

No.	Rule	Example	Further information
1	Modified release dose forms will be described as m/r in the *preferred term*	Diclofenac sodium 100mg m/r tablet	The expanded modified release is not used even in the longest term
2	For combination products the correct format will be: Drug A xmg/Drug B ymg form	Paracetamol 500mg /codeine phosphate 30mg tablet	
3	For combination products where an approved combination name is available the format follow that convention:	Co-Amilofruse 5mg/40mg tablet	If necessary this can be abbreviated to 5/40mg but the longest term should always contain the 5mg/40mg format
4	If the strength relates to a particular salt of the drug then the salt should be represented in the name field	Bosentan monohydrate 125mg tablet	Similarly, if the strength relates to the base drug then the salt is *not* represented in the name field
5	For specific packs where varying strengths of the same product are present, representation needs to occur in the strength field	Ropinirole 250mcg+500mcg+1000mcg tablet starter pack	

6.2.4.12.8.1.2 Liquid and Semi-solid dose forms

Syntactic *Normal Form*: **Name – Strength – Dose Form – unit**

No.	Rule	Example	Further information
1	Sugar-free is not expanded even in the longest term	Cimetidine 200mg/5mL s/f oral solution	
2	* Unit dose is represented in the case of single dose presentations	Calcium phosphate 3100mg+colecalciferol 20mcg powder for oral suspension sachet Morphine sulphate 10mg/5mL oral solution 5mL unit dose vial	This is not how we represent it currently

No.	Rule	Example	Further information
3	Further examples:	Cefalexin 250mg/5mL oral suspension Linezolid 100mg/5mL granules for oral suspension	Continuous solid unit dose is not represented in term

6.2.4.12.8.1.3 Parenteral dose forms

Syntactic *Normal Form*: **Name – Strength – Dose Form – unit**

No.	Rule	Example	Further information
1	Liquid parenteral dose forms will have strength quoted in the per unit volume format. The unit dose will represent the final volume of the smallest single entity of the product that can be physically handled	Frusemide 10mg/mL injection solution 2mL ampoule	The following examples would be incorrect: Frusemide 20mg/2mL injection solution ampoule Frusemide 20mg/2mL injection solution 2mL ampoule
2	For combination products the correct format is Drug A xmg/Drug B ymg form unit dose	Cilastatin 500mg/imipenem 500mg powder for infusion solution vial	
3	If the strength is the representation of base chemical present then that should be reflected in the term	Iodixanol 320mg(I)mg/mL injection solution	In this instance the 320mg relates to the equivalent amount of base Iodine and not the actual amount of Iodixanol
4	Insulin presentations are represented as follows	Insulin zinc suspension 100units/mL injection 10mL vial Insulin soluble human 100units/mL injection 3mL cartridge Insulin biphasic lispro 25/75 100units/mL injection 3mL prefilled pen	

No.	Rule	Example	Further information
5	Further examples:	Enoxaparin 100mg/mL injection solution 1mL prefilled syringe Adrenaline 1:1000 1mg/mL injection 1mL ampoule	

6.2.4.12.8.1.4 Cutaneous and transcutaneous dose forms

Syntactic *Normal Form*: **Name – Strength – Dose Form**

No.	Rule	Example	Further information
1	Strengths are added as a percentage or as xmg/g (mcg/g)	Calcipotriol 50mcg/g cream	
2	For combination products where no approved name exists, the separate elements will be expressed in order of decreasing quantity. Where element quantities are identical, element order will follow alphabetical convention	Calcipotriol 50mcg/g/Betamethasone 500mcg/g ointment	
3		Glyceryl trinitrate 15mg/24hours transdermal patch	What is the rule here — should we be representing them as xmg per 24 hours OR as the total amount of drug present in the reservoir
4	Further examples	Liquid paraffin 50%/white soft paraffin 50% ointment Alpha tocopheryl acetate 50iu/g cream	This is not how we currently represent this

6.2.4.12.8.1.5 Dose forms for Inhalation

Syntactic *Normal Form*: **Name – Strength – Dose Form – unit**

No.	Rule	Example	Further information
1	* the unit dose will be represented in the case of single-dose presentations	Ipratropium bromide 250micrograms/mL nebuliser solution 1mL unit dose vial	
2	Further examples:	Colistimethate sodium 1million iu powder for nebulisation Mometasone furoate 400mcg breath-actuated dry powder inhaler Salbutamol 100mcg inhaler	

6.2.4.12.8.1.6 Eye, Ear, Nose dose forms

Syntactic *Normal Form*: **Name – Strength – Dose Form – [unit dose]***

No.	Rule	Example	Further information
1	* In the case of single-use preparations, the unit dose will be expressed as part of the term	Tropicamide 1% eye drops 0.5mL unit dose vial	
2	Further examples:	Chloramphenicol 1% eye ointment Pilocarpine 20micrograms m/r ocular insert Naphazoline HCl 0.01%/hamamelis water 12.5% eye drops Homatropine hydrobromide 2% eye drops	This is not how we represent it currently

6.2.4.12.8.1.7 Rectal/Vaginal Dose Forms

Syntactic *Normal Form*: **Name – Strength – Dose Form – [unit dose]***

No.	Rule	Example	Further information
1	* In the case of single-use preparations, the unit dose will be expressed as part of the term	Phosphates Formula B long tube enema 128mL sachet	
2	Combination packs are represented as follows	Clotrimazole 100mg/1% pessaries+cream	How do we deal with this? Should it be represented as Clotrimazole 100mg pessaries+ !% cream
3	Further examples:	Hydrocortisone acetate 10% rectal foam Diclofenac sodium 100mg suppository Clotrimazole 200mg pessary	

6.2.4.12.9 Units of Measure

Editorial Policy: Units of Measure are used to quantify the value of strength of active ingredient.

The following rules will apply to display of units of measure in the *SNOMED CT International Release*.

Rule ID	Rule
APP-UOM-1	SI units will be used where appropriate at NPMP level
APP-UOM-2	If the value is equal to or greater than 1000 milligram (“mg”), convert to and display as gram (“g”). If the value is less than 1 milligram (“mg”) convert to and display as microgram. If the value is equal to or greater than 1000 millilitre (“mL”), convert to and display as liter (“L”). Note: for large volume parenterals injections, irrigation solutions, haemodialysis and peritoneal dialysis solutions display as millilitres (“mL”) If the value is less than 1 micromole do not convert. Where the strength unit of measure would vary for the moiety BoSS and moiety with modifier BoSS according to this rule, all units of measure for the BoSS will be standardized according to the unit of measure for the active BoSS. For example: Pharmaceutical ingredient strength: mometasone furoate 1 mg/g Moiety with modifier BoSS strength: mometasone furoate 1 mg/g Moiety BoSS strength: mometasone 0.82 mg/g (not mometasone 820 microgram/g)

The Units of Measure list:

Description	**UNIT/PROPORTION**
MASS	
kilogram	kg
kilogram /liter	kg/L
gram	g
gram/gram	g/g
gram/milliliter	g/mL
gram /liter	g/L
milligram	mg
milligram/metered dose	mg/metered dose
milligram/gram	mg/g
milligram/milligram	mg/mg
milligram/millilitre	mg/mL
milligram /liter	mg/L
milligram/24 hour	mg/24 hours
microgram	microgram
microgram/metered dose	microgram/metered dose
microgram/gram	microgram/g
microgram/millilitre	microgram/mL
microgram /liter	microgram/L
microgram/24 hour	microgram/24 hours
nanogram	nanogram
nanogram/gram	nanogram/g
nanogram/millilitre	nanogram/mL

Description	**UNIT/PROPORTION**
VOLUME	
liter	L
liter/liter	L/L
milliliter	mL
millilitre/millilitre	mL/mL
millilitre /liter	mL/L
millilitre/gram	mL/g
microlitre	microlitre
microlitre/gram	microlitre/g
microlitre/millilitre	microlitre/mL
microlitre /liter	microlitre/L
nanolitre	nanolitre
nanolitre/millilitre	nanolitre/mL
MOLECULAR EQUIVALENTS	
mole	mol
mole /liter	mol/L
millimole	mmol
millimole/millilitre	mmol/mL
millimole /liter	mmol/L
micromole	micromole
micromole/millilitre	micromole/mL
micromole /liter	micromole/L
BIOLOGICAL ACTIVITY	

Description	**UNIT/PROPORTION**
international unit	international unit
million international units	million international unit
international unit/millilitre	international unit/mL
international unit/gram	international unit/g
international unit/milligram	international unit/mg
international unit/microgram	international unit/microgram
allergy unit	allergy unit
antigen unit	antigen unit
anti-Xa international unit	anti-Xa international unit
D antigen unit	D antigen unit
Enzyme-Linked ImmunoSorbent Assay	ELISA unit
index of reactivity unit	IR
kallikrein inactivator unit	KI unit
Kyowa unit	Kyowa unit
million unit	million unit
pressor unit	pressor unit
protein nitrogen unit	protein nitrogen unit
titre unit	titre
unit	unit
MICROBIOLOGICAL CULTURES	
billion organisms unit	billion organisms
billion vibrios unit	billion vibrios
cell culture infectious dose 50% unit	CCID50 unit

Description	UNIT/PROPORTION
colony forming unit	colony forming unit
international opacity unit	international opacity units
million cell culture infectious dose 50% unit	million CCID50 unit
million colony forming unit	million colony forming units
million organisms unit	million organisms
mouse lethal dose 50% unit	mouse LD50 unit
plaque forming unit	PFU
thousand organisms unit	thousand organisms
tissue culture infectious dose 50% unit	TCID50 unit
colony forming unit	colony forming unit
international opacity unit	international opacity units
million cell culture infectious dose 50% unit	million CCID50 unit
million colony forming unit	million colony forming units
million organisms unit	million organisms
mouse lethal dose 50% unit	mouse LD50 unit
plaque forming unit	PFU
thousand organisms unit	thousand organisms
tissue culture infectious dose 50% unit	TCID50 unit
tuberculin unit	tuberculin unit

Additional units of measure will be added to this list as required.

6.2.4.12.10 Other Units of Measure

Table 187: Summary Table

Unit of measure	Definition
%v/v	Percentage Volume in Volume number of milliliters of solute in 100ml of product
%w/v	Percentage Weight in Volume number of grams of solute in 100ml of product
%w/w	Percentage Weight in Weight number of grams of solute in 100g of product
%v/w	Percentage Volume in Weight number of milliliters of solute in 100g of product
cm	Centimeter
Mu	mega units
Mu/mL	mega units/milliliter
m	Meter
mg/16 hours	milligram/16hours
mg/72 hours	milligram/72hours
mg/kg	milligram/kilogram
microgram/actuation	microgram/actuation
microgram/24 hours	
microgram/72 hours	
mL/kg	milliliter/kilogram
Molar	
mm	Millimeter
mM	millimolar
unit/gram	
unit/mg	
unit/mL	unit/milliliter

6.2.4.12.11 Definition for Clinically Distinct Complexes and Clinically Significant Modifications

A **clinically significant** modification is defined as being when “the salt representation of the base materially changes the therapeutic potency of the base, the duration of action of the base, the onset of action of the base, the pharmacological target of the base or the adverse reaction profile of the base, such that prescribing and administration decisions should, in the opinion of an appropriate expert body, be made at the level of the salt representation of the base”.

New additions to this section are expected to be made after further discussion.

6.2.4.12.11.1 Clinically distinct complexes

Clinically distinct complexes are composed of two or more moiety *concepts* each of is intended to produce a therapeutic effect upon administration. The Medicinal Entity name will consist of the full name for the complex.

Clinically distinct complexes include:

ranitidine bismuth citrate

sulfadiazine silver

hexamine hippurate

6.2.4.12.11.2 Discernible therapeutic differences to the moiety

A discernible therapeutic difference is defined as a modification to the moiety that materially changes the therapeutic potency of the moiety, the duration of action of the moiety, the onset of action of the moiety, the pharmacological target of the moiety or the adverse reaction profile of the moiety, such that prescribing and administration decisions should, in the opinion of an appropriate expert body, be made at the level of the modification to the moiety. The Medicinal Entity name will consist of the moiety name with modification, where it is deemed to be discernibly therapeutically different from the moiety.

Discernable therapeutic differences to the moiety may include modifications to the following moieties:

lipid formulations

liposomal formulations

For items that include discernible therapeutic differences to the moiety, the modification will follow the name of the substance. Where multiple modifications are present, the order will be determined on a case by case basis. For example:

doxorubicin, pegylated liposomal

It should be noted that there may be cases where more than levels required at both ME and NPMP to represent the moiety with each individual modification separately.

6.2.4.12.11.3 Enantiomers

Enantiomers will be represented only if the enantiomers of a racemic mixture have proven significantly different therapeutic potencies, duration of action, onset of action, pharmacological targets or adverse reaction profiles. If, in the opinion of an appropriate expert body, prescribing and administration decisions should be made at the level of the modification to enantiomer, the Medicinal Product will represent the active enantiomer of a racemic mixture. For example:

dexamphetamine

levobupivacaine

escitalopram

esomeprazole

6.2.4.12.11.4 Micronized and Macrocrystal Formulations

The Medicinal Product will be represented as such if it has been formulated in either a micronised form which has been proven to increase the bioavailability of the active ingredient or a macrocrystal form which has been proven to alter the bioavailability or adverse effect profile of the active ingredient.

For items that include clinically significant micronised formulations, the modification will follow the name of the substance. Where multiple modifications are present, the order will be determined on a case by case basis.

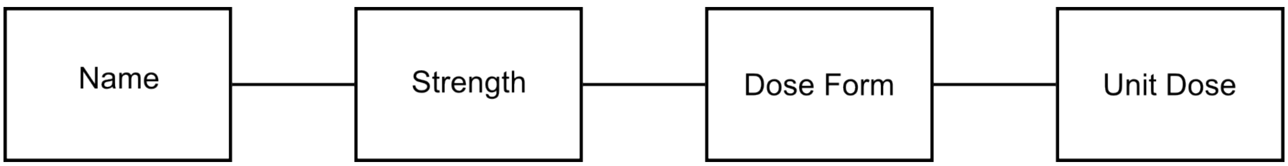
For example:

griseofulvin, micronised

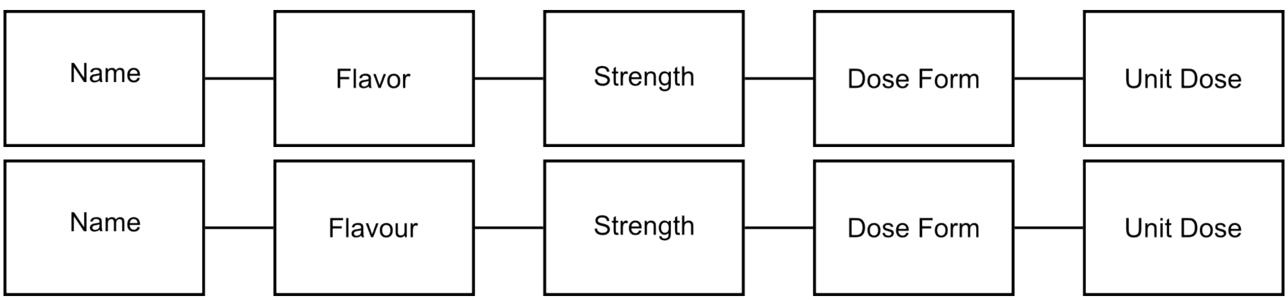
nitrofurantoin macrocrystals

6.2.4.12.12 Syntactic Normal Form

The syntactic *normal form* for a virtual medicinal product *Description* is:



The syntactic *normal form* for an actual medicinal product *Description* is:



The following sections treat

- Strength;
- Dose Form;
- Unit Dose; and
  Flavor
- Flavor

respectively.

6.2.4.12.12.1 Strength

The strength is that of the active ingredient, its value being quantified by see *Units of Measure*. Examples of the *expression* of strength are specified below:

***Concept* Types**	**Formulation Types**	**Examples of Strength Representation**
Solid oral dose forms	Tablets M/r capsules	50mg, 500mcg, 5/50
	Granules	5mg/5mL, 7g/sachet, 3.5g
	Oral powder	10mg, 750mg/sachet
	Dual ingredient tablets	50mg/25mg
	Medicated chewing gum	10mg

Concept **Types**	**Formulation Types**	**Examples of Strength Representation**
Liquid oral dose forms	Oral liquid	100mg/5mL, 500mcg/5mL, 250mg/250mg/5mL, 50mcg/mL, 62.5mg/5mL 125mg/1.25mL
	Oral drops	5mg/mL, 125mg/1.25mL
Topical liquids	Oromucosal spray	400mcg/spray, 0.15%
	Oromucosal gel	8.7%
	Mouthwash	0.15%
	Disinfectant solutions	0.05%, 1.5%/15%
Cutaneous dose forms	Cream, gel, ointment	2%, 0.1%/2%
	Cutaneous liquid	2%
	Cutaneous spray	0.5%
	Transdermal patch	10mg/24hrs
Parenteral dose forms	Solution for injection	10mg/mL, 100mcg/mL, 25,000iu/mL, 5000units/mL, 5%
	Solution for infusion	20%, 0.1%, 800mg/mL 1%(50mg/50mL)
	Powder for solution for injection	50mg, 1.5million iu 100,000iu, 30units, 500mg/500mg
Rectal/vaginal dose forms	Vaginal cream, Vaginal foam Rectal cream	0.92%, 2%, 25mg
	Vaginal ring	2mg(7.5mcg/24hrs)
	Rectal solution, vaginal solution	50%, 20mg/100mL, 10mg/2.5mL
	Suppository, pessary	4g, 5mg, 30mg
Eye/ear/nose dose forms	Eye gel, Eye ointment Eye drops, Ear drops	1%, 0.5%, 0.1%/0.5% 40mcg/mL
	Nasal cream Nasal spray solution	0.1%
	Ophthalmic insert	20mcg
Inhalation dose forms	Pressurised inhalation solution	100mcg, 100mcg/dose
	Inhalation powder capsule	200mcg
	Nebuliser solution	5mg/mL, 250mcg/mL
Miscellaneous	Radiopharmaceutical precursor	250MBq
	Kit for radiopharmaceutical preparation	74MBq/mL
	Irrigation solution	0.02%

6.2.4.12.12.2 Dose Form

The dose form is the pharmaceutical formulation of the product derived from the Summary of Product Characteristics or Label. These documents constitute the definition approved by the regulatory bodies of the European Pharmacopoeia Commission and the Food and Drug Administration (FDA) respectively.

They will be based on those dose forms expressed in the Standard terms for pharmaceutical dosage forms published by the European Pharmacopoeia Commission and the additional terms approved by the FDA defined within the HL7 messaging standard.

6.2.4.12.12.3 Unit Dose

The unit dose is the smallest single entity of the product that can be physically handled with the following defined deviations (see also *General Modeling Rules / Heuristics* on page 441 for further clarification).

Continuous liquids excluding eye-drops and topical liquids:	where the unit dose is expressed in terms of the accepted normal sub-unit used as the basis of administration
Continuous solids:	where the unit dose is expressed in terms of the accepted normal sub-unit used as the basis of administration
Continuous semi-solids eye-drops and topical liquids	where a consistent, physically measurable unit or sub-unit cannot be defined and which is therefore not instantiated

Where the unit dose and the dose form are duplicates e.g. tablet, the unit dose is not expressed again in the *Description*.

Flavor

Note: Flavor is only expressed when two or more clinically equivalent AMPs exist with different flavors.

6.2.4.12.13 Nomenclature

The preferred *Descriptions* used in the *SNOMED CT core* are those based on the rINN (Recommended International Non-Proprietary Name). The BAN (British Approved Name) should only be used in the UK extension, while the USAN (United States Adopted Name) should only be used in the USA extension.

6.2.4.12.14 Recommended International Non-proprietary Name (rINN)

The generic name given following the *World Health Organization* guidelines on naming. Does not imply recommendation of the use of the substance in medicine or pharmacy. There are different stages of the procedure: pINN = "proposed International Non-proprietary Name"/ rINN ="recommended International Non-proprietary Name.

As per our agreed recommendation we prefer to use Recommended International Non-Proprietary Names (rINNs) Within SCT Preferred *Descriptions*

- The relegation of British Approved Names (BANs) from preferred status to synonymous status where they differ from the INN list and the addition of rINNs as the preferred *Description*.
- The *concept* categories that require to be managed by this process have been identified by the Medicines Control Agency (MCA) in the UK.
- The relegation of United States Approved Names (USANs) from preferred status to synonymous status where they differ from the INN list and the addition of rINNs as the preferred *Description*.

6.2.4.12.15 United States Adopted Names (USAN)

A non-proprietary designation for any compound used as a drug, established by negotiation between the manufacturer of the compound and a nomenclature committee known as the USAN Council, which is sponsored jointly by the American Medical Association, the American Pharmaceutical Association, and The United States Pharmacopoeia Convention. A liaison representative of the United States Food and Drug Administration sits on the USAN Council. The term is currently limited to names adopted by the Council since June, 1961. These names will appear as the monograph titles in the official compendia, USP and NF, when and if the respective drugs are admitted to either compendium.[22]

6.2.4.12.16 British Approved Name (BAN)

An official non-proprietary name approved by the British Pharmacopoeia Commission.[23]

6.2.4.12.17 Specific pharmaceutical forms

The List of Standard Terms, as devised by the European Commission states clearly that the term *modified-release* is not sufficiently precise and instead further categorizes these into one of the following options: *Extended-release,Targeted-release, Prolonged-release, Delayed release, Gastro-resistant.* There is no exact representation of *e/c tablets* or capsules and these would be termed as gastro-resistant according to the European Commission.

Alternatively, the British Pharmacopoeia (BP 2002) includes the following definitions:

Modified release preparation: where the rate and/or place of release of the active substance(s) is different from that of a conventional-release dosage form, administered by the same route. This deliberate modification is achieved by a special formulation design and/or manufacturing method

The BP includes prolonged-release, delayed-release and pulsatile-release preparations as all belonging to the class of modified release preparations. It does not include gastro-resistant formulations in this *Description*.

Gastro-resistant capsules/tablets: are delayed release capsules that are intended to resist the gastric fluid and to release their active substance(s) in the intestinal fluid.

6.2.4.12.17.1 Editorial Rules

6.2.4.12.17.1.1 Modified release preparation

Where the rate and/or place of release of the active substance(s) is different from that of a conventional-release dosage form, administered by the same route. This deliberate modification is achieved by a special formulation design and/or manufacturing method

The BP includes prolonged-release, delayed-release and pulsatile-release preparations as all belonging to the class of modified release preparations. It does not include gastro-resistant formulations in this *Description*.

6.2.4.12.17.1.2 Gastro-resistant capsules/tablets

The *Description* enteric-coated (e/c) will be used to describe products that have been formulated to be gastro-resistant i.e. products that have a delayed release formulation that are intended to resist the gastric fluid and to release their active substance(s) in the intestinal fluid.

6.2.4.12.17.1.3 Intrathecal injections

From a clinical safety point of view there is arguably a case for using the pharmaceutical form - intrathecal injection. This is not in keeping with current practice, which dictates that clinical use of a product, and/or route of administration should not be intuitive in the term. However, special considerations apply to the formulation of intrathecal injections (as they do in fact to epidural injections also, which may also need to be specially considered). Any intra-spinal injection must be presented as a single-dose formulation and must not contain bactericides nor antoxidants (ref BPC 12th Ed pg 93). A complete list of parenteral preparations requiring specific consideration is given below:

Intra-arterial injections must not contain bactericides

Intracardiac injections must not contain bactericides

Intrathecal/Epidural injections must not contain bactericides/antoxidants

[**Intra-ocular injections**can be categorized as either**subconjunctival, intracameral, intravitreous** or **retrobulbar**. Whilst restrictions do not apply to subconjunctival injections, bactericides should be avoided in all other types of inte-ocular injections and the recommendation is that antoxidants should be avoided.](It's

[22] Source: *http://www.mercksource.com*
[23] Source: *http://www.mercksource.com*

better to follow the manufacturer's specific indication whether the product is only intended for particular route or depends upon the physician's assessment)

The use case scenario for representing certain parenterals in this manner is that Decision Support Systems would be able to highlight pharmaceutical forms specifically for certain routes of administration. One possible use would be to show a warning if a non-intrathecal preparation was prescribed for an intra-thecal procedure.

This method of representation needs further consideration.

6.2.4.12.18 General Modeling Rules / Heuristics

6.2.4.12.18.1 Plurality

All VMPs and AMPs will be conceptually described as singular entities i.e. tablet as opposed to tablets, capsule as opposed to capsules.

6.2.4.12.18.2 Decimal points

All *expressions* of strength will avoid using decimal *expressions* starting with the integer 0. That is:

500mcg instead of 0.5mg

500mL instead of 0.5L

However, decimal *expressions* commencing with an integer other than 0 will be used in preference to non-decimal greater than or equal to 1000.

i.e. 2.5mg will be used as opposed to 2500mcg

6.2.4.12.18.3 Additional descriptors

Flavors will be included between the name and the strength field of AMPs only.

e.g. Fybogel orange 3.5g/sachet granules

6.2.4.12.18.4 Unique terms

All *preferred terms* regardless of character length will be unique within the Pharmaceutical / biologic product hierarchy.

6.2.4.12.18.5 Units

The word units must appear in the *preferred term*. If the facility to distribute abbreviated *Descriptions* is delivered this will not be abbreviated to unit but to u.

In the *preferred term*, the number of units must be expanded fully:

1000units	not 1ku
9000units	not 9ku
10,000units	not 10ku
100,000units	not 100ku
10megaunits	not 10mu

However, if the facility to distribute abbreviated *Descriptions* is delivered these abbreviations will be used in the abbreviated *Description*.

There will be no spaces or dashes in units *Descriptions*.

i.e. megaunits not mega units, or mega-units

iu not i u, or i-u

There will be no spaces between the number and the units within the term.

i.e. 10units not 10 units.

6.2.4.12.18.6 Salts

Where appropriate to the *Description* the salt will always be written in full in the *preferred term*.
If the facility to distribute abbreviated *Descriptions* is delivered an abbreviation of the salt will be used in the abbreviated *Description*. The preferred means of abbreviating the salt will be to use the IUPAC convention where this is considered understandable by the average practising clinician.

Example:

Full salt	– hydrochloride
Abbreviation	– HCl

6.2.4.12.18.7 Unit Dose Representation

The table in this section identifies the applicability of unit dose representation.

The unit dose is the representation of the smallest physical entity of a product that can be physically handled.

Further work is required to identify how each of the following groups are accurately handled, in terms of authoring, with regard to *expression* of strength, pharmaceutical form and unit dose.

Dose form	**Theoretical Unit Dose (examples)**	**Unit Dose representation (Yes/No)**	**Notes**
Solid oral dose forms e.g. Tablets, Capsules	Tablet Capsule	**No**	In these instances the unit dose is the same as the pharmaceutical form i.e. one tablet or one capsule and is not duplicated within the term
Continuous solids e.g. granules	5g	**No** **Except for single-dose presentations**	In the case of Sennokot granules, the unit dose is 5g of granules administered from a 100g pot. As in the case of oral liquids (see below), the unit dose is therefore used as the basis of administration and strength representation and is not duplicated in the term. In the case of single-dose solids e.g. Ispaghula granules, the unit dose can be expressed as, for example, 3.5g sachet

Dose form	Theoretical Unit Dose (examples)	Unit Dose representation (Yes/No)	Notes
Oral liquids	5mL	**No** **Except for single-dose presentations**	The unit dose for oral liquids and other continuous liquids is expressed in terms of the sub-unit used as the basis of administration- generally 5mL. In theory, the oral liquid *concept* would be described as, for example, Ampicillin 250mg/5mL oral suspension 5mL but since the unit dose is expressed as part of the strength it is not duplicated within the term
Parenteral preparations (single-use presentations)	Vial 2mLAmpoule 1mLPrefilled syringe	**Yes**	In these instances the unit dose will be represented as the smallest physical entity that the product is available as, for example, *10mL ampoule*, *50mL vial, 1mL prefilled syringe*. The pharmaceutical form will be identified as *solution for injection*, *powder for solution for injection* etc
Parenteral preparations (multiple-use presentations)	20mL vial 50mL vial 10dose vial	**No**	Multidose vials need special consideration since, unlike single-use products, the smallest physical presentation of the product is not identical to the intended administration volume. For example Menjugate meningococcal vaccine is available in a 10-dose vial. The unit dose administered to the patient is 0.5mL reconstituted solution. In these instances, these products should theoretically be treated in a similar fashion to continuous liquids. However, more work needs to be done to verify how these are handled.

Dose form	Theoretical Unit Dose (examples)	Unit Dose representation (Yes/No)	Notes
Cutaneous preparations e.g. creams, sprays	1 spray 1 application of cream	**No**	In these instances, the unit dose is non-quantifiable and cannot be expressed
Inhalation preparations	1 puff 1 inhaled rotacap 1 actuation	**No** **Except for single-use presentations**	In the case of Salbutamol inhaler for example, the unit dose would be 1 actuation(puff); the strength is expressed as mcg per actuation (although in practice it is just the mcg that are expressed) so that the unit dose would be a duplication of the sub-unit used as the basis of administration. In the case of nebuliser solutions, these are often presented as single-dose units. In these cases, the unit dose would be expressed: Atrovent 250mcg/1mL nebuliser solution 1mL unit dose vial
Eye/ear/nose preparations	**1 drop**	**No** **Except for single-use presentations**	As in the case of cutaneous preparations, the unit dose is non-quantifiable and as such is not expressed. However, in the case of single-dose preparations such as Minims, the unit dose can be expressed as 0.5mL unit dose vial for example.

Dose form	Theoretical Unit Dose (examples)	Unit Dose representation (Yes/No)	Notes
Vaginal/ rectal preparations	**1 pessary** **1 suppositiry** **1 application of cream**	**No** **Except for *some* single-use presentations**	In the case of vaginal/rectal creams, these will be treated as in the case of cutaneous preparations (see above) where the unit dose is not quantifiable. In the case of suppositories and pessaries, these will be treated as in the case of solid oral dose forms where the pharmaceutical form is the same as the unit dose. In these instances, the unit dose will not be identified independently of the pharmaceutical form. In the case of single-dose enemas, the unit dose can be stated for example, *Phosphates Formula B long tube enema 128mL sachet*
Implants	**1 implant**	**No**	The unit dose and the pharmaceutical form are identical i.e. one implant, one implantation tablet etc.

6.2.4.12.18.8 Miscellaneous standard conventions

Expressions of volume will follow the SI naming convention.

i.e. mL not ml or ML, L not l.

All VMP *Descriptions* will begin with an uppercase character or integer. The remainder of the characters will be lower case unless part of an SI naming convention. AMP *Descriptions* will begin with an uppercase character or integer unless predetermined by the registered trade name as expressed in the Summary of Product Characteristics or Label. All other characters in the AMP name will be lower case part of an SI naming convention or predetermined by the registered trade name.

The *preferred term* for any drug name will use the Recommended International Non-proprietary Name. Domain specific naming conventions e.g. United States Approved Name and British Approved Name will be entered as synonyms with the facility to promote to preferred status within domain specific applications.

Multi ingredient products should be represented by a space+space sign in b/w two ingredients while avoiding a / or a :.

6.2.4.12.18.9 Products with dual represented strengths

Example: Tenecteplase 8000iu(40mg) powder and solvent for injection solution vial.

From author analysis of the Summary of Product Characteristics or Label it will be ascertained how the manufacturer expresses strength within the *Descriptions* used both in naming and advising on administration of the product. If these documents consistently use only one in *Description* and dosing information the *Description* will reflect only this. Where the documents uses both (as in the example cited above), either together or separately within different sections, both will be used.

6.2.4.12.18.10 Plastic ampoules

The fact that the ampoule is plastic is not clinically significant;.Hence as far as the VMP goes this will not be represented in the term. However, in the AMP, the plasticity of the ampoule is generally represented in the name.

6.2.4.12.18.11 Sterile-wrapped injections

In this case, the fact that the ampoules/vials are sterile wrapped is clinically significant in terms of usage for example, sterile-wrapped injections would be used in a sterile environment ie theater. As such, this feature needs to be represented in both the VMP and the AMP. The correct format will be sterile-wrapped with a hyphen.

6.2.4.12.18.12 Polyfusors

A VMP should not have any brand attributable information in its *Description* and therefore it is not permissible to use the term Polyfusor in the VMP. However, if these are termed bottles or bags the *Description* becomes ambiguous with the user automatically assuming glass bottles and PVC collapsible bags. The two key points about a Polyfusor are that they are semi-rigid and made of polyethylene. This latter point is of vital importance when it comes to issues such as latex allergy (where polyfusors form the infusion therapy in an operating theater's latex free kit) and drug: infusion computability (where e.g. GTN is adsorbed onto a standard PVC bag surface). It has been proposed therefore to describe the unit dose of all polyfusors (and any other such container) as a POLYETHYLENE BOTTLE in the VMP.

6.2.4.12.18.13 Style guide rules for Radiopharmaceuticals:

6.2.4.12.18.13.1 Radionuclide name

The complete *salt name* (eg calcium chloride) will appear **before** the *isotope* (45Ca), which will be in the format *atomic number* (45 in this case) followed by the *chemical symbol* (Ca)

e.g **Salt – (number – symbol)**

Examples:

Calcium Chloride (45Ca) is correct

Calcium (45Ca) Chloride is incorrect

Calcium Chloride (Ca45) is incorrect

6.2.4.12.18.13.2 Strength

Strength representation **does not occur in any** radiopharmaceutical product term

6.2.4.12.18.13.3 Unit dose

Unit dose representation is only applicable to those radiopharmaceutical product terms that are **NOT**reconstituted radiolabelling kits i.e. only therapeutic and diagnostic radiopharmaceutical products supplied as ready to use products should have a unit dose in the term.

In addition, unreconstituted labeling kits should have a unit dose specified.

6.2.4.12.18.13.4 Examples of radiopharmaceutical terms:

Sodium phosphate [32P] injection solution 5mL vial

Strontium chloride [89Sr] injection solution 1mL vial

MIBG [123I] injection solution 1mL vial

These terms have unit dose and volume since they are supplied as ready to use radiopharmaceuticals. They do not have any strength representation.

Technetium [99Tc] albumin microspheres injection solution

Technetium [99Tc] gluconate injection solution

These terms have neither strength nor unit dose representation since they are reconstituted radiolabelled products

Tin colloid powder for injection solution vial

Tin pyrophosphate BP powder for injection solution vial

Note:

The above two products have no radioactivity. They are simply carrier substances awaiting reconstitution with a radioactive label. These terms would be used for ordering/stock control purposes within the pharmacy and not for entry into a patients clinical record. They have no strength but a unit dose is specified.

6.2.4.12.19 Abbreviations

Currently *SNOMED concepts* are described by a single unabbreviated term. There are discussions due to take place regarding the requirement for a standardized abbreviated term. The list below identifies acceptable abbreviations that should be used during any authoring processes.

LONGEST TERM	Abbreviation 1	Abbreviation 2	Shortest abbreviation allowed
Cartridge			cart
I-V infusion	i-v inf	iv inf	inf
Prefilled syringe	p/f syringe	p/f syrg	pf syrg
Syringe	syrng		syrg
Units			u [24]
Gender			
Female	fem		f
Male			m
Pediatric	ped		p
Size			
Small	sml		S
Medium	med		M
Large	lge		L
Extra large	x-lge		XL

6.2.4.12.20 Type of drug preparation

This section of the hierarchy lists the various pharmaceutical/biologic forms that a preparation can take. For example, tablet, oral liquid, injection etc. In the table below, these types have been mapped to the VTM sub-categories.

24 'Unit' is not to be used.

Table 188: Summary table

Type	VTM sub-Type	Comments
Implant (product)	Implantable	
Implantation tablet (product)	Implantable	
Implantation chain (product)	Implantable	
Implant dosage form (product)	Implantable	
Parenteral drug preparation (product)	Injectable	
Injection (product)	Injectable	
Injection solution (product)	Injectable	
Injection suspension (product)	Injectable	
Injection emulsion (product)	Injectable	
Injection powder (product)	Injectable	
Powder for injection solution (product)	Injectable	
Powder for injection suspension (product)	Injectable	
Powder and solvent for injection solution (product)	Injectable	
Powder and solvent for injection suspension (product)	Injectable	
Injection concentrate (product)	Injectable	
Intravenous infusion (product)	Injectable	
Infusion solution (product)	Injectable	
Infusion powder (product)	Injectable	
Powder for infusion solution (product)	Injectable	
Powder and solvent for infusion (product)	Injectable	

Type	VTM sub-Type	Comments
Infusion concentrate (product)	Injectable	
Solvent for parenteral use (product)	Injectable	
Radiopharmaceutical preparation kit (product)	Injectable	
Radiopharmaceutical dosage form (product)	Injectable	Could also be oral use
Parenteral dosage form (product)	Injectable	
Oral drops (product)	Oral	
Oral drops solution (product)	Oral	
Oral drops suspension (product)	Oral	
Oral drops emulsion (product)	Oral	
Oral liquid (product)	Oral	
Oral solution (product)	Oral	
Oral suspension (product)	Oral	
Oral emulsion (product)	Oral	
Powder for oral solution (product)	Oral	
Powder for oral suspension (product)	Oral	
Granules for oral solution (product)	Oral	
Granules for oral suspension (product)	Oral	
Powder and solvent for oral solution (product)	Oral	
Powder and solvent for oral suspension (product)	Oral	
Syrup (product)	Oral	
Powder for syrup (product)	Oral	

Type	VTM sub-Type	Comments
Granules for syrup (product)	Oral	
Soluble tablet (product)	Oral	Could also be cutaneous use eg pot permang tabs
Dispersible table (product)	Oral	
Herbal tea (product)	Oral	
Instant herbal tea (product)	Oral	
Oral powder (product)	Oral	
Effervescent powder (product)	Oral	
Granules (product)	Oral	Could also be cutaneous use
Effervescent granules (product)	Oral	Could also be cutaneous use
Gastro-resistant granules (product)	Oral	
Prolonged-release granules (product)	Oral	
Modified-release granules (product)	Oral	
Cachet (product)	Oral	
Capsule (product)	Oral	
Hard capsule (product)	Oral	
Soft capsule (product)	Oral	
Gastro-resistant capsule (product)	Oral	
Prolonged-release capsule (product)	Oral	
Modified-release capsule (product)	Oral	
Tablet (product)	Oral	
Coated tablet (product)	Oral	
Film-coated tablet (product)	Oral	

Type	VTM sub-Type	Comments
Effervescent tablet (product)	Oral	
Gastro-resistant tablet (product)	Oral	
Prolonged-release tablet (product)	Oral	
Modified-release tablet (product)	Oral	
Oral lyophilisate (product)	Oral	
Pillule (product)	Oral	
Oral dosage form (product)	Oral	
Liquid and semi-solid oral dosage form (product)	Oral	
Solid oral dosage form (product)	Oral	
Oral drug preparation (product)	Oral	
Drug capsule (product)	Oral	
Pill (product)	Oral	
Time-release capsule (product)	Oral	
Caplet (product)	Oral	
Chewable tablet (product)	Oral	
Drug suspension (product)	Oral	
Drug granules (product)	Oral	Could also be cutaneous use
Oral gel (product)	Oropharyngeal	Could also be oral use
Oral paste (product)	Oropharyngeal	Could also be oral use
Oral gum (product)	Oropharyngeal	
Gargle (product)	Oropharyngeal	
Concentrate for gargle (product)	Oropharyngeal	
Powder for solution for gargle (product)	Oropharyngeal	

Type	VTM sub-Type	Comments
Tablet for solution for gargle (product)	Oropharyngeal	
Oromucosal liquid (product)	Oropharyngeal	
Oromucosal solution (product)	Oropharyngeal	
Oromucosal suspension (product)	Oropharyngeal	
Oromucosal drops (product)	Oropharyngeal	
Oromucosal spray (product)	Oropharyngeal	
Sublingual spray (product)	Oropharyngeal	
Mouth wash (product)	Oropharyngeal	
Tablet for solution for mouthwash (product)	Oropharyngeal	
Ginigival solution (product)	Oropharyngeal	
Oromucosal gel (product)	Oropharyngeal	
Oromucosal paste (product)	Oropharyngeal	
Medicated chewing-gum (product)	Oropharyngeal	
Ginigival gel (product)	Oropharyngeal	
Gingival paste (product)	Oropharyngeal	
Oromucosal capsule (product)	Oropharyngeal	
Sublingual tablet (product)	Oropharyngeal	
Buccal tablet (product)	Oropharyngeal	
Muco-adhesive buccal tablet (product)	Oropharyngeal	
Lozenge (product)	Oropharyngeal	
Dental gel (product)	Oropharyngeal	
Dental stick (product)	Oropharyngeal	

Type	VTM sub-Type	Comments
Dental insert (product)	Oropharyngeal	
Dental powder (product)	Oropharyngeal	
Dental liquid (product)	Oropharyngeal	
Dental solution (product)	Oropharyngeal	
Dental suspension (product)	Oropharyngeal	
Dental emulsion (product)	Oropharyngeal	
Toothpaste (product)	Oropharyngeal	
Oromucosal AND/OR gingival dosage form (product)	Oropharyngeal	
Dental dosage form (product)	Oropharyngeal	
Drug ointment (product)	Cutaneous	
Drug stick (product)	Cutaneous	
Drug cleansing solution (product)	Cutaneous	
Drug cream, 0,5% (product)	Cutaneous	
Drug ointment, 0,2% (product)	Cutaneous	
Drug aerosol foam, 10% (product)	Cutaneous	
Drug gel, 0,05% (product)	Cutaneous	
Drug aerosol, 0,5% (product)	Cutaneous	
Drug lotion (product)	Cutaneous	
Drug lotion, 0,025% (product)	Cutaneous	
Drug pledget, 0,5% (product)	Cutaneous	
Drug solution, 0,5% (product)	Cutaneous	
Drug ointment, 2,5% (product)	Cutaneous	
Drug cream (product)	Cutaneous	

Type	VTM sub-Type	Comments
Drug aerosol foam (product)	Cutaneous	
Drug cream, 0,01% (product)	Cutaneous	
Drug cream, 2,5% (product)	Cutaneous	
Drug aerosol, 0,04% (product)	Cutaneous	
Drug patch (product)	Cutaneous	
Drug shampoo (product)	Cutaneous	
Drug paste (product)	Cutaneous	
Bath additive (product)	Cutaneous	
Cream (product)	Cutaneous	
Gel (product)	Cutaneous	
Ointment (product)	Cutaneous	
Cutaneous paste (product)	Cutaneous	
Cutaneous foam (product)	Cutaneous	
Shampoo (product)	Cutaneous	
Cutaneous spray (product)	Cutaneous	
Cutaneous solution spray (product)	Cutaneous	
Cutaneous liquid (product)	Cutaneous	
Cutaneous solution (product)	Cutaneous	
Concentrate for cutaneous solution (product)	Cutaneous	
Cutaneous suspension (product)	Cutaneous	
Cutaneous emulsion (product)	Cutaneous	
Cutaneous powder (product)	Cutaneous	
Solution for iontophoresis (product)	Cutaneous	

Type	**VTM sub-Type**	**Comments**
Transdermal patch (product)	Cutaneous	
Collodion (product)	Cutaneous	
Medicated nail laquer (product)	Cutaneous	
Poultice (product)	Cutaneous	
Cutaneous stick (product)	Cutaneous	
Cutaneous sponge (product)	Cutaneous	
Impregnated dressing (product)	Cutaneous	
Wound stick (product)	Cutaneous	
Sealant (product)	Cutaneous	
Powder and solvent for sealant (product)	Cutaneous	
Impregnated pad (product)	Cutaneous	
Cutaneous AND/OR transdermal dosage form (product)	Cutaneous	
Cutaneous powder spray (product)	Cutaneous	
Cutaneous suspension spray (product)	Cutaneous	
Drug ointment, 0,25% (product)	Cutaneous	
Drug ointment, 1% (product)	Cutaneous	
Drug solution, 0,05% (product)	Cutaneous	
Drug gel, 0,25% (product)	Cutaneous	
Drug cream, 0,025% (product)	Cutaneous	
Drug lotion, 1% (product)	Cutaneous	
Drug lotion, 0,05% (product)	Cutaneous	
Drug lotion, 0,5% (product)	Cutaneous	

Type	VTM sub-Type	Comments
Drug ointment, 0,05% (product)	Cutaneous	
Drug lotion, 0,1% (product)	Cutaneous	
Drug aerosol (product)	Cutaneous	
Drug dressing, 4mg/cm^2 (product)	Cutaneous	
Drug dressing (product)	Cutaneous	
Drug paste, 0,5% (product)	Cutaneous	
Drug powder, 0,1% (product)	Cutaneous	
Drug cream, 0,1% (product)	Cutaneous	
Drug cleansing suspension (product)	Cutaneous	
Drug solution, 0,01% (product)	Cutaneous	
Liniment (product)	Cutaneous	
Drug pledget (product)	Cutaneous	
Drug gel, 0,1% (product)	Cutaneous	
Drug cream, 0,25% (product)	Cutaneous	
Drug aerosol, 0,01% (product)	Cutaneous	
Drug ointment, 0,1% (product)	Cutaneous	
Drug solution, 1,0% (product)	Cutaneous	
Drug gel (product)	Cutaneous	
Drug cream, 1% (product)	Cutaneous	
Drug solution (product)	Cutaneous	
Drug cream, 0,02% (product)	Cutaneous	
Drug lotion, 0,25% (product)	Cutaneous	
Drug cream, 0,05% (product)	Cutaneous	

Type	VTM sub-Type	Comments
Drug powder (product)	Cutaneous	could also be oral
Drug solution, 0,1% (product)	Cutaneous	could also be oral
Drug lotion, 2,5% (product)	Cutaneous	
Drug ointment, 0,025% (product)	Cutaneous	
Eye cream (product)	Occular	
Eye gel (product)	Occular	
Eye ointment (product)	Occular	
Eye drops (product)	Occular	
Eye drops solution (product)	Occular	
Powder and solvent for eye drops solution (product)	Occular	
Powder and solvent for eye drops suspension (product)	Occular	
Prolonged-release eye drops (product)	Occular	
Solvent for eye drops reconstitution (product)	Occular	
Eye lotion (product)	Occular	
Solvent for eye lotion reconstitution (product)	Occular	
Ophthalmic insert (product)	Occular	
Eye/ear/nose drops (product)	Occular	More than one sub-type
Eye/ear/nose ointment (product)	Occular	More than one sub-type
Ocular dosage form (product)	Occular	
Ocular AND otic AND nasal dosage form (product)	Occular	More than one sub-type
Nasal cream (product)	Nasal	

Type	VTM sub-Type	Comments
Nasal gel (product)	Nasal	
Nasal ointment (product)	Nasal	
Nasal drops (product)	Nasal	
Nasal drops solution (product)	Nasal	
Nasal drops suspension (product)	Nasal	
Nasal drops emulsion (product)	Nasal	
Nasal powder (product)	Nasal	
Nasal spray (product)	Nasal	
Nasal spray solution (product)	Nasal	
Nasal spray suspension (product)	Nasal	
Nasal spray emulsion (product)	Nasal	
Nasal wash (product)	Nasal	
Nasal stick (product)	Nasal	
Nasal dosage form (product)	Nasal	
Ear cream (product)	Aural	
Ear gel (product)	Aural	
Ear ointment (product)	Aural	
Ear drops (product)	Aural	
Ear drops solution (product)	Aural	
Ear drops emulsion (product)	Aural	
Ear powder (product)	Aural	
Ear spray (product)	Aural	
Ear spray solution (product)	Aural	
Ear spray suspension (product)	Aural	

Type	VTM sub-Type	Comments
Ear spray emulsion (product)	Aural	
Ear wash (product)	Aural	
Ear wash solution (product)	Aural	
Ear wash emulsion (product)	Aural	
Ear tampon (product)	Aural	
Ear stick (product)	Aural	
Otic dosage form (product)	Aural	
Nebuliser liquid (product)	Respiratory	
Nebuliser solution (product)	Respiratory	
Nebuliser suspension (product)	Respiratory	
Powder for nebuliser suspension (product)	Respiratory	
Powder for nebuliser solution (product)	Respiratory	
Nebuliser emulsion (product)	Respiratory	
Pressurised inhalation (product)	Respiratory	
Pressurised inhalation solution (product)	Respiratory	
Pressurised inhalation suspension (product)	Respiratory	
Pressurised inhalation emulsion (product)	Respiratory	
Inhalation powder (product)	Respiratory	
Hard capsule inhalation powder (product)	Respiratory	
Pre-dispensed inhalation powder (product)	Respiratory	

Type	VTM sub-Type	Comments
Inhalation vapor (product) Inhalation vapor (product)	Respiratory	
Inhalation vapor powder (product) Inhalation vapor powder (product)	Respiratory	
Inhalation vapor capsule (product) Inhalation vapor capsule (product)	Respiratory	
Inhalation vapor solution (product) Inhalation vapor solution (product)	Respiratory	
Inhalation vapor tablet (product) Inhalation vapor tablet (product)	Respiratory	
Inhalation vapor ointment (product) Inhalation vapor ointment (product)	Respiratory	
Inhalation vapor liquid (product) Inhalation vapor liquid (product)	Respiratory	
Inhalation gas (product)	Respiratory	
Endotracheopulmonary instilliation solution (product)	Respiratory	
Powder for endotracheopulmonary instillation solution (product)	Respiratory	
Powder and solvent for endotracheopulmonary instillation solution (product)	Respiratory	
Endotracheopulmonary intillation suspension (product)	Respiratory	
Tracheopulmonary dosage form (product)	Respiratory	
Inhalation dosage form (product)	Respiratory	
Rectal cream (product)	Rectal	
Rectal gel (product)	Rectal	

Type	VTM sub-Type	Comments
Rectal ointment (product)	Rectal	
Rectal foam (product)	Rectal	
Enema (product)	Rectal	
Rectal solution (product)	Rectal	
Rectal suspension (product)	Rectal	
Rectal emulsion (product)	Rectal	
Concentrate for rectal solution (product)	Rectal	
Powder for rectal suspension (product)	Rectal	
Tablet for rectal solution (product)	Rectal	
Tablet for rectal suspension (product)	Rectal	
Suppository (product)	Rectal	
Rectal capsule (product)	Rectal	
Rectal tampon (product)	Rectal	
Rectal dosage form (product)	Rectal	
Drug suppository (product)	Rectal	
Vaginal cream (product)	Vaginal	
Vaginal gel (product)	Vaginal	
Vaginal ointment (product)	Vaginal	
Vaginal foam (product)	Vaginal	
Vaginal liquid (product)	Vaginal	
Vaginal solution (product)	Vaginal	
Vaginal suspension (product)	Vaginal	

Type	VTM sub-Type	Comments
Vaginal emulsion (product)	Vaginal	
Tablet for vaginal solution (product)	Vaginal	
Pessary (product)	Vaginal	
Vaginal capsule (product)	Vaginal	
Hard vaginal capsule (product)	Vaginal	
Soft vaginal capsule (product)	Vaginal	
Vaginal tablet (product)	Vaginal	
Effervescent vaginal tablet (product)	Vaginal	
Medicated vaginal tampon (product)	Vaginal	
Vaginal device (product)	Vaginal	
Vaginal dosage form (product)	Vaginal	
Intravesical solution (product)	Genitourinary	
Bladder irrigation (product)	Genitourinary	
Powder for bladder irrigation (product)	Genitourinary	
Urethral gel (product)	Genitourinary	
Urethral stick (product)	Genitourinary	
Intravesical AND/OR urethral dosage form (product)	Genitourinary	
Type of drug preparation (product)		Not true 'types'
Fish insulin (product)		Not true 'types'
Porcine insulin (product)		Not true 'types'
Water soluble contrast medium (product)		Not true 'types'

Type	VTM sub-Type	Comments
Lipiodol (product)		Not true 'types'
Radiographic iodinated contrast medium (product)	Not true 'types'	
Types of insulin (product)		Not true 'types'
Types of contrast medium (product)		Not true 'types'
Drug preparation (product)		Not true 'types'
Peritoneal dialysis solution (dose form) (product)	? Need an extra sub-type class	
Haemofiltration solution (product) Hemofiltration solution (product)		? Need an extra sub-type class
Haemodialysis solution (product) Hemodialysis solution (product)		? Need an extra sub-type class
Haemodialysis solution concentrate (product) Hemodialysis solution concentrate (product)	? Need an extra sub-type class	
Endocervical gel (product)		? Need an extra sub-type class
Powder and solvent for endocervical gel (product)	? Need an extra sub-type class	
Intrauterine device (product)		? Need an extra sub-type class
Radiopharmaceutical precursor (product)		? Need an extra sub-type class
Radionuclide generator (product)		? Need an extra sub-type class
Gastroenteral liquid (product)		? Need an extra sub-type class
Gastroenteral solution (product)		? Need an extra sub-type class
Gastroenteral suspension (product)		? Need an extra sub-type class
Gastroenteral emulsion (product)		? Need an extra sub-type class
Organ preservation solution (product)		? Need an extra sub-type class

Type	VTM sub-Type	Comments
Irrigation solution (product)		? Need an extra sub-type class
Stomach irrigation (product)		? Need an extra sub-type class
Gastroenteral dosage form (product)		? Need an extra sub-type class
Endocervical dosage form (product)		? Need an extra sub-type class
Dialysis dosage form (product)		? Need an extra sub-type class

6.2.5 Miscellaneous Topics

6.2.5.1 References for Editorial Rules and Known Problems

The approved set of detailed editorial rules and guidelines are documented in the *SNOMED CT* Editorial Guide. Parts of the Editorial Guide are reproduced verbatim in the User Guide.

Known problems and issues are not documented here but instead are tracked on the *SNOMED CT Collaborative Space* at csfe.aceworkspace.net, under project "*IHTSDO*". It is possible to review a brief summary of each project without a login, but if you would like access to the *Collaborative Space*, please contact collabnet(at)ihtsdo.org with your contact details and a list of the project(s) to which you would like access. Known problems and issues are found in the content projects tracker under project "*IHTSDO*".

6.2.5.2 Terms Prefaced with Symbols

There are some *terms* in *SNOMED CT* that are prefaced with a symbol in square brackets. These *concept* codes were inherited from *CTV3* and were used to facilitate mapping to *ICD-10*. They have all been *retired* by moving them to the *UK NHS extension*, and are not recommended for use in clinical records.

Explanations of these *term* prefixes are as follows:

Table 189: Term Preface Symbols

[X]	*Terms* starting with [X] were initially used in the *Read Codes* in the 1995 release, in order to identify *ICD-10 terms* that were not present in ICD-9.
[D]	*Terms* starting with [D] are also from *CTV3*, and identify *terms* contained in ICD-9 Chapter XVI 'Symptoms signs and ill-defined conditions' and *ICD-10* Chapter XVIII 'Symptoms signs and abnormal clinical and laboratory findings, not elsewhere classified'. The [D] meant that in CTV3the code was intended for use in a diagnosis field in the record, even though the *term* meaning is not a kind of disease.
[V]	A *term* starting with [V] identifies *concept* codes derived from ICD-9 'Supplementary classification of factors influencing health *status* and contact with health services (V codes)', and *ICD-10* Chapter XXI 'Factors influencing health *status* and contact with health services (Z codes)'.
[M]	A *term* starting with [M] identifies Morphology of Neoplasm *terms* present in *ICD-9* and ICD 10.

[SO] A *term* starting with [SO] signifies that the *term* was contained in OPCS-4 (Office of Population, Censuses and Surveys - Classification of Surgical Operations and Procedures - 4th Revision) Chapter Z subsidiary classification of sites of operation in *CTV3*.

[Q] A *term* starting with [Q] identifies temporary qualifying *terms* inherited from *CTV3*.

6.2.5.3 Negation

The meaning of some *concept* codes in *SNOMED CT* depends conceptually on negation (e.g. absence of *X*, lack of *X*, unable to do X etc).

6.2.5.3.1 Negation and Context

The | Situation with explicit context | *hierarchy* is intended to manage this kind of semantic situation. The *concept model* allows a *concept* code in the | Situation with explicit context | *hierarchy* to be related to the | Clinical finding | about which context is asserted. For example, | Absence of nausea and vomiting (situation) |is modeled as a | Situation with explicit context | in which the finding of | Nausea and vomiting (disorder) | is absent.

The inclusion of negated meanings introduces complications into *query* formulation, machine classification, and reasoning tasks. The inclusion of a NOT logical operator into the *SNOMED CT* compositional model could simplify modeling of negated meanings. The current release of *SNOMED CT* does not directly support classification using this operator, but some modeling formalisms in current use today (including database formalisms, *Description Logic* formalisms) include a NOT operator as a fundamental modeling *primitive*.

6.3 Machine Readable Concept Model

The *Concept Model* is the set of rules that govern the ways in which *SNOMED CT concepts* are permitted to be modeled using *Relationships* to other *Concepts*. The *Machine Readable Concept Model* (*MRCM*) represents these rules in a form that can be read by a computer and applied to test that *concept* definitions comply with the rules.

The primary requirements addressed by the current *MRCM* relate to supporting content authoring and validation prior to distribution. However, the *MRCM* also has a potential value for implementers as a source of the rules that determine whether particular *postcoordinated expression* refinements are permitted.

The *MRCM* is based on a logical model that specifies:

- CM-Domains: Sets of *SNOMED CT concepts* to which a common set of *Concept Model* Constraints apply.
- CM-Attributes: Sets of *SNOMED CT concepts* that can be used as *relationship types*.
- CM-Ranges: Sets of *SNOMED CT concepts* that can be used as values for a particular defining *Relationship*.
- CM-Constraints: Rules that determine which combinations of CM-Attributes and CM-Ranges may be applied to *concepts* in a CM-Domain.

The current prototype version of the *MRCM* is represented as a relational database schema with an XML schema to support export and import of data.

Subsequent activities within *IHTSDO* Working Group and related work by *IHTSDO Members* has identified requirements for additional representations that are more readily refinable to support implementation use-cases.

Note: An updated representation of the *MRCM* is in preparation are part of work on a consistent family of processable languages for querying, constraining and binding SNOMED CT. Based on current expectations a *Technology Preview* should be available in the first quarter of 2015.

Chapter 7

7 Terminology Services Guide

7.1 Representing SNOMED CT resources

7.1.1 Choosing a terminology server view

SNOMED CT Release Format 2 is designed to enable the distribution and use of a full historical view of *SNOMED CT* from its first release in 2002 up to its most recent release. This allows *terminology servers* to provide a range of different views of *SNOMED CT*. However, it does not require that all *terminology servers* support the full range of views.

Table 190 identifies three options for the views that a *SNOMED CT terminology server* may support. The simplest of these is the single *snapshot view* which provides access to a single *release version*. This closely matches the view provided by the original *SNOMED CT release format* (*RF1*). The most powerful *full view* which allows the server to provide access to any selected version of *SNOMED CT* from a single representation of the *SNOMED CT* resource. This makes full use of the version features in *RF2*. Alternatively a server may provide a selected set of snapshots representing versions of known interest to its users.

People designing a *terminology server* need to decide whether their server will only provide access to a single current view of the *SNOMED CT* resource or will also support retrospective views of earlier versions of the terminology. The single *snapshot view* is simplest to implement and matches the service most vendors offered with original *SNOMED CT release format* (*RF1*). A more complete view is now possible using *Release format 2* and this offers several significant advantages. It supports incremental updates allowing smoother transition as new versions become available. It also allows changes between versions to be detected more easily and can be used to evaluate queries against an earlier version for comparative purposes.

People choosing a *terminology server* need to consider whether a server that only supports a single *snapshot view* of the current version meets their requirements. If they require access to previous versions a server that supports the *full view* is likely to be the best long term solution. A server that allows access to multiple discrete snapshots may provide a reasonable interim solution but may be less flexible and less easy to maintain.

Table 190: SNOMED CT views that may be supported by terminology servers

View	*Description*
Snapshot view	A *snapshot view terminology service* provides access to the content of the current state of all the *components* of the *International Release* and any chosen *Extension Releases*.

View	***Description***
Multi-*snapshot view*	A "multi-*snapshot view*" *terminology service* provides access to: • the content of the current state and content of all *components* of the *International release* and any chosen *Extension Releases*; • the content of one or more additional *snapshot views*, each of which represents the state of all *components* at a different fixed point in time. A "multi-*snapshot view*" *terminology server* may provide access to delta views that report the differences between two *snapshot views*. This is limited to comparisons of specific points represented by the available *snapshot views*.
Full view	A *Full view terminology service* provides access to: • the complete content of the full *International release* and any chosen *Extension Releases*; • the state and content of all *components* as they were at any specified point in time. A *full view terminology server* should also provide access to views that show the changes to *components* between any two specified points in time.

The *full view* is required to support some *SNOMED CT* use cases but many requirements can be adequately met by providing access to a current *Snapshot view*. The multi-*snapshot view* is an approach that may meet some requirements that are not met by a single snapshot without requiring support for the *Full view*.

Note: *terminology servers* that do not support the *Full view* still need to be able to import from a *Full release* as *Extension* providers are not required to provide the snapshot or *delta releases* (*Importing release types*)

7.1.2 Choosing a technical approach

People designing a *terminology server* need to decide how they will store and access the *SNOMED CT* resources. This decision depends on a variety of factors including: types of *Terminology services* required, the technical environment in which development is undertaken and the experience of the developers.

People choosing a *terminology server* need to know whether the server will meet their requirements and whether it works effectively in their preferred technical environment. They will also wish to be sure it delivers the required functionality and performance. While they may not be directly interested in technical approach to representation of *SNOMED CT* resources, these design decisions are likely to affect the ability of a server to meet their requirements.

The following sub-sections briefly outline some of the technical options.

7.1.2.1 Direct use of release files in a relational database

The distributed *release files* can be imported directly into a database schema that matches the distribution file specification. This data then provides the core resource at the heart of a *terminology server*.

This direct use of distributed files in a relational database has the advantage of allowing simple installation. However, it may not be the most efficient approach in terms of performance or file size. Some *terminology services* require relatively complex queries with multiple joins, and need to be completed in fractions of a second to provide an acceptable *user interface*.

Example: To display the set of *subtype children* of a *concept* with their *preferred terms* in a specified language or *dialect* requires joins several joins between *concepts*, *Relationships*, *Descriptions* and a language *refset*.

To search for a term matching a supplied pattern in a *concept* that represents a type of procedure also requires multiple joins to link the *Descriptions* with matching terms to the relevant *concept* and test whether it is a *subtype* of the 71388002 | Procedure (procedure) | *concept*.

The performance criteria of searches and joins in very large relational databases vary significantly. Therefore, different optimizations may need to be used to achieve acceptable response times according to the nature of the relational database system.

An additional consideration for *RF2* implementations is the way in which alternative views are supported since, without optimization these may have a significant impact on performance.

7.1.2.2 Alternative relational structures

There is no requirement to use the data structure as distributed. Other structures can be used provided that they are able to deliver the range of *Terminology services* required. Options include:

- Partially denormalized representations that omit direct representation of some components.

  Example: Frequently used information distributed as part of a *Refset* could be represented by direct inclusion of the added information as additional columns in the table representing the referenced *component*.

- Omission of some of the tables where a particular function is not required.

  Example: The *Refset* tables representing *maps* could be omitted if the intended uses of the *terminology server* explicitly exclude *mapping*.

- Replacement of some of the supporting tables with proprietary alternatives that deliver equivalent or enhanced functionality.

  Example: The word search support tables could be replaced by other tables or indices generated by the *terminology server* when loading the distribution files.

7.1.2.3 Non-relational structures

Although the primary distribution format is relational, this does not require *terminology servers* to utilize a relational database as the primary or only storage format. The requirements for *terminology services* may also be met by representing some or all of the distributed data in other forms including object-oriented databases, Extensible Mark-up Language (XML) and/or proprietary data structures. These structures may be used separately or, in some cases, in combination with a relational database.

7.1.3 Example of a Full View Relational Representation

This section outlines an example of a relational approach to representation of a *full view* of the *SNOMED CT* Resources. The example has been developed and tested using the Open Source database MySql Community Edition.

The example schema is based closely on the *RF2* structure and is used in subsequent discussions of implementation issue and options for addressing those issues.

Note: The approach described here is only an illustrative example. It shows one way to represent the data but should not to be interpreted as a recommended or standard approach.

The general approach is as follows:

- Each datatype in the *RF2* specification is expressed with a common mapping to a database datatype:
  - Alternative implementations following the same general pattern could use a different datatype map but the mapping should be consistent within an implementation. Reasons for different datatype maps include implementer preferences and the capabilities of the database.
- Each of the main file types specified in *RF2* is instantiated as a database table:
  - Each table is named for the *component* type (e.g. sct2_Concept, sct2_Description, sct2_Relationship, sct2_Identifier).
  - Each field in these tables has column name from the *release file*
  - Each field is assigned the appropriate datatype (and where appropriate size).
- *Refsets* are represented slightly differently from the other files:
  - One table structure for each distinct structure present in the release data:
    - der2_Refset.
    - der2_Reset_c.
    - der2_Refset_cc.
    - der2_Refset_ci.
    - der2_Refset_i.
    - der2_Refset_s.
    - der2_Refset_ss.
    - ... etc as new structures are added.
  - The first six fields in these tables have the common column names from the *release file*
  - The subsequent fields are named by type and position:
    - sctid1.
    - string1.
    - integer1.
    - ... etc.
  - This polymorphic field approach to column naming is used because column names may varies between *release files* for different *Reference Set* patterns, even when column data types are the same.

    Note: Two other approaches could be used here.

    1. A separate table for each type of *Refset* based on column names rather than on structure. This would require a several tables with similar types of *Relationships* to other *components*.
    2. A single general purpose *Refset* table with multiple polymorphic fields. For example, *strings* that could be used to represent the other data types. This could cause inefficiencies for sctid type fields as the joins between these and target *components* would be heterogeneous.

7.1.3.1 Example Datatype Mapping for Relational View

The following table provides example mapping from the *SNOMED CT RF2* datatypes to appropriate datatypes supported by MySql.

Table 191: Example Datatype Mappings

RF2 **Datatype**	**MySql Datatype**	**Comment on Mapping**
SCTID	BIGINT	Both these datatypes represent 64-bit integers.
UUID	BINARY(16)	MySql does not have a native datatype for *UUID*. The BINARY(16) representation is most economical for storage and most efficient for indexing. This requires a transformation on storage or review. The example queries in this guide use the simple transformations functions shown in *Table 192*. An alternative is to use CHAR or VARCHAR representations. This does not require the transformations noted above. However, use of VARCHAR (36) costs 38 bytes rather than 16 bytes per *UUID* and due to use of UTF8 using CHAR (36) consumes a fixed 108 bytes per UID in a MySql table. More importantly the index performance is poorer for these *string* representations.
Integer	INT	Both these datatypes represent 32-bit integers.
String	VARCHAR (Len)	VARCHAR is used in preference to CHAR as it provides more space efficient storage. Note that in the UTF8 encoded tables required for the MyISAM database reserves three bytes per character for fixed length *strings*. In contrast VARCHAR uses the number of bytes actually plus one or two bytes to specify length. Use of VARCHAR does result in some loss of performance but *strings* are only used in *Descriptions*, *string refsets* and *Identifier* tables. In all these cases *strings* with a significant range of lengths are used and the space penalty for using CHAR datatypes would be high.
Boolean	TINYINT	MySql treats the datatype name *boolean* as an alias for TINYINT. In the examples this mapping is made explicit.
Time	DATETIME	This is the full representation of date and time and is used to ensure compatibility with existing data and potential accommodation of time stamped data. The more compact DATE type could be used with current data as the *effectiveTime* is currently a date only representation. However, the more flexible DATETIME has been preferred in the examples because this emphasizes the fact that in an International environment the *effectiveTime* implies the UTC time and thus the date alone is not a precise representation.

Table 192: Example UUID transformation

Action	
Load or insert to storage	SET [column-name] = UNHEX(REPLACE(@uid,'-',''))
Select from storage	RenderUid([column-name])

Action	
UNHEX	A built in MySql function that converts a hexadecimal *string* to binary.

RenderUid	`FUNCTION `RenderUid`(Uid blob) RETURNS varchar(36) CHARSET utf8`
	`BEGIN`
	`Set @Tmp = Hex(uid);`
	`RETURN CONCAT(SUBSTRING(@Tmp,1,8),'-',SUBSTRING(@Tmp,9,4),'-',`
	
	`SUBSTRING(@Tmp,13,4),'-',SUBSTRING(@Tmp,17,4),'-',SUBSTRING(@Tmp,21));`
	`END`

7.1.3.2 Example Full View Concept Table

```
CREATE TABLE `sct2_concept` (
  `id` BIGINT NOT NULL DEFAULT 0,
  `effectiveTime` DATETIME NOT NULL DEFAULT '0000-00-00 00:00:00',
  `active` TINYINT NOT NULL DEFAULT 0,
  `moduleId` BIGINT NOT NULL DEFAULT 0,
  `definitionStatusId` BIGINT NOT NULL DEFAULT 0,
  PRIMARY KEY (`id`,`effectiveTime`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

Figure 71: Create Concept Table

Tip: Some of the approaches to optimization suggested elsewhere in the guide result in changes to this example schema. You may wish to consider these before implementing this schema.

```
LOAD DATA LOCAL INFILE '[path]sct2_concept_[AdditionalInfo].txt'
INTO TABLE `sct2_concept`
LINES TERMINATED BY '\r\n' IGNORE 1 LINES;
```

Figure 72: Import Concept file

7.1.3.3 Example Full View Description Table

```
CREATE TABLE `sct2_description` (
  `id` BIGINT NOT NULL DEFAULT 0,
  `effectiveTime` DATETIME NOT NULL DEFAULT '0000-00-00 00:00:00',
  `active` TINYINT NOT NULL DEFAULT 0,
  `moduleId` BIGINT NOT NULL DEFAULT 0,
  `conceptId` BIGINT NOT NULL DEFAULT 0,
  `languageCode` VARCHAR(3) NOT NULL DEFAULT '',
  `typeId` BIGINT NOT NULL DEFAULT 0,
  `Term` VARCHAR(255) NOT NULL DEFAULT '',
  `caseSignificanceId` BIGINT NOT NULL DEFAULT 0,
  PRIMARY KEY (`id`,`effectiveTime`),
  KEY `sct2_description_concept` (`conceptId`)
) ENGINE=MyISAM DEFAULT CHARSET=utf
```

Figure 73: Create Description Table

Tip: Some of the approaches to optimization suggested elsewhere in the guide result in changes to this example schema. You may wish to consider these before implementing this schema.

```
LOAD DATA LOCAL INFILE '[path]sct2_description_[AdditionalInfo].txt'
INTO TABLE sct2_description
LINES TERMINATED BY '\r\n' IGNORE 1 LINES;
```

Figure 74: Import Description file

```
ADD INDEX ix_sct2_description_3 ON sct2_description(`conceptId`,`typeId`,`languageCode`)
```

Figure 75: Index Description Table - Concept

7.1.3.4 Example Full View relationships table

```
CREATE TABLE `sct2_relationship` (
  `id` BIGINT NOT NULL DEFAULT 0,
  `effectiveTime` DATETIME NOT NULL DEFAULT '0000-00-00 00:00:00',
  `active` TINYINT NOT NULL DEFAULT 0,
  `moduleId` BIGINT NOT NULL DEFAULT 0,
  `sourceId` BIGINT NOT NULL DEFAULT 0,
  `destinationId` BIGINT NOT NULL DEFAULT 0,
  `relationshipGroup` INT NOT NULL DEFAULT 0,
  `typeId` BIGINT NOT NULL DEFAULT 0,
  `characteristicTypeId` BIGINT NOT NULL DEFAULT 0,
  `modifierId` BIGINT NOT NULL DEFAULT 0,
  PRIMARY KEY (`id`,`effectiveTime`),
  KEY `sct2_relationship_source` (`sourceId`,`characteristicTypeId`,`typeId`,`destinationId`),
  KEY `sct2_relationship_dest` (`destinationId`,`characteristicTypeId`,`typeId`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

Figure 76: Create relationships table

Tip: Some of the approaches to optimization suggested elsewhere in the guide result in changes to this example schema. You may wish to consider these before implementing this schema.

```
LOAD DATA LOCAL INFILE '[path]sct2_relationship_[AdditionalInfo].txt'
INTO TABLE sct2_relationship
LINES TERMINATED BY '\r\n' IGNORE 1 LINES;
```

Figure 77: Import Relationship file

7.1.3.5 Example Full View Identifier Table

```
CREATE TABLE `sct2_identifier` (
  `identifierSchemeId` BIGINT NOT NULL DEFAULT 0,
  `alternateIdentifier` VARCHAR(255) NOT NULL DEFAULT '',
  `effectiveTime` DATETIME NOT NULL DEFAULT '0000-00-00 00:00:00',
  `active` TINYINT NOT NULL DEFAULT 0,
  `moduleId` BIGINT NOT NULL DEFAULT 0,
  `referencedComponentId` BIGINT NOT NULL DEFAULT 0,
  PRIMARY KEY (`identifierSchemeId`,`alternateIdentifier`,`effectiveTime`),
  KEY `sct2_relationship_sctid` (`referencedComponentId`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

Figure 78: Create Identifier Table

Tip: Some of the approaches to optimization suggested elsewhere in the guide result in changes to this example schema. You may wish to consider these before implementing this schema.

```
LOAD DATA LOCAL INFILE '[path]sct2_identifier_[AdditionalInfo].txt'
INTO TABLE sct2_identifier
LINES TERMINATED BY '\r\n' IGNORE 1 LINES;
```

Figure 79: Index Identifier Table - Primary

7.1.3.6 Example Full View Refset Table

```
CREATE TABLE `sct2_refset_c` (
  `id` binary(16) NOT NULL DEFAULT '\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0',
  `effectiveTime` DATETIME NOT NULL DEFAULT '0000-00-00 00:00:00',
  `active` TINYINT NOT NULL DEFAULT 0,
  `moduleId` BIGINT NOT NULL DEFAULT 0,
  `refsetId` BIGINT NOT NULL DEFAULT 0,
  `referencedComponentId` BIGINT NOT NULL DEFAULT 0,
  `sctId1` BIGINT NOT NULL DEFAULT 0,
  PRIMARY KEY (`id`,`effectiveTime`),
  KEY `refset_c_id` (`refsetId`,`referencedComponentId`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

Figure 80: Create Component Refset Table

Tip: Some of the approaches to optimization suggested elsewhere in the guide result in changes to this example schema. You may wish to consider these before implementing this schema.

```
LOAD DATA LOCAL INFILE '[path]der2_cRefset_[AdditionalInfo].txt'
INTO TABLE `sct2_refset_c`
LINES TERMINATED BY '\r\n' IGNORE 1 LINES
(@uid, `effectiveTime`, `active`, `moduleId`, `refsetId`, `referencedComponentId`, `sctId1`)
SET id=UNHEX(REPLACE(@uid,'-',''));
```

Figure 81: Import Component Refset File

7.2 Importing SNOMED CT release data

7.2.1 Choosing a Release Type to import

The first step in selecting the set of *release files* to be imported is to decide which *Release Type* will be used. *SNOMED CT Release Format 2* specifies three distinct *Release Types*: *full release*, *snapshot release* and *delta release*. These are described in the table below.

The *Release Format 2* specification states that:

- The *SNOMED CT International Release* will include all three *Release Types*;
- A *SNOMED CT Extension Release* must include the *full release*;
- A *SNOMED CT Extension Release* may optionally include a *snapshot release* and/or *delta release*.

A *SNOMED CT-enabled terminology server* must be able to import data from a *full release* because this is the only *Release Type* that is required to be produced by all *Extension* developers. A *SNOMED CT-enabled terminology server* should also be able to import from other *Release Types* where these are available as these may allow more efficient updating.

The choice of a particular *Release Type* depends on the type of *terminology views* that the *terminology server* is designed to support and on whether this is an initial import or a subsequent update.

Note: The requirement to be able to import data from the *full release* does not mean that all *terminology servers* must provide access to the complete historical set of data provided by a *full release*. The *full release* can be selectively imported to used to populate a *snapshot view* for applications that do not require access to historical data.

7.2.1.1 Release Types

Table 193 specifies the content of each of the *Release Format 2 Release Types*.

This table is followed by illustrations of each of the *Release Types* using the small same pattern of content development over seven release cycles. These illustrations highlight the key differences and the *Relationships* between the *Release Types*.

Table 193: SNOMED CT Release Types

Release Type	***Description***
Full	The files representing each type of component contain **every** version of every component ever released.
Snapshot	The files representing each type of component contain **one** version of every component released up to the time of the snapshot. The version of each component contained in a snapshot is the most recent version of that component at the time of the snapshot.
Delta	The files representing each type of component contain only component versions created since the previous release. Each component version in a *delta release* represents either a new component or a change to an existing component.

The seven columns in each of the following illustrations represent the content of seven releases (numbered 1-7). Each *component* is identified by a letters (A-K). A *component* version is represented by the identifying letter followed by a number (1-7) representing the release cycle in which that *component* version became effective.

Figure 82 shows the content of a series of *full releases*. The yellow background color highlights the set of *component* versions that are also present in the snapshot for the same *release version* (see *Figure 84*). *component* versions are shown in gray in releases versions after they have been superseded by a new *component* version. Newly added *component* versions, shown in red, are also present in the delta for the same *release version* (see *Figure 83*).

The content of the *full release* in any chosen version is identical to the combined content of all the *snapshot releases* up to and including that version. Thus adding a *delta release* to the previous version of the *full release* creates the *full release* for the new version. The *snapshot release* is derived from the *full release* by removing all except the most recent version of each *component*.

Earlier<-- FULL RELEASES FOR SEVEN RELEASE CYCLES --> Later						
1	2	3	4	5	6	7
A,1	A,1	A,1	A,1	A,1	A,1	A,1
B,1	B,1	B,1	A,4	A,4	A,4	A,4
C,1	B,2	B,2	B,1	B,1	B,1	B,1
D,1	C,1	C,1	B,2	B,2	B,2	B,2
E,1	D,1	C,3	C,1	C,1	B,6	B,6
	E,1	D,1	C,3	C,3	C,1	C,1
	F,2	E,1	C,4	C,4	C,3	C,3
	G,2	F,2	D,1	D,1	C,4	C,4
		F,3	E,1	E,1	C,6	C,6
		G,2	F,2	F,2	D,1	D,1
		H,3	F,3	F,3	E,1	E,1
			G,2	G,2	F,2	F,2
			H,3	H,3	F,3	F,3
			I,4	I,4	G,2	G,2
				J,5	H,3	H,3
					H,6	H,6
					I,4	H,7
					J,5	I,4
						J,5
						K,7

Figure 82: Full release illustration

Earlier<-- DELTA RELEASES FOR SEVEN RELEASE CYCLES --> *Later*						
1	2	3	4	5	6	7
A,1			A,4			
B,1	B,2				B,6	
C,1		C,3	C,4		C,6	
D,1						
E,1						
	F,2	F,3				
	G,2					
		H,3			H,6	H,7
			I,4			
				J,5		
						K,7

Figure 83: Delta release illustration

Earlier<-- SNAPSHOT RELEASES FOR SEVEN RELEASE CYCLES -->*Later*						
1	2	3	4	5	6	7
A,1	A,1	A,1	A,4	A,4	A,4	A,4
B,1	B,2	B,2	B,2	B,2	B,6	B,6
C,1	C,1	C,3	C,4	C,4	C,6	C,6
D,1	D,1	D,1	D,1	D,1	D,1	D,1
E,1	E,1	E,1	E,1	E,1	E,1	E,1
	F,2	F,3	F,3	F,3	F,3	F,3
	G,2	G,2	G,2	G,2	G,2	G,2
		H,3	H,3	H,3	H,6	H,7
			I,4	I,4	I,4	I,4
				J,5	J,5	J,5
						K,7

Figure 84: Snapshot release illustration

Note: In a real *SNOMED CT release* each of the letters A-K would be replaced by a *component* id (a *SNOMED CT identifier*) and each of the release cycle numbers 1-7 would be replaced by the *effectiveTime* of a *release version*.

7.2.1.2 Importing and maintaining a Full view

7.2.1.2.1 Importing a Full view

To provide access to the *full view* of the content of the *SNOMED CT International Release*, a *terminology server* must initially import content from the *full release* files for the *International Release*.

The complete content of all the main *release files* should be imported into the chosen internal representation.

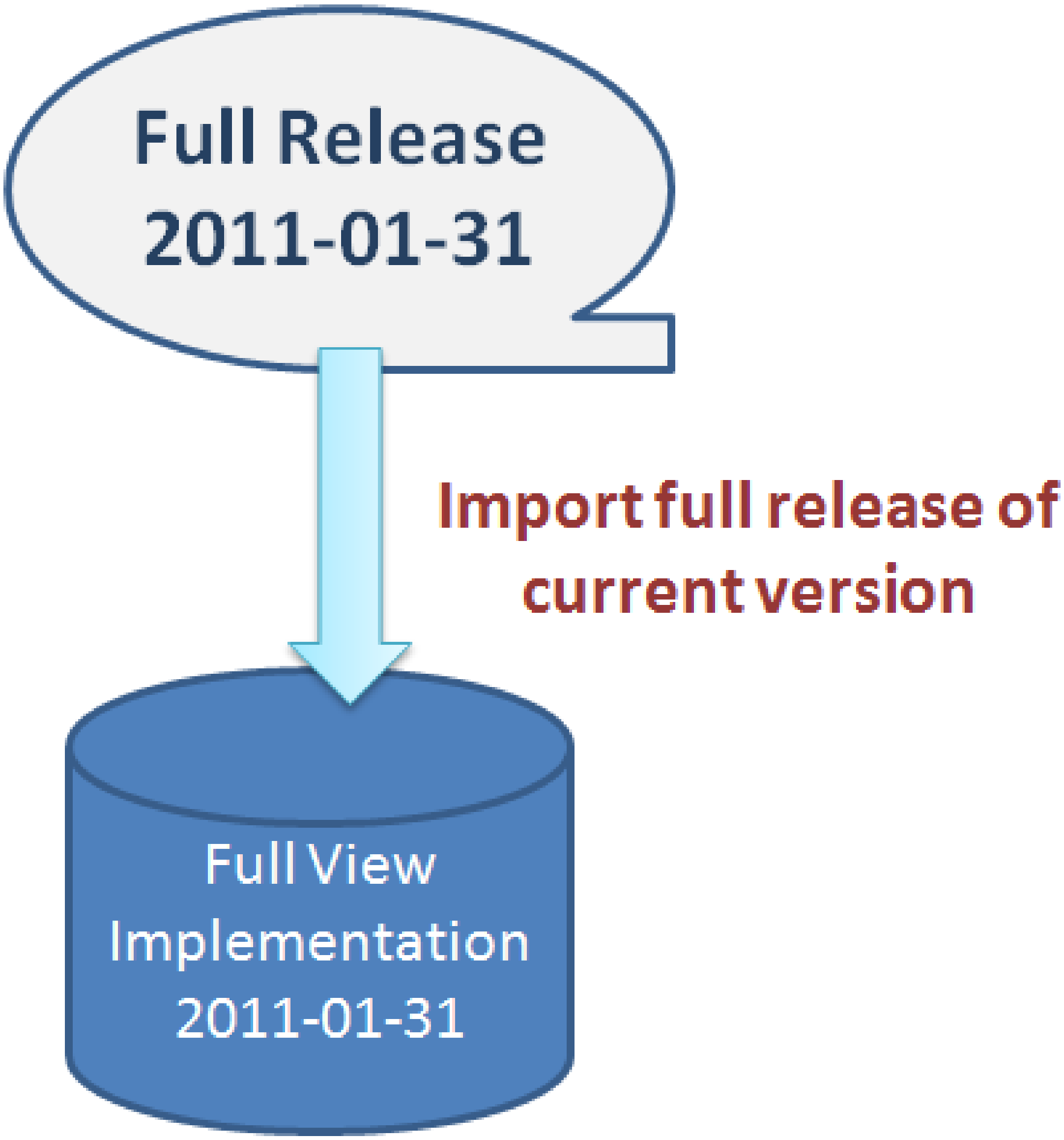


Figure 85: Initial import to create a full view

Tip: The files that form part of a particular release can be identified by pattern matching based on the *IHTSDO* filenaming conventions (see *Identifying release files using regular expressions*).

7.2.1.2.2 Updating a Full view

A *full view* can be updated by one of the following approaches:

1. Append the content of the relevant *delta release* files to a previously created *full view*:

- The delta files contain only the changes since a previous release. Appending the data from these files to the *full view* for the previous version creates the *full view* for the new version. There is no need to change or delete existing data.

Caution: A *delta release* must be applied to the immediately previous version. Appending a *delta release* to earlier versions will result in omission of content and this will lead to significant errors when interpreting the data.

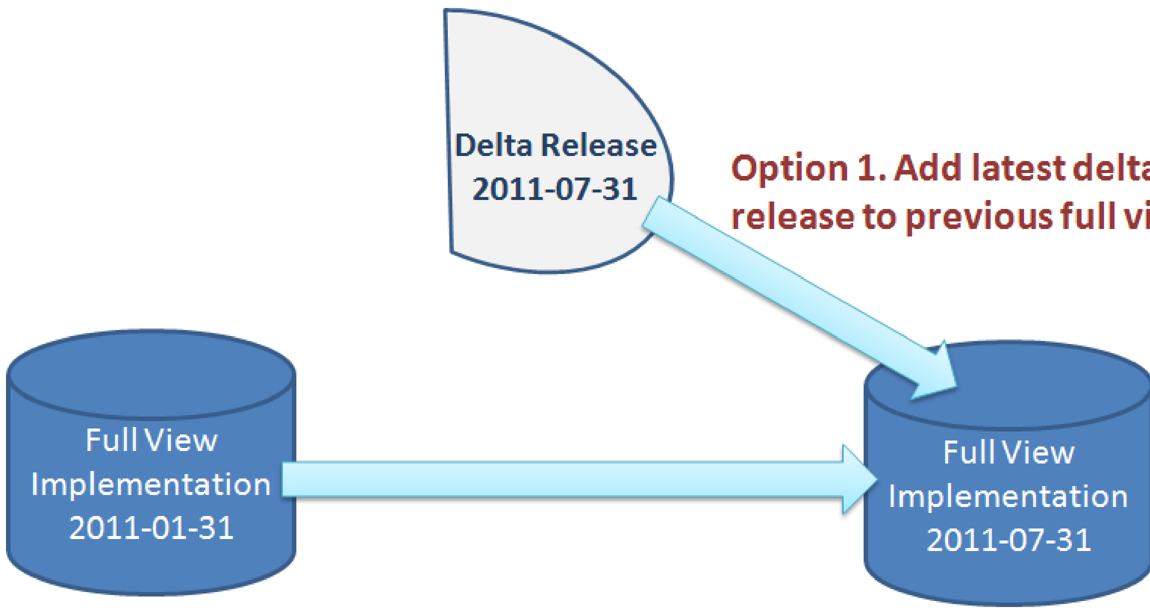


Figure 86: Updating a full view using a snapshot release

2. Filter the relevant *full release* files to generate a *delta release* then apply this as in 2 above:

- The *delta release* consists of all items in the *full release* with an *effectiveTime* greater than the *effectiveTime* of the most recent previous release. Therefore, it is easy to filter a *full release* to generate a set of *delta release* files.
- Alternatively a "virtual delta" release may be used by filtering the *full release* while importing.

  Note: This allows *Extensions* that are not distributed with *delta releases* to be processed by a general update process that is optimized to work with *delta releases*.

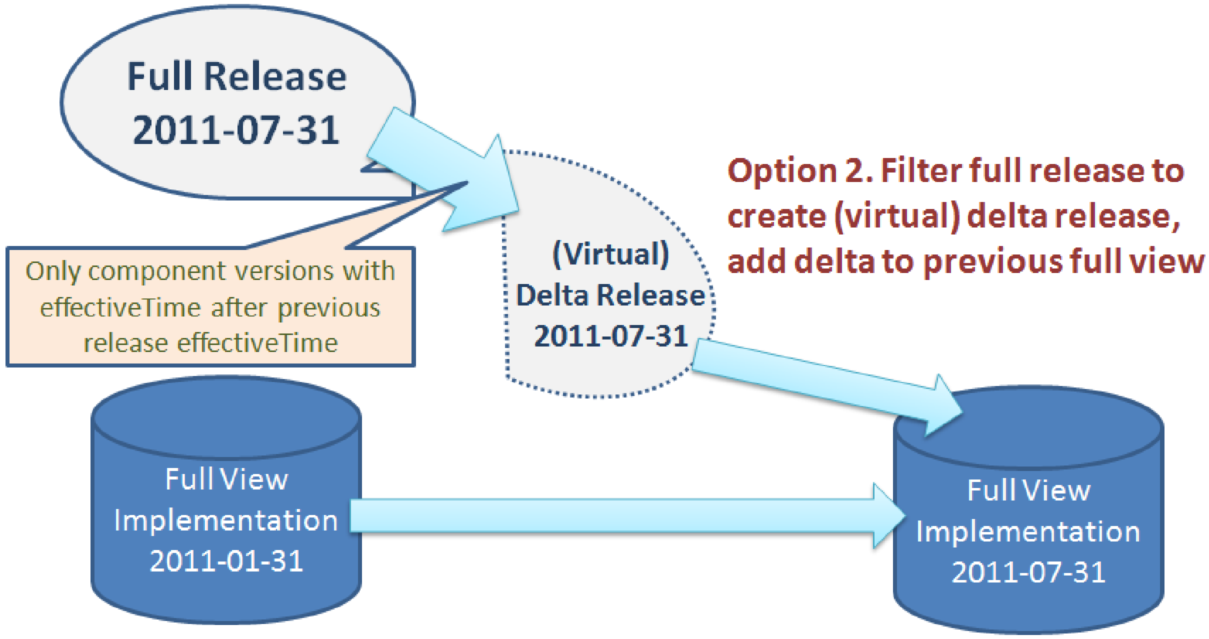


Figure 87: Updating a full view using a full release

3. Use the *full release* files to completely replace previously imported data:
   - This follows the approach described earlier to *import a full view*.

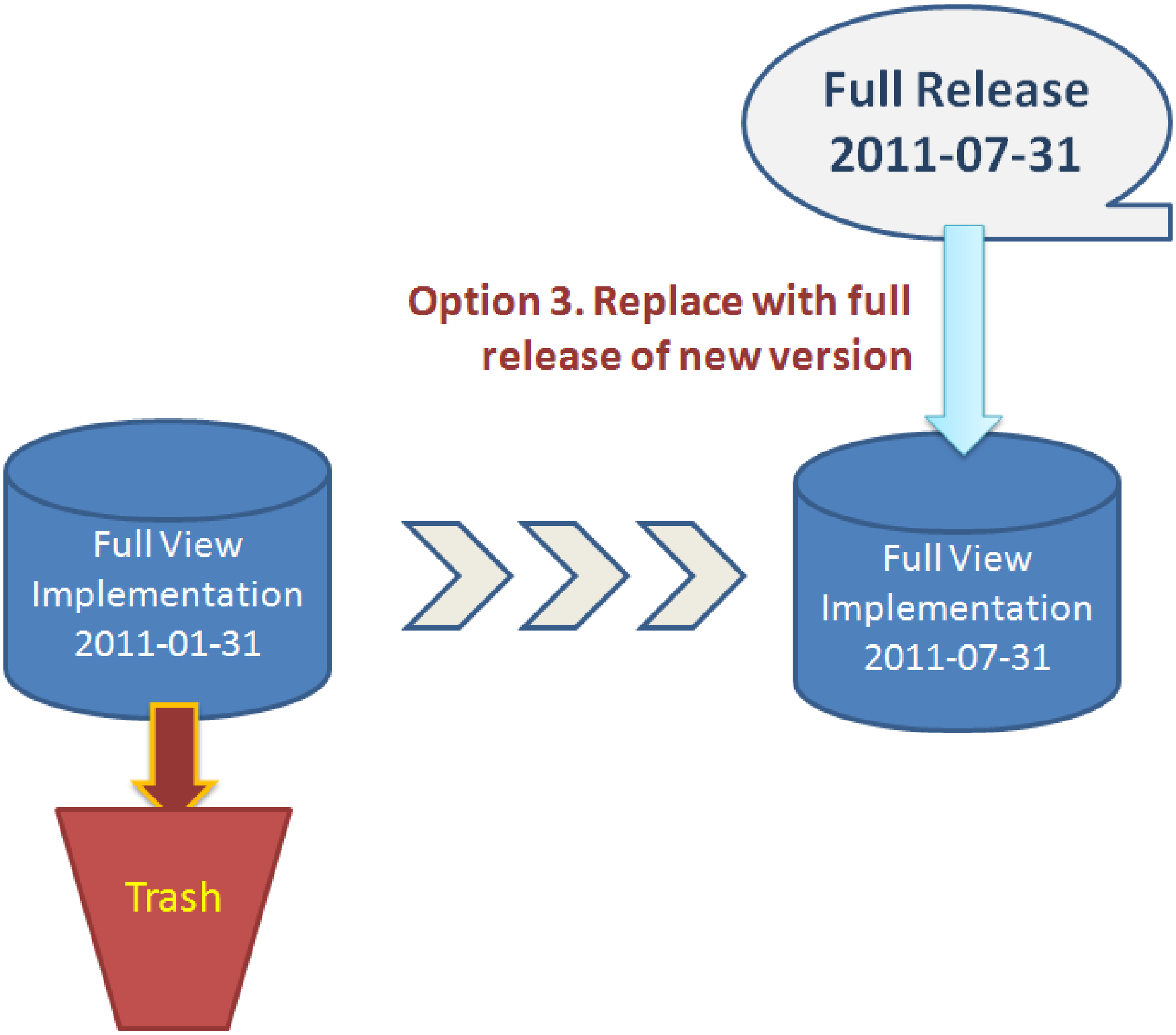


Figure 88: Updating a full view by replacement

Tip: The files that form part of a particular release can be identified by pattern matching based on the *IHTSDO* filenaming conventions (see *Identifying release files using regular expressions*).

7.2.1.2.3 Importing and updating Extensions for a Full view

To provide access to the *full view* of one or more *Extensions*, a *terminology server* must initially import content from the *full release* files for each of the required *Extensions*. Thereafter, the *full view* of each *Extension* can be maintained by using any of the techniques described for *updating a full view*.

When a *full view* of *Extension* data is initially imported or subsequently updated care needs to be taken to ensure the relevant versions of the *International Release* and any other *Extensions* on which it depends have been imported. Failure to follow do this may lead to errors as a result of references from *Extension components* to missing or out of date *components* in the *International Release* or in another *Extension*.

A *full view* may include more recent versions of the *International Release* than is required to support the *Extension*. In this case, when the *Extension* is viewed the *International Release* can, if necessary, be viewed as it would have been in the version to which the *Extension* is related. Similarly, if one *Extension* depends on content in another *Extension*, the version the *Extension* on which it depends may be a more recent version.

The table below summarizes the compatibility between the *full views* of given versions of an *Extension* and the *International Release*. It also indicates the ways in which a *full view* may be used when the latest installed versions are not directly compatible. If one *Extension* depends on another *Extension*, the same considerations apply to compatibility between the versions of those *Extensions*.

Table 194: Compatibility between full views of versions of an Extension and the International Release

Relationship* between the version of the *Extension* and the *International Edition	**Notes on compatibility and usability**
Installed *International Release* is **older** than the version on which the *Extension* was based	**Incompatible - unless recent *Extension* content is excluded**. The *Extension* may include *Relationships* to *concepts* that do not exist in this version of the *International Release*. This will lead to errors that cannot be reconciled while viewing the *Extension* content. A system with this mix of installed versions could be safely used by excluding the content of the more recent *Extension* versions. This can be done by excluding any *Extension component*-version with an *effectiveTime* of one of the versions based on a newer *International Release*. In effect this approach rolls back the *Extension* to the last *Extension* that is valid with the installed version of the *International Release*.
Installed *International Release* is **same** version as the one on which the *Extension* was based	**Fully compatible**. This is the version the *Extension* was created for so it should behave as intended.

Relationship* between the version of the *Extension* and the *International Edition	**Notes on compatibility and usability**
Installed *International Release* is **newer** than the version on which the *Extension* was based.	**Compatible - subject to appropriate configuration and usage**. The *International Edition* for this version may include: • Additional *components*. These will not cause errors because the *International Release* does not reference the *Extension* and the *Extension* content cannot reference *components* that did not exist when in the version it was based on. • Changes to the state of some *components*. These changes may affect the interpretation of some parts of the *Extension*. However, despite these issues the *full view* resulting from this combination can be used in several ways: **1.** Configured to roll-back the *International Release* to the version on which the *Extension* was based. This can be done with a *full release* by creating a virtual view of the *International Release components* which excludes *component* versions with an *effectiveTime* greater than the version on which the *Extension* was based. This type of view is described in more detail in *Implementing the State-Valid view*. **2.** Configured to exclude the *Extension*. In this case the most recent version of the *International Release* can be viewed. **3.** Configured to use those parts of the *Extension* that support translation of *International Release* content. In this case, the *Extension* will enable translated rendering of pre-existing translated content. This would leave new and untranslated *concepts* to be rendered in English (or another available language). **4.** Accepting and working within the constraints imposed by the omissions and anomalies noted above. This mode should not be used routinely but may useful for assessing the impact of changes to the *International Release* on the *Extension*.

Note: The compatibility and usability notes are specific to a *full view* implementation. Different considerations apply to snapshot and multi- *snapshot views*.

7.2.1.3 Importing and maintaining a Snapshot view

7.2.1.3.1 Importing a Snapshot view

To provide access to the *snapshot view* of the content of the *SNOMED CT International Release*, a *terminology server* must initially do one of the following:

1. Import the content from a set of *snapshot release* files for a version of the *International Release*:
    - All the rows from a set of *snapshot release* files must be imported.

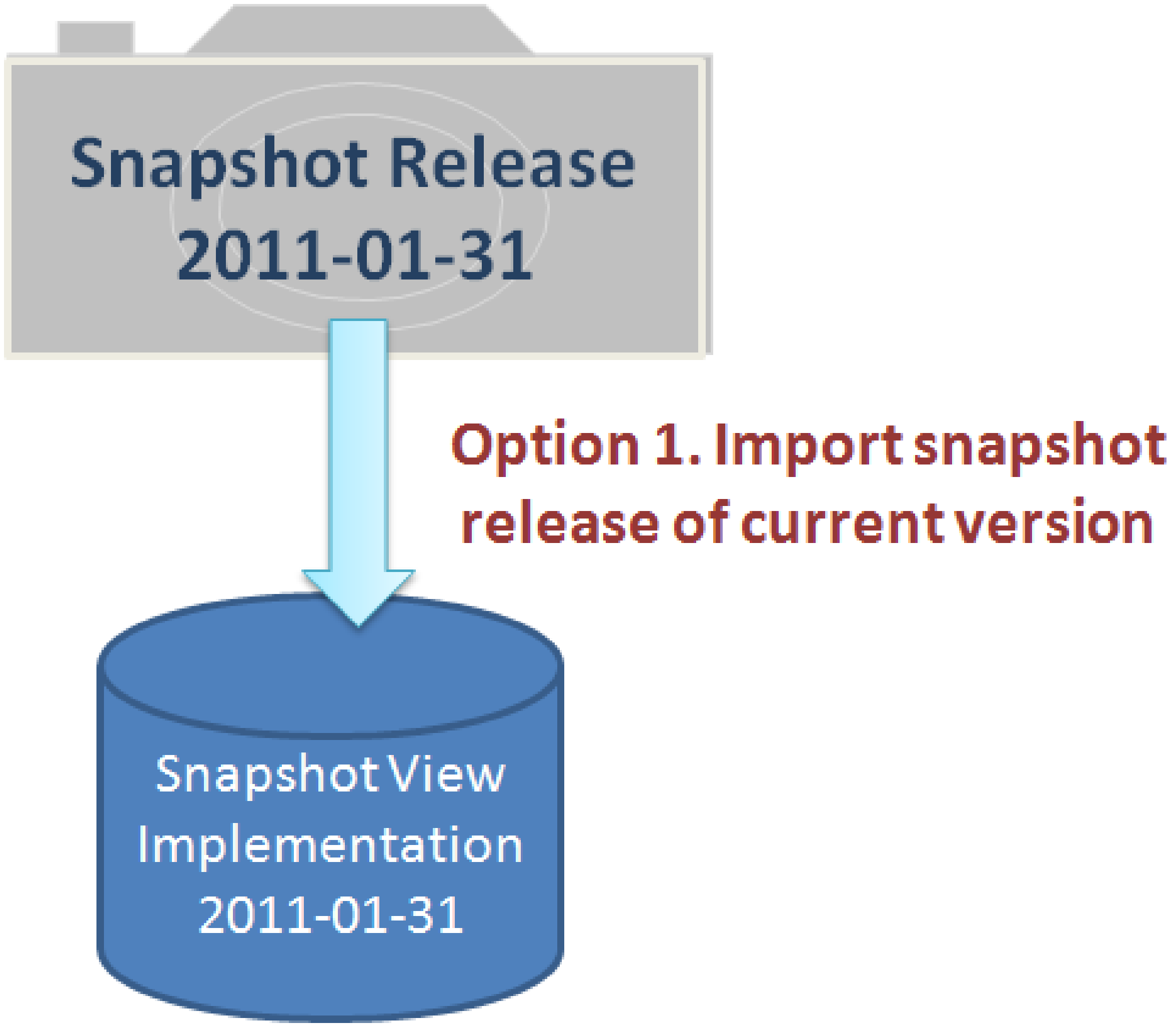


Figure 89: Initial import to create a snapshot view

2. Create a snapshot from a set of *full release* files for a version of the *International Release*:
    - When creating a snapshot from a *full release*, only those rows that represent the most recent version of each *component* are imported.
    - Where two or more rows have the same id, only the row with the most recent *effectiveTime* is imported into the snapshot.

    Caution: Only take account of the id and *effectiveTime* fields when determining which rows to import into a snapshot. A common mistake is to look for the most recent active row. This results in serious errors. The active field should only be considered after importing the data and then provides information on whether that *component* is or is not active as part of the snapshot.

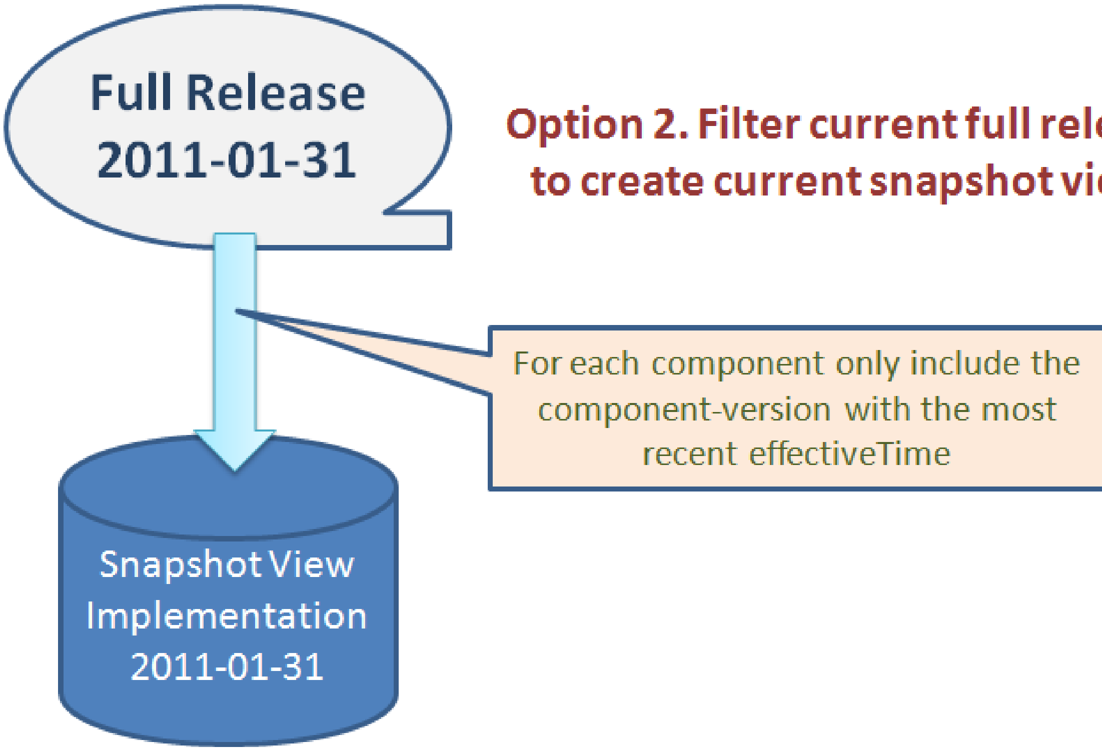


Figure 90: Initial import to create a snapshot view from a full release

Note: Option 2 can also be used to create an earlier *snapshot view*. To do this only import rows that represent the most recent version of each *component* with an *effectiveTime* that is no later that the time of the required snapshot.

Tip: The files that form part of a particular release can be identified by pattern matching based on the *IHTSDO* filenaming conventions (see *Identifying release files using regular expressions*).

7.2.1.3.2 Updating a Snapshot view

A *snapshot view* can be updated by one of the following approaches:

1. Use a set of *delta release* files to update the a *snapshot view* of the previous version:
    - The overall process can be described as follows:
        - Append the *delta release* to the previous snapshot;
        - Filter to remove rows that have the same id so that only the row with the most recent *effectiveTime* remains.
    - An efficient way to achieve this end result is to take account of the fact that the most recent version of any given *component* will be in the new *delta release* rather than in the previous version of the *snapshot view*.

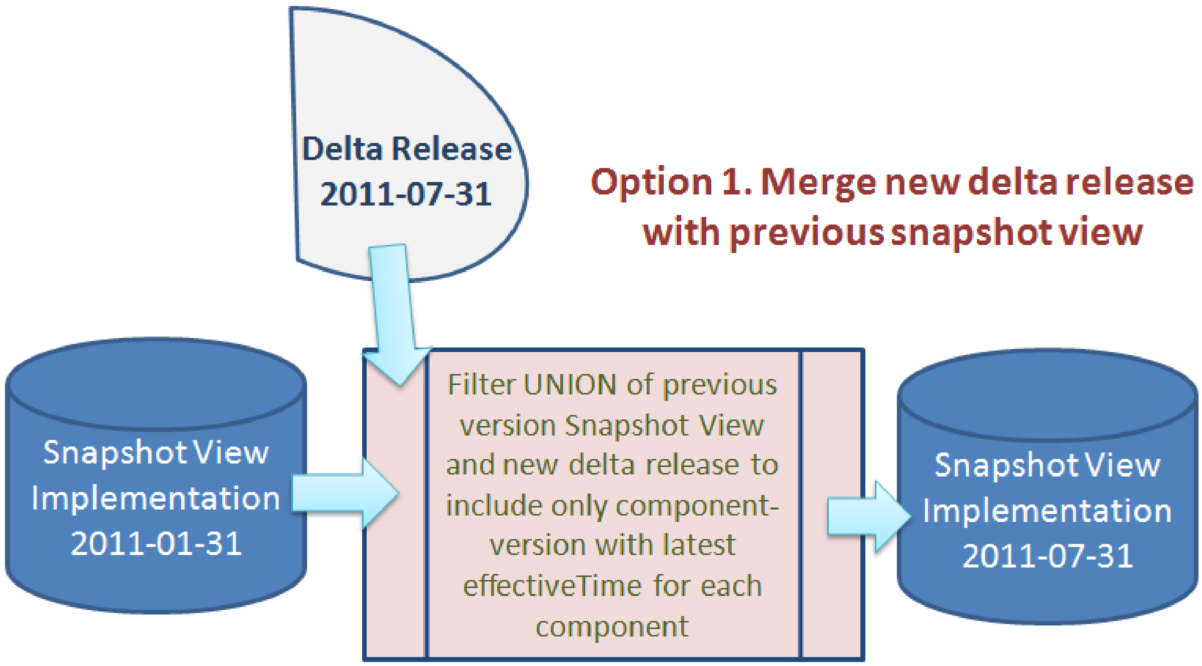


Figure 91: Updating a snapshot view using a delta release

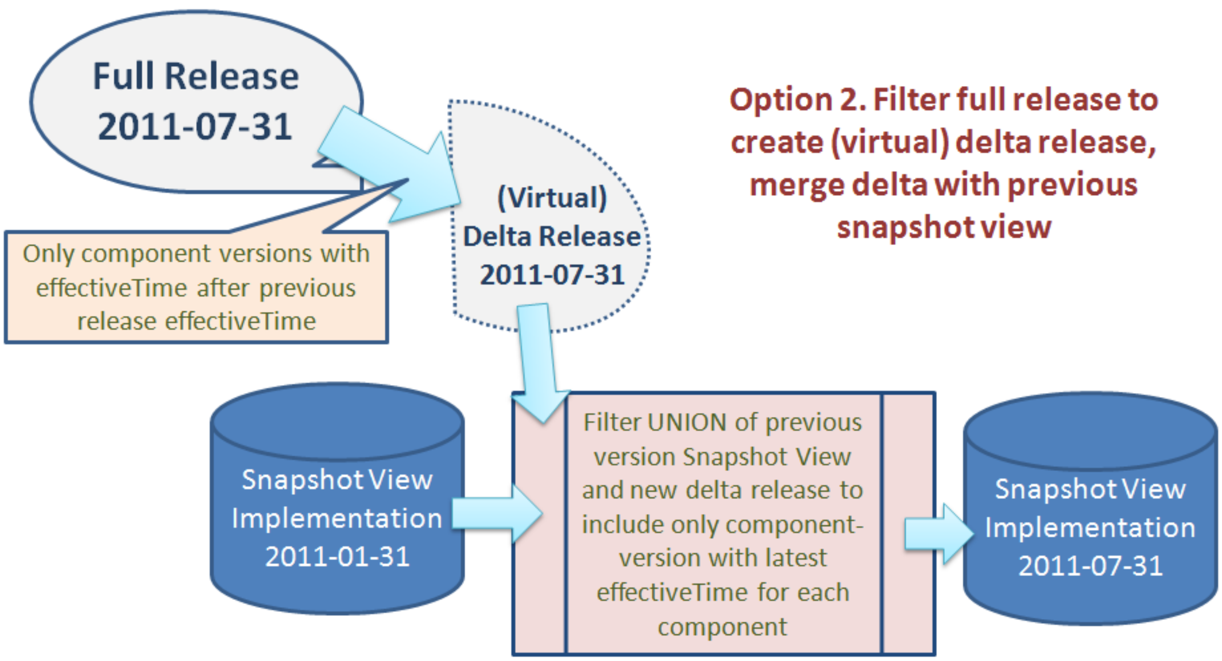


Figure 92: Updating a snapshot view using a full release

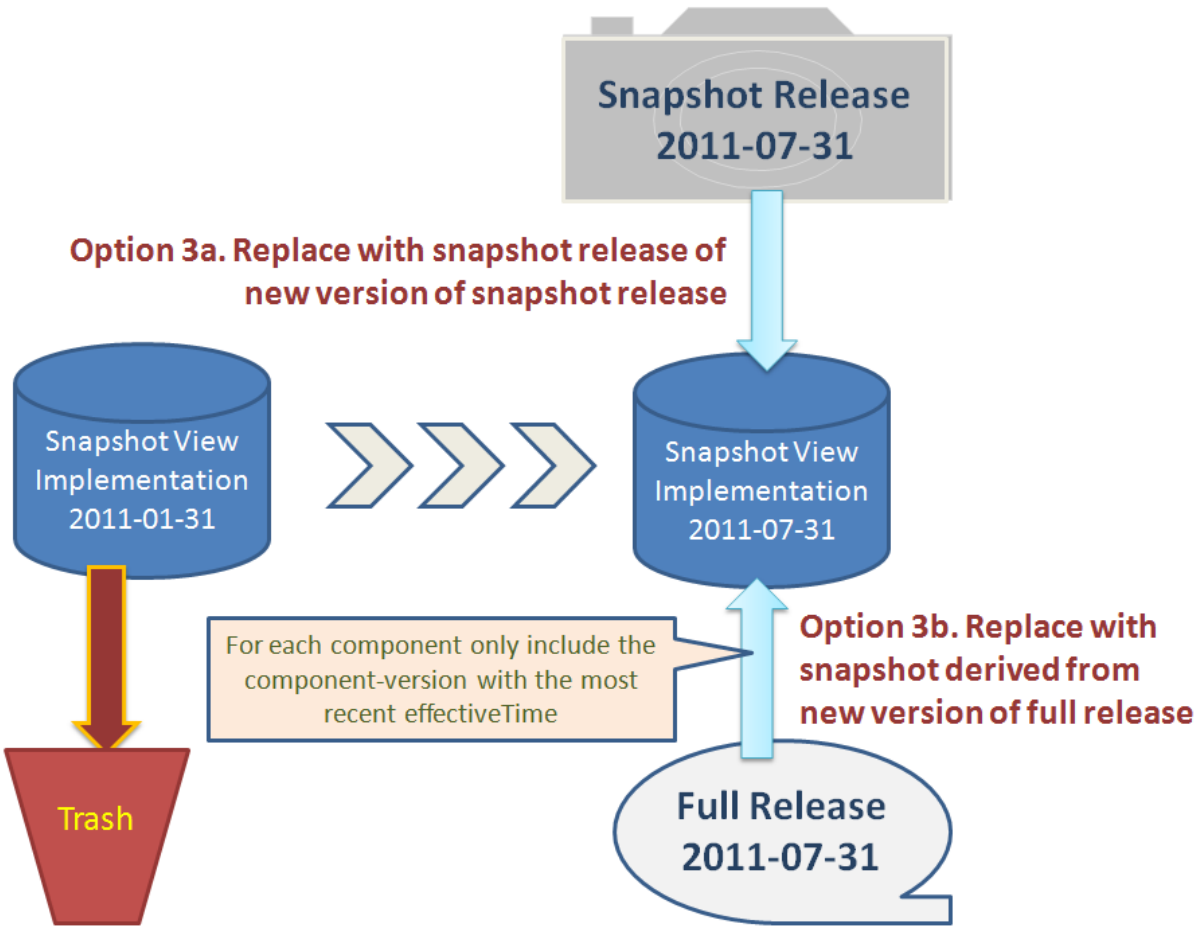


Figure 93: Updating a snapshot view by replacement

Tip: The files that form part of a particular release can be identified by pattern matching based on the *IHTSDO* filenaming conventions (see *Identifying release files using regular expressions*).

7.2.1.3.3 Importing and updating Extensions for a Snapshot view

To provide access to the *snapshot view* of one or more *Extensions*, a *terminology server* must initially import or create the current snapshot for each required *Extension*. This can be done either using *snapshot release* files or *full release* files as described for *importing a Snapshot view*. Thereafter, the *snapshot view* of each *Extension* can be maintained by using any of the techniques described for *updating a Snapshot view*

When a *snapshot view* of *Extension* data is initially imported or subsequently updated care needs to be taken to ensure the relevant versions of the *International Release* and any other *Extensions* on which it depends have also been imported or updated. Failure to follow do this will lead to errors as a result of references from *Extension components* to missing or out of date *components* in the *International Release* or in another *Extension*.

For all normal uses the *snapshot view* of an *Extension* version must be combined with the *snapshot view* of the versions of the *International Release* on which it was based. Similarly, if an *Extension* is dependent on another *Extension*, the snapshot of the *Extension* on which it depends must be for the version on which the dependent *Extension* version was based.

The table below summarizes the compatibility between the *snapshot views* of given versions of an *Extension* and the *International Release*. It also identifies some limited cases in which a *snapshot view* may be used when not directly compatible with the relevant *International Release* snapshot. If one *Extension* depends on another *Extension*, the same considerations apply to compatibility between the versions of those *Extensions*.

Table 195: Compatibility between snapshot views of versions of an Extension and the International Release

Relationship* between the version of the *Extension* and the *International Edition	**Notes on compatibility and usability**
Installed *International Release* is **older** than the version on which the *Extension* was based	**Incompatible**. The *Extension* may include *Relationships* to *concepts* that do not exist in this version of the *International Release*. This will lead to errors that cannot be reconciled while viewing the *Extension* content. Excluding *components* that conflict in this way results in other errors the previous state of the *Extension* content is not available in a *snapshot view*. A system with this mix of snapshot versions cannot be safely used.
Installed *International Release* is **same** version as the one on which the *Extension* was based	**Full compatible**. This is the version the *Extension* was created for so it should behave as intended.
Installed *International Release* is **newer** than the version on which the *Extension* was based.	**Partially compatible - subject to appropriate configuration and usage**. The *International Edition* for this version may include: • Additional *components*. These will not cause errors because the *International Release* does not reference the *Extension* and the *Extension* content cannot reference *components* that did not exist when in the version it was based on. • Changes to the state of some *components*. These changes may affect the interpretation of some parts of the *Extension*. In a *snapshot view* the *International Release* cannot be rolled back to its previous state and, as a result, cannot be aligned with the version on which the *Extension* was based. Therefore, the potential for safe use of combinations of this is limited to the following: **1.** Configured to exclude the *Extension*. In this case the most recent snapshot of the *International Release* can be viewed. The incompatible *Extension* is ignored. **2.** Configured to use those parts of the *Extension* that support translation of *International Release* content. In this case, the *Extension* will enable translated rendering of pre-existing translated content. This would leave new and untranslated *concepts* to be rendered in English (or another available language).

Note: A *full view* implementation can be configured to be more tolerant to different versions of installed *Extensions* and *International Releases*. In effect the *full view* allows virtual snapshots of the state of each *Extension* to be used to deliver a compatible set of *component*-versions.

7.2.1.4 Maintaining a Multi-snapshot view

If more than one *snapshot view* is required, the most effective approach is to implement a *full view* that enables a *dynamic snapshot* to be provided for any chosen time. The alternative approach is to create several separate *snapshot views* and to allow users to choose and where necessary switch between these static snapshots.

Each of these views in a multi-*snapshot view* is separately created and maintained in the same way as a single *snapshot view*. The required view for a particular purpose is selected from those available in the server. Where necessary more than one view may be selected to identify changes between versions.

In the long-term this approach requires more maintenance effort and more storage space than a *full view* and is far less flexible. It assumes a small set of discrete views such as those that arise from a relatively infrequent releases of *SNOMED CT* content. A more gradual evolution of content may occur in future as a result of the additions to *Extensions* and the ability to distribute *delta releases*. The multiple-snapshot approach may still meet the limited requirements of an organization needing access to two or three specified *snapshot views* (e.g. for current, previous and perhaps one other defined reference date). This approach may be useful as an interim measure in an environment that is unable to provide adequate performance for *dynamic snapshot views*.

7.2.2 Choosing the release files to import

The *International Release* files to be imported should all be selected from the set of files representing a single *Release Type* for a chosen version of the *SNOMED CT International Release*.

Within the chosen file set the files identified in *Table 196* must be imported. The files listed in *Table 197* should also be imported as these provide important information about *inactive concepts* and metadata about *Description* types. The decision on whether to import the files listed in *Table 198* depends on whether the additional features identified in that table are required for the planned implementation. Finally the supplementary files listed in *Table 199* may be used to assist implementation but are not essential as the data they contain can be generated from the other files and/or replaced by alternative approaches to provide similar functionality.

Table 196: Mandatory import files

File type	Content	Notes
sct2_concept_ *[rt]* _INT...	*concepts*	The primary *components* of *SNOMED CT*. Essential for all implementations.
sct2_description_ *[rt]* _INT...	*Descriptions*	
sct2_relationship_ *[rt]* _INT...	*Relationships*	
sct2_cRefset_Language *[rt]*- *[lang]* _INT...	Language *Refset*(s)	At least one Language *Refset* must be imported. The English Language *refset* should be imported unless another Language *refset* covering the full content is available and imported.

Table 197: Highly recommended import files

File type	Content	Notes
der2_cRefset_AttributeValue *[rt]* _...	*concept* Inactivation *refset*	Provides information about the status of *inactive concepts*.
	Description Inactivation *Refset*	Provides information about the status of *inactive descriptions*.

File type	Content	Notes
der2_cRefset_DescriptionType *[rt]* _INT...		
der2_cRefset_AssociationReference *[rt]* _INT...		

Table 198: Optional import files

File type	Content	Notes
der2_sRefset_SimpleMap *[rt]* _INT...	Maps from *NHS Clinical Terms Version 3* codes, other *Read Codes* to *SNOMED CT*	Only required if the server needs to be able to lookup *SNOMED CT concepts* based on a *CTV3 Identifier* or *Read Code*.
	Maps from legacy *SNOMED CT* 3 codes to *SNOMED CT*	Only required if the server needs to be able to lookup *SNOMED CT concepts* based on a legacy *SNOMED CT* 3 code.
WordEquivalents		
StatedRelationships		

Table 199: Supplementary import files

File type	Content	Notes
DescWordKey		
DescDualKey		
ExcludedWords		
TransitiveClosure		Generated from the *Relationships*. Needs to be regenerated or updated if *Extensions* are imported.

If *Extensions* are required to support an implementation, the *release files* to be imported should be selected from the set of files for a single *Release Type* for a chosen version of that *Extension*. It is important to ensure that the *International Release* version(s) on which all the imported *Extensions* are based has also been imported. The files that need to be imported from a chosen *Extension* may vary depending on the scope of the *Extension*.

Note: Advice should be sought from the *Extension* provider on the essential and recommended requirements of files to be imported and supported.

7.2.3 Choosing extension files to import

The process of importing an *Extension* is similar to importing the main distribution files. However, some additional functionality is required to ensure appropriate installation, maintenance and use of *Extensions*.

Applications should:

- Allow the users or user communities to specify the *Extensions* to be recognized by their systems. Before recognizing any *Extension*, users should check that:
  - The *Extension* has been supplied by the *IHTSDO* or another organization authorized by the *IHTSDO* to provide such *Extensions*.
  - You are satisfied with the quality control procedures of the providing organization:
    - Authorization of an organization to produce *Extensions* does not imply any seal of approval related to the quality of *Extensions* provided by those organizations;
    - Installation of *Extensions* is done entirely at the risk of the user subject to their license agreement with the provider of the *Extension* and/or the application developer.

7.2.4 Identifying release files using regular expressions

The files that form part of each release follow *IHTSDO* file naming conventions. These conventions allow the files that form part of a particular *Release Type*, version or extension to be identified by pattern matching. The following tables include examples of *standard regular expressions* that selectively match particular sets of *release files*.

Table 200: General patterns for Release Types

<table>
<tr><th>Release</th><th>Type</th><th>Regular Expression</th></tr>
<tr><td>International</td><td>full</td><td>^x?(sct|der|res)2_[^_]+_[^_]*Full(-[a-z-]{2,6})?_INT_2[0-9]{7}.txt$</td></tr>
<tr><td>International</td><td>delta</td><td>^x?(sct|der|res)2_[^_]+_[^_]*Delta(-[a-z-]{2,6})?_INT_2[0-9]{7}.txt$</td></tr>
<tr><td>International</td><td>snapshot</td><td>^x?(sct|der|res)2_[^_]+_[^_]*Snapshot(-[a-z-]{2,6})?_INT_2[0-9]{7}.txt$</td></tr>
<tr><td>Any Extension</td><td>full</td><td>^x?(sct|der|res)2_[^_]+_[^_]*Full(-[a-z-]{2,6})?_([A-Z]{2})?[0-9]{7}_2[0-9]{7}.txt$</td></tr>
<tr><td>Any Extension</td><td>delta</td><td>^x?(sct|der|res)2_[^_]+_[^_]*Delta(-[a-z-]{2,6})?_([A-Z]{2})?[0-9]{7}_2[0-9]{7}.txt$</td></tr>
<tr><td>Any Extension</td><td>snapshot</td><td>^x?(sct|der|res)2_[^_]+_[^_]*Snapshot(-[a-z-]{2,6})?_([A-Z]{2})?[0-9]{7}_2[0-9]{7}.txt$</td></tr>
</table>

Table 201: Example patterns for specific versions and extensions

Release	**Regular *Expression***
International Type: full Version: 2010-07-31	^x?(sct|der|res)2_[^_]+_[^_]*Full(-[a-z-]{2,6})?_INT_20100731.txt$
International Type: full Version: 2011-01-31	^x?(sct|der|res)2_[^_]+_[^_]*Full(-[a-z-]{2,6})?_INT_20110131.txt$
Member Extension Type: full Country: GB Namespace: 1000001 Version: 2011-04-01	^x?(sct|der|res)2_[^_]+_[^_]*Full(-[a-z-]{2,6})?_GB1000001_20110401.txt$
Affiliate Extension Type: full Namespace: 1000003 Version: 2011-07-31	^x?(sct|der|res)2_[^_]+_[^_]*Full(-[a-z-]{2,6})?_1000003_20110731.txt$

7.2.5 Checking during the import process

The import process should check the imported data to confirm that:

- The distribution files imported are all part of the same release.
- The set of files imported is complete and includes all mandatory components.
- In the case of a *delta release*, the data previously imported is the version immediately prior to the *delta release* being imported.
- In the case of a snapshot or *full release*, pre-existing data has been removed:
  - Alternatively the import process may be configured to overwrite duplicate rows so that:
    - The end result of a snapshot import does not contain any obsolete rows;
    - The end result of a *full release* import is identical to the content of the *full release*.
- All component *Identifiers* have:
  - A *partition identifier* appropriate to the type of component;
  - A valid *check-digit*.
- All fields meet data type, size and value constraints specified for the relevant tables.

Other consistency checks may also be applied to ensure the integrity of the data.

7.2.5.1 Additional checks when importing Extensions

The process of importing an *Extension* is similar to importing the main distribution files. However, some additional functionality is required to ensure appropriate installation, maintenance and use of *Extensions*. Applications should:

Check each *Extension* prior to installation to ensure that:

- It is one of the *Extensions* recognized by the user.
- It is supported by or based on the currently installed *International Release* version.
- The required versions of other *Extensions* on which this *Extension* depends have already been installed (or have been selected for installation as part of the same import process).
- Any dependencies of the *Extension* have been met. These dependencies may include:
  - Installation of a particular *SNOMED CT release*;
  - Prior installation of other *Extensions*.

  Note: Dependencies are represented using the *moduleId* and the *Module Dependency Reference Set*.

- The installation procedure has pre-checked all *components* in the *Extension* to ensure that:
  - All Component *Identifiers*:
    - Are unique;
    - Have a *partition identifier* appropriate to the type of component;
    - Have a *namespace Identifier* appropriate to the provider of that *Extension*
    - Have a valid *check-digit*
  - All fields meet data type, size and value constraints specified for the relevant tables.

Caution: If any *components* fail any of these tests the entire *Extension* must be rejected. Rejecting individual components is liable to lead to inconsistent data. Accepting data that fails these test may create conflicts between different *Extensions* or between the *Extension* and the *International release*.

- Reject, highlight or apply other agreed business rules to information received by the system that contains *SCTIDs* for *components* from namespaces that are not in the list, or recognized *Extensions*.

7.2.6 Pre-processing of distribution files by terminology server suppliers

The import process may be time-consuming due to the need to build indices or other data structures. It may also require substantial spare storage capacity for temporary files. Therefore a *terminology server* provider may choose to pre-import the distribution files and provide them to users in pre-prepared form. However, an import facility should also be available in a suitably secured form to end-user organizations, to enable installation and maintenance of locally required *Extensions*.

7.3 Implementing Dynamic Snapshot Views

A key feature of *SNOMED CT Release Format 2* is that it allows a single database table to represent the *full view* of a *SNOMED CT component*. This view includes all versions of the *component* from its first release up to its state in the latest release. This offers several significant benefits which are described elsewhere in the guide.

Most frequently used *SNOMED CT* functions need to provide access to a 'snapshot' view of the content of *SNOMED CT* at a point in time.

- Everyday use of *SNOMED CT* for data entry and retrieval will generally require a current 'snapshot' view.

  Example: To see the active content of *SNOMED CT* including all the most up to date *components* and excluding any *components* that have been marked as inactive.

- There are some situations in which a retrospective 'snapshot' view of the data at a selected point in the past is required.

  Example:

  To see the definition of a *concept* as it was when a record entry was created.

  To see the version of the *International Release* on which the latest available version of an *Extension* was based.

7.3.1 Generating Dynamic Snapshot Views

The general method for creating a snapshot 'view' for a specified SnapshotTime is as follows:

1. Exclude all *Component* versions with an *effectiveTime* greater than the SnapshotTime.

   Note: In theory the most recent *snapshot view* step could be omitted. However, a release will often be distributed before its *effectiveTime*. Therefore, this approach is not recommended as a general approach in a live system.

2. From each set of *Component* versions with the same id select the *Component* version with the highest (most recent) *effectiveTime*.

The most flexible approach is to apply this method dynamically so that a different snapshot time can be configured as needed to meet new requirements. The following example code illustrates an implementable approach to this.

```
SELECT `c`.* FROM `sct2_concept` AS `c`
  WHERE `c`.`effectiveTime` = (SELECT MAX(`c2`.`effectiveTime`)
  FROM `sct2_concept` `c2`
```

```
   WHERE `c2`.`id` = `c`.`id`
      AND `c2`.`effectiveTime` <= `snapshotTime`())
```

Figure 94: General form of SQL to create a snapshot view

In this sample code `snapshotTime()` is a function that returns the time to be applied to this snapshot. For the most recent *snapshot view* this can be omitted as shown below:

```
SELECT `c`.* FROM `sct2_concept` AS `c`
   WHERE `c`.`effectiveTime` = (SELECT MAX(`c2`.`effectiveTime`)
   FROM `sct2_concept` `c2`
   WHERE `c2`.`id` = `c`.`id`)
```

Figure 95: SQL to create the latest snapshot view

Similar views can be created for each of the *Component* tables by simply replacing the table name in both the outer and nested queries.

Note: The SQL queries in this and the following section assume applicability of a common versioning view for all *modules*. In some case, where a *module* that is being used depends on an earlier version of another *module*, more complex queries and optimizations may be needed. For further information about how dependencies between *module* are represented see details of the *Module dependency reference set*.

7.3.2 Optimizing Dynamic Snapshots Views

Some databases may be able to generate dynamic *snapshot views* sufficiently rapidly to enable real time use. However, in other cases, even if the nested queries used in the general *snapshot views* work quickly on their own, more complex queries involving joins between different *component* tables may lead to performance degradation. There are several approaches that can be taken to optimizing performance and two of these are in the following subsections.

7.3.2.1 Optimizing using a Snapshot View Flag

The first optimization approach is provide a simple way to optimize the current snapshot and can be extended to cover a limited number of additional *snapshot views*. A column is added to each *component* table to hold a *boolean* value that indicates whether or not a particular row is part of the current snapshot. In the following *Description* and example this added column is called `inSnapshot` and is referred to as a "*snapshot view* flag".

After importing or updating *SNOMED CT* content the *snapshot view* flag is updated using the results of a *snapshot view* query such as one illustrated in *Figure 96*. The example uses an intermediate temporary table. In some relational database environments nested queries could be used to reduce the number of steps in the script. However, the longer form is used here as some environments do not work (or are unpredictable) when updating a table that is also referenced by a nested select query.

```
/* Clear the inSnapshot flag */
UPDATE `sct2_concept` SET `inSnapshot`=False;

/* Create temporary table to hold latest id+effectiveTime */
DROP TEMPORARY TABLE IF EXISTS `tmp_ids`;

CREATE TEMPORARY TABLE `tmp_ids` (`id` BIGINT,`effectiveTime` DATETIME, PRIMARY KEY (`id`));

/* replace the line above with the line below for Refsets as the Id is a UUID rather than SCTID */
/* CREATE TEMPORARY TABLE `tmp_ids` (`id` BINARY(16),`effectiveTime` DATETIME, PRIMARY KEY
(`id`)); */

/* Populate the temporary table with id+effectiveTime for the latest view*/
INSERT INTO `tmp_ids` SELECT `id`,`effectiveTime` FROM `sct2_concept` AS `c`
   WHERE `c`.`effectiveTime` = (SELECT MAX(`c2`.`effectiveTime`)
```

```
   FROM `sct2_concept` `c2`
   WHERE `c2`.`id` = `c`.`id`);

/* Use the temporary table to update the inSnapshot flag for relevant rows */
UPDATE `sct2_concept` AS `c`,`tmp_ids` AS `t`
SET
`inSnapshot` = True
   WHERE `c`.`id`=`t`.`id` AND `c`.`effectiveTime` = `t`.`effectiveTime`;

/* Clean up by removing the temporary table */
DROP TEMPORARY TABLE `tmp_ids`;
```

Figure 96: Setting the latest snapshot view flag

The following query illustrates the simple query that can be used to return the current *snapshot view* using the *snapshot view* flag.

```
SELECT `c`.* FROM `sct2_concept` AS `c` WHERE `c`.`inSnapshot` = True;
```

Figure 97: Using a snapshot view flag to select components in a snapshot view

The same approach can be applied to each of the *components* by replacing `sct2_concept` with the relevant table name.

Additional *snapshot view* flags can be added, set and used in a similar way for a few other snapshot times that need to be optimized.

```
/* Clear the inSnapshotPrev flag */
UPDATE `sct2_concept` SET `inSnapshotPrev`=False;

/* Create temporary table to hold latest id+effectiveTime */
DROP TEMPORARY TABLE IF EXISTS `tmp_ids`;

CREATE TEMPORARY TABLE `tmp_ids` (`id` BIGINT,`effectiveTime` DATETIME, PRIMARY KEY (`id`));

/* replace the line above with the line below for Refsets as the Id is a UUID rather than SCTID */
/* CREATE TEMPORARY TABLE `tmp_ids` (`id` BINARY(16),`effectiveTime` DATETIME, PRIMARY KEY
(`id`));

/* Populate the temporary table with id+effectiveTime for the specified view date time */
INSERT INTO `tmp_ids` SELECT `id`,`effectiveTime` FROM `sct2_concept` AS `c`
   WHERE `c`.`effectiveTime` = (SELECT MAX(`c2`.`effectiveTime`)
   FROM `sct2_concept` `c2`
   WHERE `c2`.`id` = `c`.`id` AND `c2`.`effectiveTime` <= CAST('2010-01-31', DATETIME));

/* Use the temporary table to update the inSnapshotPrev flag for relevant rows */
UPDATE `sct2_concept` AS `c`,`tmp_ids` AS `t`
SET
`inSnapshotPrev` = True
   WHERE `c`.`id`=`t`.`id` AND `c`.`effectiveTime` = `t`.`effectiveTime`;

/* Clean up by removing the temporary table */
DROP TEMPORARY TABLE `tmp_ids`;
```

Figure 98: Setting the snapshot view flag for a specified date

This approach provides a simple approach to optimization of a limited number of views. However, it is constrained by the need to allocate a column for each time for which an optimized *snapshot view* is required.

7.3.2.2 Optimizing using a Superseded Time

This approach to optimization of *dynamic snapshot views* uses a single additional column in each *component* table to denote the time at which a row was superseded by a new version of the same *component*. This is more flexible but may not deliver the same performance improvement as the *snapshot view* flag approach.

After importing or updating *SNOMED CT* content the superseded time values are checked and updated where relevant using a query such as one illustrated in *Figure 99*. In this example, a fixed distant future date (31-12-9999) is used for *Components* which have not been superseded. The alternative would be a null date but the fixed distant date avoids the need to look for null as an exception at runtime. It also allows additional optimization of the current view - particularly if the supersededTime is indexed.

```
/* Create temporary table to hold latest id+effectiveTime */
DROP TEMPORARY TABLE IF EXISTS `tmp_supersede`;

CREATE TEMPORARY TABLE `tmp_supersede` (`id` BIGINT,`effectiveTime` DATETIME,`supersededTime`
 DATETIME, PRIMARY KEY (`id`,`effectiveTime`));

/* replace the line above with the line below for Refsets as the Id is a UUID rather than SCTID */
/* CREATE TEMPORARY TABLE `tmp_supersede` (`id` BINARY(16),`effectiveTime`
DATETIME,`supersededTime` DATETIME, PRIMARY KEY (`id,`effectiveTime``)); */

/* Populate the temporary table with id+effectiveTime+supersededTime */
INSERT INTO `tmp_supersede`
   SELECT `c`.`id`, `c`.`effectiveTime`, (SELECT IFNULL(MIN(`c2`.`effectiveTime`),CAST('9999-12-31' AS
DATETIME))
   FROM `sct2_concept` AS `c2` WHERE `c`.`id`=`c2`.`id` AND `c`.`effectiveTime`<`c2`.`effectiveTime`) AS
 supercededTime
   FROM `sct2_concept` AS `c`;

/* Use the temporary table to update the supersededTime flag for relevant rows */
UPDATE `sct2_concept` AS `c` JOIN `tmp_supersede` AS `t`
   ON `t`.`id`=`c`.`id` AND `t`.`effectiveTime`=`c`.`effectiveTime`
   SET `c`.`supersededTime`=`t`.`supersededTime`;

/* Clean up by removing the temporary table */
DROP TEMPORARY TABLE `tmp_supersede`;
```

Figure 99: Populating or updating the superseded time after importing content

Figure 100 illustrates the general query for returning the *snapshot view* for a specified time. To be included in the view the *effectiveTime* must be the same as or before the snapshot time and the supersededTime must be after the snapshot time.

```
SELECT `c`.* FROM `sct2_concept` AS `c` WHERE `c`.`effectiveTime`<=`snapshotTime`() AND
`c`.`supersededTime`>`snapshotTime`();
```

Figure 100: Using the superseded time to select components in the a specified snapshot view

Figure 101 illustrates the simpler query that can be used to return the current *snapshot view* using the superseded time value. If the supersededTime is included in the relevant composite indexes this may further improve the optimization for this commonly required view.

```
SELECT `c`.* FROM `sct2_concept` AS `c` WHERE `c`.`supersededTime` = CAST('9999-12-31' AS
DATETIME);
```

Figure 101: Using the superseded time to select components in the current snapshot view

The same approach can be applied to each of the *Components* by replacing `sct2_concept` with the relevant table name.

7.4 Working with metadata

SNOMED CT RF2 files represent some key information about core release *components* by reference to other *SNOMED CT components*. Two types of metadata (*Concept Enumerations* and *Reference Sets*) are described in the following sections:

- *Concept Enumerations* provide sets of values for enumerated fields in *SNOMED CT components*.
- *Reference Sets* support a wide range of functions and the following section subdivide these functions according their relative importance:
  - *Essential Reference Sets*;
  - *Optional Reference Sets*;
  - *Reference Sets supporting advanced functionality*.
  - <xml></xml>

7.4.1 Concept enumerations

SNOMED CT core components have some fields that have values represented by *concepts* in specific parts of the *SNOMED CT* hierarchy. These are referred to as *concept* enumerations.

The range of permitted values for each of the *concept* enumerations is the set of *subtypes* of a specified *concept* which is itself a *subtype* of 900000000000442005 | Core metadata concept (core metadata concept) |. The current set of *concept* enumeration types is shown in *Table 202*. The values of each of these and the ways they should be used in implemented systems are described in the following subsections.

Table 202: Core metadata concept (core metadata concept) (900000000000442005)

Id	Term	Comment
900000000000443000	| Module (core metadata concept) |	Each *subtype* of this *concept* represents a development module. These *concepts* provide values to the *moduleId* field that is present in all *SNOMED CT component* file. The value indicates the module within which a *component* was created and is being maintained.
900000000000444006	| Definition status (core metadata concept) |	Each *subtype* of this *concept* represents a value that can be applied to the *concept.definitionStatusId* field. This is used to indicate whether the current set of defining *Relationships* applied to a *concept* are sufficient to fully-define it relative to its supertypes.

Id	Term	Comment
900000000000446008	| Description type (core metadata concept) |	Each *subtype* of this *concept* represents a value that can be applied to the *Description*.*typeId* field. This is used to indicate whether the *Description* represents a *Fully Specified Name*, a synonymous term, a definition or some other symbolic or textual representation of the associated *concept*.
900000000000447004	| Case significance (core metadata concept) |	Each *subtype* of this *concept* represents a value that can be applied to the *Description. caseSignificanceId* field. This is used to indicate whether the text of the term can be modified to by switching characters from upper to lower case (or vice-versa).
900000000000449001	| Characteristic type (core metadata concept) |	Each *subtype* of this *concept* represents a value that can be applied to the *Relationship. characteristicTypeId* field. This is used to indicate whether a *Relationship* forms part of the definition of the source *concept*.
900000000000450001	| Modifier (core metadata concept) |	Each *subtype* of this *concept* represents a value that can be applied to the *Relationship.modifierId* field. This is used to indicate the type of *Description Logic* (DL) restriction (some, all, etc.) that applies to the *Relationship*.
900000000000453004	| Identifier scheme (core metadata concept) |	Each *subtype* of this *concept* represents a value that can be applied to the *Identifier. identifierSchemeId* field. This is used to indicate the scheme to which the *Identifier* value belongs.

Note: Many of the *concept* enumerations include values that significantly impact the meaning or use of a *component*. Therefore, implementers may find it necessary to partially hard-code the way their systems process particular values. In these cases, the *concept* referenced by the value is only of value when there is a requirement to display a human readable rending of the value. The main exceptions to this are 900000000000443000 | Module (core metadata concept) | and 900000000000453004 | Identifier scheme (core metadata concept) | both of which represent extensible sets of values as new modules or alternative *Identifier* schemes may be added in local *Extensions*.

7.4.1.1 Concept enumerations for moduleId

This *concept* enumeration applies to the *moduleId* field which is present in all released *SNOMED CT components* (*RF2*). The value applied to a particular *component* indicates the development module within which that *component* was created and is being maintained.

Each of the values in *Table 203* represents a development module. The range of permitted list of values is extensible by addition of branches to the hierarchy shown in *Figure 102* modules managed by other organizations (i.e. in an extensions namespace) and to add specific module *Identifiers* within each branch.

Table 203: International Health Terminology Standards Development Organisation maintained module (core metadata concept) (900000000000445007)

Id	Term	Comment
900000000000012004	| SNOMED CT model component module (core metadata concept) |	
900000000000207008	| SNOMED CT core module (core metadata concept) |	

- 900000000000443000 | Module (core metadata concept) |
  - 900000000000445007 | International Health Terminology Standards Development Organisation maintained module (core metadata concept) |
    - 900000000000012004 | SNOMED CT model component module (core metadata concept) |
    - 900000000000207008 | SNOMED CT core module (core metadata concept) |

Figure 102: Hierarchy of SNOMED CT moduleId values

7.4.1.2 Concept enumerations for definintionStatusId

This *concept* enumeration represents a value that can be applied to the *concept*. *definitionStatusId* field. This is used to indicate whether the current set of defining *Relationships* applied to a *concept* are sufficient to fully-define it relative to its supertypes.

Table 204 shows the current set of values for this *concept* enumeration.

Table 204: Definition status (core metadata concept) (900000000000444006)

Id	Term	Comment
900000000000073002	| Sufficiently defined concept definition status (core metadata concept) |	The set of defining *Relationships* applied to the *concept* are asserted to fully define the *concept*. Any *concept* or *expression* for which all these defining *Relationships* are true is either equivalent to or subsumed by this *concept*. Any *concept* or *expression* for which any of these defining *Relationships* is not true is neither equivalent to nor subsumed by this *concept*.

Id	Term	Comment
900000000000074008	| Necessary but not sufficient concept definition status (core metadata concept) |	The set of defining *Relationships* applied to the *concept* are asserted to be incompletely define the *concept*. The *concept* is currently considered to be *primitive*. A *concept* or *expression* for which all these defining *Relationships* are true **may** be equivalent to or subsumed by this *concept*. However, it is not possible to compute this from the definition - because the missing element in the definition may or may not apply to the other *concept* or *expression*. Any *concept* or *expression* for which any of these defining *Relationships* is not true is neither equivalent to nor subsumed by this *concept*.

7.4.1.3 Concept enumerations for Description typeId

This *concept* enumeration represents a value that can be applied to the *Description*. *typeId* field. This is used to indicate whether the *Description* represents a *Fully Specified Name*, a synonymous term, a definition or some other symbolic or textual representation of the associated *concept*.

Table 205 shows the current set of values for this *concept* enumeration.

Table 205: Description type (core metadata concept) (900000000000446008)

Id	Term	Comment
900000000000003001	| Fully specified name (core metadata concept) |	The *Description*.term represents the *Fully Specified Name* of the associated *concept* in the language indicated by the *Description*.languageCode.
900000000000013009	| Synonym (core metadata concept) |	The *Description*.term represents a term that is used to represent the associated *concept* in the language indicated by the *Description*.languageCode. **Note:** The *preferred term* used in a given language or *dialect* is marked as a *synonym*. Preference and acceptability of a particular synonymous term is indicated by a Language *refset*.
900000000000550004	| Definition (core metadata concept) |	The *Description*.term represents a textual definition of the associated *concept* in the language indicated by *Description*.languageCode.

7.4.1.4 Concept enumerations for caseSignificanceId

This *concept* enumeration represents a value that can be applied to the *Description*. *caseSignificanceId* field. This is used to indicate whether the text of the term can be modified to by switching characters from upper to lower case (or vice-versa).

Table 206 shows the current set of values for this *concept* enumeration.

Table 206: Case significance (core metadata concept) (900000000000447004)

Id	Term	Comment
900000000000017005	| Entire term case sensitive (core metadata concept) |	The text of the *Description*.term must be presented in the case in which it is specified.
900000000000020002	| Only initial character case insensitive (core metadata concept) |	The initial character of the *Description*.term is case insensitive and can be changed from upper to lower case (or vice-versa) if appropriate to the context in which it is used. This applies only to the first character of the term as a whole, not to the initial character of other words in the term.
900000000000448009	| Entire term case insensitive (core metadata concept) |	The entire *Description*.term is case insensitive and can be can be changed from upper to lower case (or vice-versa) if appropriate to the context in which it is used.

7.4.1.5 Concept enumerations for characteristicTypeId

This *concept* enumeration represents a value that can be applied to the *Relationship*. *characteristicTypeId* field. This is used to indicate whether a *Relationship* forms part of the definition of the source *concept*.

Table 207 shows the current set of values for this *concept* enumeration. Note that two the values 900000000000010007 | Stated relationship (core metadata concept) | and 900000000000011006 | Inferred relationship (core metadata concept) | are *subtypes* of the more general value 900000000000006009 | Defining relationship (core metadata concept) |.

Table 207: Characteristic type (core metadata concept) (900000000000449001)

Id	Term	Comment
900000000000006009	| Defining relationship (core metadata concept) |	The *Relationship* is part of the *description logic* definition of the source *concept*.
- 900000000000010007	Stated Relationship (core metadata concept)	Indicates that this defining *Relationship* was stated by a terminology author.

Id	Term	Comment
- 900000000000011006	Inferred Relationship (core metadata concept)	Indicates that this defining *Relationship* was inferred by a *description logic classifier* from the set of stated *Relationships*.
900000000000225001	| Qualifying relationship (core metadata concept) |	The *Relationship* is not part of the definition of the *concept* but indicates a possible qualification that may be applied to refine a *postcoordinated expression* that refers to the source *concept*.
900000000000227009	| Additional relationship (core metadata concept) |	The *Relationship* is not part of the definition of the *concept* but is used to convey some additional information about the *concept*. This additional information may only be applicable to a particular jurisdiction or use case.

7.4.1.6 Concept enumerations for modifierId

This *concept* enumeration represents a value that can be applied to the *Relationship.modifierId* field. This is used to indicate the type of *Description Logic* (DL) restriction (some, all, etc.) that applies to the *Relationship*.

Table 208 shows the current set of values for this *concept* enumeration.

Table 208: Modifier (core metadata concept) (900000000000450001)

Id	Term	Comment
900000000000451002	| Existential restriction modifier (core metadata concept) |	Indicates that *description logic* restriction represented by this defining *Relationship*applies to **some** aspect of the *concept*.
900000000000452009	| Universal restriction modifier (core metadata concept) |	Indicates that *description logic* restriction represented by this defining *Relationship*applies to **all aspects** of the *concept*.

7.4.1.7 Concept enumerations for identifierSchemeId

This *concept* enumeration represents a value that can be applied to the *Identifier.identifierSchemeId* field. This is used to indicate the scheme to which the *Identifier* value belongs.

Table 209 shows the current set of values for this *concept* enumeration. This set of values is extensible to allow additional *Identifiers* to be used to represent *SNOMED CT components* where this is necessary.

Table 209: Identifier scheme (core metadata concept) (900000000000453004)

Id	Term	Comment
900000000000002006	| SNOMED CT universally unique identifier (core metadata concept) |	The identification scheme in which the *Identifiers* are *UUID*'s allocated to *SNOMED CT components*.
900000000000294009	| SNOMED CT integer identifier (core metadata concept) |	The scheme comprising all *SNOMED Clinical Terms Identifiers* (*SCTID*).

7.4.1.8 Other Concept enumerations

Reference sets can also include *concept* enumeration values and the values for these are *subtypes* of 900000000000491004 | Attribute value (foundation metadata concept) |. The values applicable to each *Attribute* in each type of *Reference set* are specified by the 900000000000456007 | Reference set descriptor reference set (foundation metadata concept) |.

Note: In the current pre-release *RF2* data some sets of *concept* enumerations are *subtypes* of 900000000000457003 | Reference set attribute (foundation metadata concept) |. However, in future it is anticipated that they will all be *subtypes* of 900000000000491004 | Attribute value (foundation metadata concept) |.

7.4.2 Essential Reference Sets

The *Reference Set* mechanism provides flexibility and extensibility to the core terminology. The *Reference Sets* described in this section are essential and need to be supported by all *SNOMED CT enabled terminology servers*.

Other *Reference Sets* are used to deliver specific added value functionality and/or for local configuration. While implementers are advised to consider providing full *Reference Set* support the specific requirements for these depend on the intended uses of the systems and these are described elsewhere in the guide.

7.4.2.1 Language Reference Sets

At least one language *Reference Set* needs to be imported. This is essential to enable the *preferred term* to be identified for each *concept*.

The language *Refsets* supported in the *International Release* are shown in *Table 210*.

Table 210: English [International Organization for Standardization 639-1 code en] language reference set (foundation metadata concept) (900000000000507009)

Id	Term	Comment
900000000000508004	| Great Britain English language reference set (foundation metadata concept) |	
900000000000509007	| United States of America English language reference set (foundation metadata concept) |	

The Language *Reference Set* hierarchy is extensible and other languages and *dialects* will be added to the hierarchy shown in *Figure 103* to either as part of the *International Release* or an *Extension*.

- 900000000000506000 | Language type reference set (foundation metadata concept) |
  - 900000000000507009 | English [International Organization for Standardization 639-1 code en] language reference set (foundation metadata concept) |
    - 900000000000508004 | Great Britain English language reference set (foundation metadata concept) |
    - 900000000000509007 | United States of America English language reference set (foundation metadata concept) |

Figure 103: The Language Reference Set hierarchy

Each language *Reference set* refers to each of the *Descriptions* that is used in that language or *dialect* and assigns a value for the acceptability of the term associated with that *Description* when applied to the *Concept* associated with that *Description*. The values for acceptability are *concept* enumerations show in *Table 211*.

Table 211: Acceptability (foundation metadata concept) (900000000000511003)

Id	Term	Comment
900000000000548007	| Preferred (foundation metadata concept) |	The term associated with this *description* is the preferred *description*, of the specified *Description*.type, for the associated *concept*, in the language or *dialect* represented by this *Reference set*. • If the *Description*.type is *synonym*, this *description* is the *preferred term*. • If the *Description*.type is *fully specified name* this *description* is the preferred *fully specified name*. For each *concept* there should be exactly one preferred *description* of each *Description*.type in each language *Reference set*.
900000000000549004	| Acceptable (foundation metadata concept) |	The term associated with this *description* is acceptable for use in language or *dialect* represented by this *Reference set*. For each *concept* there may be any number of acceptable *descriptions* of each *Description*.type in each language *Reference set*.

7.4.2.2 Component Inactivation Reference Sets

The *Component* Inactivation *Reference Sets* are required to determine the reason why a *concept*, *Description* or *Relationship* is inactive. The *boolean* active field in each *component* indicates whether it is active but does not explain why a previously active *component* has been inactivated. The reason for inactivation may affect the way in which *components* that have been made inactive are dealt with when they have been used to create records, protocols or queries prior to inactivation.

The three *Component* Inactivation *Reference Sets* are shown in *Table 212*.

Table 212: Component Inactivation Reference Sets

Id	***Fully Specified Name***	**Note**
900000000000489007	I Concept inactivation indicator attribute value reference set (foundation metadata concept) I	Indicates the reason that a *concept* has been made inactive.
900000000000490003	I Description inactivation indicator attribute value reference set (foundation metadata concept) I	Indicates the reason that a *description* has been made inactive.
900000000000547002	I Relationship inactivation indicator attribute value reference set (foundation metadata concept) I	(Not currently provided - for future use)

The reason for inactivation is specified by a *concept* enumeration. The permitted values for this enumeration for a *Concept* are shown in *Table 213* and the permitted values for a *Description* are shown in *Table 214*.

Table 213: Concept inactivation value (foundation metadata concept) (900000000000481005)

Id	***Fully Specified Name***	**Comment**
900000000000482003	I Duplicate component (foundation metadata concept) I	The *Concept* has been made inactive because it has the same meaning as another *Concept* .
900000000000483008	I Outdated component (foundation metadata concept) I	The *Concept* has been made inactive because it is an outdated *concept* that is no longer used.
900000000000484002	I Ambiguous component (foundation metadata concept) I	The *Concept* has been made inactive because it is inherently ambiguous either because of an incomplete *fully specified name* or because it has several associated terms that are not regarded as synonymous or partial synonymous.
900000000000485001	I Erroneous component (foundation metadata concept) I	The *Concept* has been made inactive because it contains an error.
900000000000486000	I Limited component (foundation metadata concept) I	The *Concept* is of limited value as it contains classification categories such as 'Not Elsewhere Classified' which do not have a stable meaning within *SNOMED CT*. Until 2010 *concepts* with this status were regarded as active but since then they have been marked as inactive.
900000000000487009	I Component moved elsewhere (foundation metadata concept) I	The *Concept* has been made inactive because it has been moved to another namespace.

Id	*Fully Specified Name*	Comment
900000000000492006	| Pending move (foundation metadata concept) |	The *Concept* is **still active** but it is in the process of bing moved to another namespace and when the move is complete it will be marked as inactive.

Table 214: Description inactivation value (foundation metadata concept) (900000000000493001)

Id	*Fully Specified Name*	Comment
900000000000482003	| Duplicate component (foundation metadata concept) |	The *Description* has been made inactive because it duplicates another *Description* .
900000000000483008	| Outdated component (foundation metadata concept) |	The *Description* has been made inactive because it is an outdated name or spelling that is no longer used.
900000000000485001	| Erroneous component (foundation metadata concept) |	The *Description* has been made inactive because it contains an error.
900000000000486000	| Limited component (foundation metadata concept) |	The *Description* refers to a *Concept* that has limited status. **Note:** This value should not be used in future releases as Limited status *Concepts* are now inactive. However, this value may appear on retrospective data in a *full release* .
900000000000487009	| Component moved elsewhere (foundation metadata concept) |	The *Description* has been made inactive because it has been moved to another namespace.
900000000000492006	| Pending move (foundation metadata concept) |	The *Description* is **still active** but it is in the process of bing moved to another namespace and when the move is complete it will be marked as Inactive.
900000000000494007	| Inappropriate component (foundation metadata concept) |	The *Description* has been made inactive because the associated term is does not describe the associated *Concept* .
900000000000495008	| Concept non-current (foundation metadata concept) |	The *Description* is **still active** but the *Concept* it refers to is now inactive.

The *component*.active field allows rapid determination of whether a *component* is intended for active use. However, where a full interpretation of the status of a *component* is required two factors must be taken into account. The absence of a row in the relevant inactivation *Refset* implies a default meaning which and this default meaning depends on whether the *component* is active or inactive:

- For an active *component* it means active and in current use as distinct from active *pending move*
- For an *inactive component* it means inactive with no reason given for inactivation.

This leads to the set of interpretations for each possible combination of values shown in *Table 215*.

Table 215: Concept Status evaluation table

Most recent *Concept* row for a *concept*.id	**Most recent *Refset* row for the RefsetMember.id in " *Concept* inactivation *Refset*" for the *concept*.id**		**ConceptStatus (with RF1 enumerated value)**
Exists/active	**Exists/active**	***valueId***	
None	None	-	*Not applicable (not yet released)*
Active	None or Inactive	-	Current (0)
Inactive	None or Inactive	-	Inactive with no reason given for inactivation (1)
Inactive	Active	900000000000482003	| duplicate | (2)
Inactive	Active	900000000000483008	| outdated | (3)
Inactive	Active	900000000000484002	| ambiguous | (4)
Inactive	Active	900000000000485001	| erroneous | (5)
Inactive	Active	900000000000486000	| limited | (6) (from 2010-01-31)
Active	Active	900000000000486000	| limited | (6) (before 2010-01-31)
Inactive	Active	900000000000487009	| moved elsewhere |
Active	Active	900000000000492006	| pending move |
Any combinations not shown above			*Future releases may add new values or rules. Otherwise, values and combinations not shown in this table have no agreed interpretation and must be regarded as data errors.*

7.4.2.3 Historical Association Reference Sets

Historical Association *Reference Sets* provide links between *inactive concepts* and their active replacements or equivalents. There is one Historical Association *Reference Set* for each type of historical association as shown in *Table 216*.

Table 216: Historical association reference set (foundation metadata concept) (900000000000522004)

Id	Term	Comment
900000000000523009	| POSSIBLY EQUIVALENT TO association reference set (foundation metadata concept) |	Applies to a *concept* that is ambiguous. The targetComponent is an active *concept* that represents one of the possible meanings of the inactive *concept*. Multiple rows are used to refer to each of the possible meanings of the ambiguous *concept*. Previously referred to as "MAY BE A".
900000000000524003	| MOVED TO association reference set (foundation metadata concept) |	Applies to a *component* that has been moved to (or are pending a move to) another namespace. The targetComponent identifies the target namespace (not the new *component*).
900000000000525002	| MOVED FROM association reference set (foundation metadata concept) |	Applies to a *component* that has been moved to this namespace from another namespace. The targetComponent identifies the original *component Identifier* in its previous namespace.
900000000000526001	| REPLACED BY association reference set (foundation metadata concept) |	Applies to an erroneous, obsolete and other *inactive component* for which there is a single active replacement. The targetComponent identifies the active *component* that replaces this *component*.
900000000000527005	| SAME AS association reference set (foundation metadata concept) |	Applies to a *component* that is a duplicate. The targetComponent identifies the active *component* that this *component* duplicates.
900000000000528000	| WAS A association reference set (foundation metadata concept) |	Links an inactive classification *concept* such as "not otherwise specified" or "otherwise specified" with the active *concept* that was formerly its most proximal supertype.
900000000000529008	| SIMILAR TO association reference set (foundation metadata concept) |	(not used currently)
900000000000530003	| ALTERNATIVE association reference set (foundation metadata concept) |	Links an inactive classification *concept* derived from ICD-9 Chapter XVI 'Symptoms signs and ill-defined conditions' with the most similar active *concept*.

Id	Term	Comment
900000000000531004	I REFERS TO concept association reference set (foundation metadata concept) I	Applies to an inactive *description* which is inappropriate to the *concept* it is directly linked to but instead should refer to the *concept* referenced by the targetComponent.

7.4.2.4 Module Dependency Reference Set

The Module Dependency *Reference Set* provides information about dependencies between different version of particular development modules. This *Reference Set* (identified as 900000000000534007 I Module dependency reference set (foundation metadata concept) I) should be checked when importing data to ensure that all dependencies are satisfied.

The rows in this *Reference Set* that originate in a given module (identified by *moduleId*) indicate a dependency on the module identified by the *referencedComponentId*. The two *string* values each contain dates that indicate the version of source module and the required version of the module on which it depends.

7.4.3 Optional Reference Sets

The *Reference Sets* described in the following sections are required for specific purposes. If an implementation does not need to address a particular requirement (e.g. mapping from a legacy coding scheme) or supports a more up to date approach (e.g. the *Machine Readable Concept Model* rather than the use of refinability flags) then that *Reference Set* need not be imported or may be imported and not used.

7.4.3.1 Legacy Code Map Reference Sets

Legacy Code Map *Reference Sets* are simple maps to *SNOMED CT* from legacy code systems, including *SNOMED* codes (i.e. codes used in *SNOMED* 3) and *NHS Clinical Terms Version 3 Identifiers* (including all versions of the *Read Codes*). There is one *Reference Set* for legacy *SNOMED* codes and one for *Clinical Terms Version 3* as shown in *Table 217*.

In both cases, the referenceComponentId refers to a *SNOMED CT concept* and the mapTarget *string* value is the code in the other coding scheme.

Table 217: Simple map type reference set (foundation metadata concept) (900000000000496009)

Id	Term	Comment
900000000000497000	I CTV3 simple map reference set (foundation metadata concept) I	The map between *Clinical Terms Version 3* and all version of the *Read Codes* and *SNOMED CT*.
900000000000498005	I SNOMED RT identifier simple map (foundation metadata concept) I	The map between legacy *SNOMED* codes and *SNOMED CT*.

7.4.4 Reference Sets supporting advanced functionality

Some of the *Reference Sets* included as part of the *SNOMED CT International Release* support advanced uses and may not need to be implemented. In particular *Reference Sets* that provide information about other *Reference Set* can be valuable but are not essential provided the implementation fully supports all the *Reference Sets* required by its users.

7.4.4.1 Description Format Reference Set

The *Description* Format *Reference Set* provides information about the format of each of the *Description* types. This *Reference Set* is identified as 900000000000538005 | Description format reference set (foundation metadata concept) |.

The *referencedComponentId* of each member of the *reference set* refers to a *Description* type, represented by a *subtype* of the *concept* 900000000000446008 | Description type (core metadata concept) |. The *descriptionFormat* refers to one of the *Concept enumeration* values shown in *Table 218*. The *descriptionLength* indicates the longest permitted string for this *Description* type.

Table 218: Description format (foundation metadata concept) (900000000000539002)

Id	Term	Comment
900000000000540000	| Plain text (foundation metadata concept) |	*Descriptions* of this types linked to this format are in plain text. This applies *fully specified names* and *synonyms*.
900000000000541001	| Limited HyperText Markup Language (foundation metadata concept) |	*Descriptions* of this types linked to this format use a limited version of HTML markup.
900000000000542008	| Extensible HyperText Markup Language (foundation metadata concept) |	*Descriptions* of this types linked to this format may use the full scope of XHTML markup.
900000000000543003	| Darwin Information Typing Architecture (foundation metadata concept) |	*Descriptions* of this types linked to this format are represented as *DITA* topics using XML markup.

7.4.4.2 Reference Set Descriptor Reference Set

The *Reference Set* Descriptor *Reference Set*, which is identified as 900000000000456007 | Reference set descriptor reference set (foundation metadata concept) |, provides information about the structure of each type of *Reference Set*.

7.4.5 Using other Reference Sets

7.4.5.1 Importing Reference Sets

One or more *Reference Sets* may be held in a single *Reference Set release file*. However, if there are more than one *Reference Sets* in a single file, they will all have the same structure (i.e. - the same number of additional fields of the same top level types of *component*, *Integer* or *String*).

Each record in the *Reference Set* file represents a member of the *reference set*. The refsetId column identifies the *Reference Set* that the member record belongs to.

The refsetId is an *SCTID* that can be used to look up the *concept* in the | *Reference Set* | metadata that describes the *reference set*. Up to three *Descriptions* (with three different *typeIds*) may be associated with the *Reference Set concept*:

- A *Description* with a *typeId* of IFSNI, used to formally describe the *Reference Set*. This *Description* will always exist.

- A *Description* with a *typeId* of | *Synonym* |, used to name the *Reference Set*. This *Description* will always exist, and can be used to display the name of the *Reference Set* within a system.
- A *Description* with a *typeId* of |Purpose|, used to describe the purpose of the *Reference Set*. This *Description* may or may not be present.

The refsetId can also be used to look up the *Reference Set* Descriptor, in the | *Reference set* descriptor| *Reference Set*. This can be done by identifying the member records in the | *Reference set* descriptor| *reference set* with a referencedComponentIdthat matches the refsetId of the *Reference Set*.

There will be one Descriptor record describing the *referencedComponentId* field in the *Reference Set* and one additional record for each optional field within the *Reference Set*. The Descriptor record with an *attributeOrder* field value of '0' describes the referencedComponentIdfield; a Descriptor record with an *attributeOrder* field value of '1' would describe the first optional field; etc.

For each *Reference set* field being described (i.e. - the *referencedComponentId* and each optional field), two fields in the Descriptor record provide additional information:

- The *attributeType* field is a reference to a *concept* under the | *Attribute* type| metadata *hierarchy* that provides typing information for the field. At the top level, this could be | *component* type|, | *Integer* | or | *String* |, and would then match the typing information available within the *Reference Set* file name (see the *SNOMED CT - File Naming Conventions*). However, the type of a field can also be specified at a finer level of granularity using the *attributeType* field. For instance, instead of the *attributeType* being specified simply as an | *Integer* |, it may instead be specified as an |Unsigned *integer* | or a |Signed *integer* |. For a full list of types, see the | *Attribute* type| metadata *hierarchy*.
- The *attributeDescription* field is a reference to a *concept* under the | *Reference set* attribute | metadata *hierarchy* that also provides additional information about each *Reference Set* field. Up to three *Descriptions* (with three different *typeIds*) may be associated with each of these *concepts*:
    - A *Description* with a *typeId* of |FSN|, used to formally describe the *Reference Set* field. This *Description* will always exist.
    - A *Description* with a *typeId* of | *Synonym* |, used to name the *Reference Set* field. This *Description* will always exist, and can be used to display a column header for each *Reference Set* field used within a system.
    - A *Description* with a *typeId* of |Purpose|, used to describe the purpose of the *Reference Set* field. This *Description* may or may not be present.

Additionally, if the *attributeType* is | *Concept* type component |, then the *children* of the *concept* referred to by the *attributeDescription* provide a list of allowed *concept* enumeration values for the *Reference Set* field. Each of these *concepts* will have two *Descriptions* with *typeIds* of |FSN| and of | *Synonym* |, and the latter set of *Descriptions* can be used to validate field entry for *concept* enumeration type *Reference Set* fields or to create pick-lists to allow users to select one or more values. Where the *attributeDescription concept* does not have any *children*, then no limitation is placed on the *concepts* allowed in the *Reference Set* field.

7.4.5.2 Using Reference Sets without Descriptors

All *Reference Sets* that are released from *IHTSDO* or from a *National Release* Center will have an associated Descriptor for the *Reference Set*. However, Descriptors are optional for other organizations that create *Reference Sets*. Where you are using a *Reference Set* for which a Descriptor has not been created, and you need additional information about the *Reference Set*, the Descriptor of the closest *ancestor* of the *concept* describing the *Reference Set* that does have a Descriptor may be used. This situation should be rare, as an organization that releases *Reference Sets* should only release them without Descriptors if it is sure that its consumers do not require the information held within the Descriptors.

7.4.5.3 Using Reference Sets to hold simple value sets

Where it is known that a single simple *Reference Set* is held in a file, a simple *value set* may be retrieved from the *Reference Set* by taking the *referencedComponentIds* of each record with an *active* field set to '1'. Each value in the *value set* is then an *SCTID* of a *SNOMED CT component*.

Where a *release file* contains multiple simple *Reference Sets*, then a number of *value sets* may be retrieved from the file by taking the *referencedComponentIds* of each record with an *active* field set to '1', and grouping them into *value sets* by using the refsetId field. Each value in the *value set* is an *SCTID* of a *SNOMED CT component*. In order to retrieve the name of each *value set*, its refsetId can be used to identify a | *Reference set* | metadata *concept* that will have a *Description* with a *typeId* of | *Synonym* | that provides a name for the *value set*.

7.5 Foundation Terminology services

This section summarizes a set of services that all *terminology servers* require. Some of these services are described in more detail in subsequent sections. The more advanced services specified in other sections depend on one or more of these foundation services.

7.5.1 Access to release information

Terminology servers should enable client applications and users to access the current *SNOMED CT release*

7.5.2 Access to components

Most *Terminology services* depend on the ability to efficiently access information about the set of components in a selected *snapshot view*. The following sections outline the types of information that need to be accessible and provide illustration of some of the common patterns of data access that are required. The illustrations are expressed as SQL queries based on the *example relational representation* and *dynamic snapshot views approaches* discussed in earlier sections.

7.5.2.1 Access to concepts

A *terminology server* should enable client applications to rapidly find the current version of a *Concept* by its unique *Identifier* (*Concept*.id).

Once a *Concept* has been found, the client application should be able to read the values of the properties of that *Concept* which are either:

1. Provided directly as *concept file* fields:
   - active;
   - *definitionStatusId*.
2. Provided indirectly through associations to other *components*:
   - *Descriptions*.
   - *Relationships*.
3. Provided indirectly via relevant *Reference sets*:
   - For example *Information about Inactive Concepts*.

7.5.2.1.1 Information about Inactive Concepts

The *Concept*.active field is a *boolean* value which distinguishes between active and *inactive concepts*. To find out more information about the status of a *concept* it is necessary to look for a relevant row in the 900000000000480006 | Attribute value type reference set (foundation metadata concept) |.

The example query below illustrates this process.

```
/* sv_concept refers to a snapshot view of concept */
/* sv_refset_status refers to a snapshot view of the */
/* inactivation Refset with term lookup  see below */
```

```
SELECT `c`.`id` AS `ConceptId`,
(CASE WHEN (`r`.`RsActive` = 1) THEN
   `r`.`ValueTerm` else
   (CASE WHEN `c`.`active` THEN
      'Current' ELSE
      'Inactive no reason' END)
   END) AS `Status`
 FROM (`sv_concept` `c`
   LEFT JOIN `sv_refset_status` `r`
      ON (`r`.`ItemId` = `c`.`id`))
   WHERE `c`.`id`=[some-concept-id];

/* Query generating the sv_refset_status view */
/* sv_refset_c is snapshot view of the cRefset table */
/* sv_fsn is a snapshot view of fully specified names */
/* See section on access to Descriptions for details */

SELECT `r`.`active` AS `RsActive`,`r`.`referencedComponentId` AS `ItemId`
   ,`d2`.`conceptId` AS `ValueId`,`d2`.`term` AS `ValueTerm`
   FROM (`sv_refset_c` `r` join `sv_fsn` `d2`)
      WHERE ((`d2`.`conceptId` = `r`.`valueId`)
         AND (`r`.`refsetId` = 900000000000489007));
```

Figure 104: Determining concept status

If a *concept* is inactive then, it may be necessary to follow the historical associations to locate the *active concept*(s) that have replaced or disambiguated the *inactive concept*. *Figure 105* illustrates and finds the id of the active equivalent of a duplicate *concept*.

```
/* Find SAME AS reference for a duplicate concept  */
/* sv_refset_c is snapshot view of the cRefset table */

SELECT `targetComponent`
   FROM `sv_refset_c`
   WHERE `refsetId`=900000000000527005
      AND `referencedComponentId`=[some-concept-id];
```

Figure 105: Following historical associations

7.5.2.2 Access to Descriptions

A *terminology server* should enable client applications to rapidly find the current version of any *Description* or set of *Descriptions* by any of the following criteria:

- Its unique *Identifier* (*Description*.id);
- conceptId of the *concept* with which it is associated;
- A combination of conceptId, *DescriptionType*, Language or *dialect* and Acceptability (in that language or *dialect*).

Once a *Description* has been found the client application should be able to read the values of any of the properties of that *Description* which are either:

- Provided directly as *Description file* fields:
  - active;
  - term;
  - *caseSignificanceId*
  - languageCode;
  - *typeId* (the *Description* Type).
- Provided indirectly via relevant *Reference sets*:

- For an example see *Determining Description Type and Acceptability*.

7.5.2.2.1 Determining Description Type and Acceptability

The active field indicates whether the *Description* is in current active use. The *typeId* and languageCode indicate the *Description* type and the language of the associated term. This information is useful but it is not sufficient to determine the *preferred term*. In order to determine the acceptability of or preference for use of a particular *Description* it is necessary to apply a language *Reference set*. This is illustrated by *Figure 106*.

```
/* sv_description is a snapshot view of the description file */
/* sv_refset_c is snapshot view of the cRefset table */
/* configLang() is a function that returns the chosen language RefsetId */

SELECT `d`.*
FROM (`sv_description` `d` join `sv_refset_c` `rs`
   ON((`d`.`id` = `rs`.`referencedComponentId`)))
      WHERE ((`d`.`active` = 1) AND (`d`.`typeId` = 900000000000013009)
      AND (`d`.`conceptId`=[some-concept-id] AND (`rs`.`refsetId` = `configLangId`())
      AND (`rs`.`active` = 1) AND (`rs`.`valueId` = 900000000000548007)));
```

Figure 106: Identifying the preferred term

The *Fully Specified Name* for a particular language or *dialect* can also be determined in the same way as shown in *Figure 107*. The only difference between this and the *preferred term* example is the change in the *typeId* predicate.

Note: The *Fully Specified Name* may not be present in all supported languages therefore a fall-back to the US English may be necessary.

```
SELECT `d`.*
FROM (`sv_description` `d` join `sv_refset_c` `rs`
   ON((`d`.`id` = `rs`.`referencedComponentId`)))
      WHERE ((`d`.`active` = 1) AND (`d`.`typeId` = 900000000000003001)
      AND (`d`.`conceptId`=[some-concept-id] AND (`rs`.`refsetId` = `configLangId`())
      AND (`rs`.`active` = 1) AND (`rs`.`valueId` = 900000000000548007)));
```

Figure 107: Identifying the preferred fully specified name

Figure 108 illustrates an approach to returning all the acceptable or *preferred terms* together with an indication of which *Description* type and preference.

```
/* sv_description is a snapshot view of the description file */
/* sv_refset_c is snapshot view of the cRefset table */
/* configLang() is a function that return the chosen language RefsetId */

SELECT `d`.*, (CASE WHEN `rs`.`valueId`=900000000000548007 THEN
      'Preferred' ELSE
      'Acceptable' END) AS `Acceptability`
      (CASE WHEN `d`.`typeId`=900000000000013009 THEN
      'Synonym' ELSE
      'FSN' END) AS `DescriptionType`
FROM (`sv_description` `d` join `sv_refset_c` `rs`
   ON((`d`.`id` = `rs`.`referencedComponentId`)))
      WHERE ((`d`.`active` = 1) AND ((`d`.`typeId` = 900000000000013009) OR (`d`.`typeId` =
900000000000003001))
      AND (`d`.`conceptId`=[some-concept-id] AND (`rs`.`refsetId` = `configLangId`())
      AND (`rs`.`active` = 1) AND
      ((`rs`.`valueId` = 900000000000548007) OR (`rs`.`valueId` = 900000000000548007))));
```

Figure 108: Finding all the acceptable terms

7.5.2.3 Access to Relationships

A *terminology server* should enable a client application to rapidly find the current version of any *Relationship* or set of *Relationships* by any of the following criteria:

- Its unique *Identifier Relationship*.id;
- *sourceId*
- *sourceId*, *characteristicTypeId* and *typeId*
- *sourceId*, *characteristicTypeId*, *relationshipGroup* and *typeId*
- *destinationId*
- *destinationId*, *characteristicTypeId* and *typeId*

Once a *Relationship* has been found the client application should be able to read the values of any of the properties of that *Relationship*:

- Provided directly as *Relationship file* fields:
  - active;
  - *sourceId*
  - *characteristicTypeId*
  - *typeId*
  - *destinationId*
  - *relationshipGroup*
  - *modifierId*
- Provided indirectly in the *concepts* that it refers to:
  - For example *Using and traversing relationships.*

7.5.2.3.1 Using and traversing Relationships

The defining *Relationships* of a *concept* can be shown by following the relevant *concept identifier* and displaying the relevant terms as showing in *Figure 109*.

```
/* sv_relationship is a snapshot view of the relationship file */
/* sv_pref is a snapshot of descriptions filtered to preferred term */

SELECT  `r`.`typeId` AS `type_id`,`typ`.`term` AS `type_term`
 ,`r`.`destinationId` AS `dest_id`,`dest`.`term` AS `dest_term`
 ,`r`.`relationshipGroup` AS `relationshipGroup`
  FROM (((`sv_relationship` `r`
  JOIN `sv_pref` `src`
   ON ((`r`.`sourceId` = `src`.`conceptId`)))
  JOIN `sv_pref` `typ`
   ON ((`r`.`typeId` = `typ`.`conceptId`)))
  JOIN `sv_pref` `dest`
   ON ((`r`.`destinationId` = `dest`.`conceptId`)))
WHERE ((`r`.`active` = 1)
 AND (`r`.`characteristicTypeId` = 900000000000006009)
 AND (`r`.`sourceId` = [some-concept-id])));
```

Figure 109: Showing the defining Relationships of a concept

A simplification of the defining *Relationship* query can be used to return the *supertype parent concepts* as shown in *Figure 110*.

```
/* sv_relationship is a snapshot view of the relationship file */
/* sv_pref is a snapshot of descriptions filtered to preferred term */

SELECT `r`.`destinationId` AS `id`,`d`.`term` AS `term`,`r`.`sourceId` AS `conceptId`
 FROM (`sv_relationship` `r`
```

```
 JOIN `sv_pref` `d`
  ON ((`r`.`destinationId` = `d`.`conceptId`)))
WHERE ((`r`.`active` = 1)
 AND (`r`.`typeId` = 116680003)
 AND (`r`.`sourceId` = [some-concept-id])));
```

Figure 110: Showing the supertype parents of a concept

By swapping the *sourceId* and *destinationId* from the previous example the *subtype children* of the *concept* can be displayed as shown in *Figure 111*.

```
/* sv_relationship is a snapshot view of the relationship file */
/* sv_pref is a snapshot of descriptions filtered to preferred term */

SELECT `r`.`sourceId` AS `id`,`d`.`term` AS `term`,`r`.`destinationId` AS `conceptId`
 FROM (`sv_relationship` `r`
 JOIN `sv_pref` `d`
  ON ((`r`.`sourceId` = `d`.`conceptId`)))
WHERE ((`r`.`active` = 1)
 AND (`r`.`typeId` = 116680003)
 AND (`r`.`destinationId` = [some-concept-id])));
```

Figure 111: Showing the subtype children of a concept

7.5.3 Access to essential concept Identifiers

Terminology servers should provide efficient access to the *Identifiers* that represent *concepts* with structurally significant *Roles* within the terminology. *Table 219* lists the *concepts* that have the most clear-cut structurally significant *Roles*. A *terminology server* should enable access to these *Identifiers* by an easy to use name of enumeration. In addition a *terminology server* should provide a service that rapidly determines whether a given *concept* is a *subtype* of any of these *concepts*. It is also useful to for the *terminology server* to extend similar functionality to all direct *subtypes* of the *root concept* (| SNOMED CT Concept |) and to *subtype descendants* of | concept model attribute |.

Table 219: Essential concept Identifiers

Id	*Preferred Term*	Significance
138875005	| SNOMED CT Concept |	The *root concept*. All other *active concepts* are *subtypes* of this *concept*.
900000000000441003	| SNOMED CT Model Component |	All active metadata *concepts* are *subtypes* of this *concept*.
900000000000442005	| core metadata concept |	All enumerated values applicable to core *components* are *subtypes* of this *concept*.
900000000000454005	| foundation metadata concept |	All *reference sets* and all *reference set* related metadata *concept* are *subtypes* of this *concept*.
900000000000455006	| reference set |	All *reference sets* are *subtypes* of this *concept*.

Id	*Preferred Term*	Significance
116680003	I is a I	The *Attribute* used to specify the *subtype Relationship* between *concepts*.
246061005	I attribute I	All *Attribute* (*relationship type*) *concepts* are *subtypes* of this *concept*.
410662002	I concept model attribute I	With the exception of the *subtype Relationship* (see above) all *relationship types* that are used in the *SNOMED CT Concept Model* are *subtypes* of this *concept*.
370136006	I namespace concept I	Each *subtype* of this *concept* represents an extension namespaces allocated by the *IHTSDO*.
363743006	I navigational concept I	*Subtypes* of this *concept* to provide nodes in *navigation hierarchies*. They act as grouper categories that do not have any semantic meaning and thus do not appear elsewhere in the *SNOMED CT* hierarchy.

7.6 User Interface Terminology services

This section of the guide is concerned with *Terminology services* that allow users to view and select of *SNOMED CT Concepts* and *Descriptions*.

7.6.1 Text Searches

Effective implementation of *SNOMED CT* depends on the speed and simplicity with which users can locate the *terms* and *concepts* that they wish to use. A busy clinical user may become frustrated if the content they need cannot be quickly located when they search using familiar words or phrases. For this reason an efficient search strategy should address the following issues:

- Speed of search:
    - Search speed should be optimized by use of appropriate indexes.
- Search should not be too sensitive to word *order* or exact phrasing:
    - Search should be insensitive to word - *order* variants:
        - For example, "head pain" for I pain in head I
    - Allow use of acronyms or abbreviations for frequently used *terms*:
        - For example, "MI" for "myocardial infarction" or "mitral incompetence".

- Search should take account of word form variants:
    - For example, "inflamed", "inflammatory", "inflammation".
- Excessive search results should not hinder selection of the required *concept*:
    - When several *synonyms* of the same *concept* match the search key, only one should be displayed.

The purpose of this section of the implementation guide is to describe strategies a developer might use to implement the search requirements outlined above.

The *SNOMED CT* Developer Toolkit contains several files, which help to support efficient search mechanisms. These include the *Excluded Words Table*, four *keyword* indexes and the *Word Equivalents Table* summarized by *Table 32*.

7.6.1.1 Single keyword index

The single *keyword* table, (DescWordKey), provides a pointer from each *keyword* used in any *Description*, to the *Descriptions* in which that *keyword* is used. The purpose of the single *keyword* index is to support a search capability, which is independent of the *order* in which words appear in a *description*. The single *keyword* index represents the minimum necessary supporting structure for searches on *SNOMED CT* content. Searches involving target words that appear in many *descriptions* may be unacceptably slow if searches are carried out using the single *keyword* index alone. Developers wishing to produce applications with faster search times are encouraged to supplement their system with a multiple *keyword* index such the DescDualkey table (see *Word Search Tables*) provided as part of the *SNOMED CT release*.

Note that some words that are used in *description* are linking words, which are unlikely to be in the target of a search. These words are not considered to be *keywords* and may be excluded from the *keyword* index. They are found in *Excluded Words* File.

7.6.1.1.1 Generating the single keyword index

Although single *keyword* indexes are available as part of the *International Release*, developers need to know how to add *keyword* entries for any locally generated *descriptions* added as part of an *Extension*.

Entries may be added to the single *keyword* table by following the method outlined below.

For each *description*, parse the text of the *term*:

- To avoid inappropriate case mismatches, convert all characters to the same case.
- Extract words by breaking at spaces, punctuation marks, and brackets.
- For each word:
    - If the word is not in a list of *excluded words*, add a row to *keyword* table.

7.6.1.1.1.1 Example: Generation of keywords for a sample Description

Table 220: Sample Description

Description Identifier	*Concept Identifier*	*Term*
22565018	13185000	I pyrogallol 1,2-oxygenase I

- Convert all characters to the same case.

I pyrogallol 1,2-oxygenase I -> "PYROGALLOL 1,2-OXYGENASE"

- Extract words by breaking at spaces, punctuation marks, and brackets.

"PYROGALLOL 1,2-OXYGENASE" -> ' (1) = "PYROGALLOL"

(2) = "1"

(3) = "2"

(4) = "OXYGENASE"

- For each word:
  - If the word is not in a list of *excluded words*, and length of word > 1, and first character is not numeric:
    - Add a row to *keyword* tables;
    - Only the first eight characters are used in the *keyword*.

Table 221: DescKey Words

KeyWord	*Description Identifier*
PYROGALL	22565018
OXYGENAS	22565018

Table 222: ConcKeyWords

KeyWord	*Concept Identifier*
PYROGALL	13185000
OXYGENAS	13185000

7.6.1.1.2 Search using the single keyword index

A single *keyword* search may be conducted as follows:

- The user-typed search *string* is converted to consistent case;
- The *string* is parsed, breaking at spaces and punctuation characters;
- One word is selected from the parsed word list to use as a look-up on the single *keyword* index;
- Look-up on the single *keyword* index may be "exact" or "starts with," depending on wild card conventions used in the search *string*.

7.6.1.1.2.1 Example: Search using single key-word index

The user searches for "Hip* replacement*" (where "*" represents the wild card for any number of extra characters).

- The user-typed search *string* is converted to consistent case.

"Hip* replacement" -> "HIP* REPLACEMENT*"

- The *string* is parsed, breaking at spaces and punctuation characters.

"HIP* REPLACEMENT*" -> (1) "HIP*"

(2) "REPLACEMENT*"

- Look up "HIP" on the single *keyword* index using "starts with" *query*.

Table 223: Example results for a Search for "hip"

Count	*Description Identifier*	*Concept Identifier*	*Term*
1	49926016	29836001	| hip |
2	196344018	24136001	| hip |
3	2296013	736004	| abscess of hip |
4	1480791012	386649003	| partial hip replacement by prosthesis |
. . . . .	. . . .	. . . .	. . . .
315	371616001	1210239015	| methenamine hippurate 1g tablet |

Descriptions in the search results are converted to consistent case and screened, to see if they contain any words starting with "REPLACEMENT" - only those terms that do are included in the final search results.

Using a *Dual Key* index is more efficient as the same search finds only 11 matches.

Table 224: Sample results of a search for "hip replacement" using DualKey "HIPREP"

Count	*Description Identifier*	*Concept Identifier*	*Term*
1	1480791012	386649003	| partial hip replacement by prosthesis |
2	33592011	19954002	| total replacement of hip with use of methyl methacrylate |
3	50150016	29969002	| replacement of acetabulum of hip |
4	54398014	32581000	| partial hip replacement by cup with acetabuloplasty |
. . . . .	. . . .	. . . .	. . . .
11	183737015	112728000	| total revision of hip replacement with use of methyl methacrylate |

7.6.1.2 Multiple keywords

The performance of single *keyword* searches is highly dependent on the number of candidate *descriptions* returned by the *keyword* for subsequent filtering. The extremely high number of matches for some words in common use makes it likely that some searches will be unacceptably slow.

One way to alleviate this problem would be to create a table containing a row for all combinations of word pairs in each *description*. In some database environments that support optimization of multiple key searches, this may offer no benefits. However, in other environments, such a table may substantially speed searches.

A comprehensive word pair table would be very large. Such a table covering the full content of *SNOMED CT* would contain approximately 1.5 million unique word pairs and 6 million rows. Limiting the unique keys to the first three letter of each word reduces the table size to a more readily optimized set of keys. This requires the final part of the search to be conducted using text comparison (since the keys are incomplete).

7.6.1.2.1 Generating the DualKey index

Although *Dualkey* indexes are available as part of the Developer Toolkit , it is important to know how this table is generated. *SNOMED CT* users that generate *Extensions* should follow the method outlined below to generate new entries in the *Dualkey* index, based on the *descriptions* in the *Extension*.

For each *description*, parse the text of the *term*:

- To avoid inappropriate case mismatches, convert all characters to the same case;
- Extract words by breaking at spaces, punctuation marks, and brackets;
- For each word of three characters or more that is not in the list of *excluded words*, extract the first 3 characters, and arrange the word fragments in alphabetical *order*;
- Generate the dual keys for this *description* by concatenating each word fragment with those that come after it in the list;
- For each dual key, add a row to the word pair tables.

7.6.1.2.1.1 Example: Generation of keywords for a sample Description

Table 225: Sample Description

Description Identifier	*Concept Identifier*	*Term*
33592011	19954002	Total replacement of hip with use of methyl methacrylate

- To avoid inappropriate case mismatches, convert all characters to the same case.

“TOTAL REPLACEMENT OF HIP WITH USE OF METHYLE METHACRYLATE”

- Extract words by breaking at spaces, punctuation marks, and brackets.

1. TOTAL;
2. REPLACEMENT;
3. OF;
4. HIP;
5. WITH;
6. USE;
7. OF;
8. METHYLE;
9. METHACRYLATE.

- For each word of three characters or more, that is not in the list of *excluded words*, extract the first 3 characters, and arrange the word fragments in alphabetical *order*.

1. HIP;
2. MET;
3. REP;
4. TOT;
5. USE.

Note:

"OF" is less than 3 characters and is an *excluded word*, "WITH" is an *excluded word* and "MET" is duplicated, so we only include it once.

- Generate the dual keys for this *description* by concatenating each word fragment with those that come after it in the list;
- For each dual key, add rows to the word pair tables.

Table 226: DescDualKey

Dual key	*Description Identifier*
HIPMET	33592011
HIPREP	33592011
HIPTOT	33592011
HIPUSE	33592011
METREP	33592011
METTOT	33592011
METUSE	33592011
REPTOT	33592011
REPUSE	33592011
TOTUSE	33592011

Table 227: ConcDualKey

Dual key	*Concept Identifier*
HIPMET	19954002
HIPREP	19954002
HIPTOT	19954002
HIPUSE	19954002
METREP	19954002
METTOT	19954002

Dual key	***Concept Identifier***
METUSE	19954002
REPTOT	19954002
REPUSE	19954002
TOTUSE	19954002

7.6.1.2.2 Searching for Descriptions using the DualKey index

A search on the dual key index can only be carried out if the user enters a search *string* that contains at least two word fragments both of which are three characters or more in length. If the search *string* does not meet this criterion, the single *keyword* search mechanism must be used.

- The user-typed search *string* is converted to consistent case;
- The *string* is parsed, breaking at spaces and punctuation characters;
- For each word of three characters or more, extract the first 3 characters, and arrange the word fragments in alphabetical *order*;
- Create a dual key by concatenating the first two 3 letter word fragments;
- Use this dual key to look up exact matches on the word pair index;
- *Descriptions* found by searching on the word pair index are screened, to see if they contain the complete words in the original search *string*

7.6.1.2.2.1 Example: Search using word pair index

User searches for "PYRO* 1 OXYGEN*".

- The *string* is parsed, breaking at spaces and punctuation characters.

1. "PYRO*";
2. 1;
3. "OXYGEN*".

- For each word of three characters or more, extract the first 3 characters, and arrange the word fragments in alphabetical *order*.

1. "OXY";
2. "PYR".

- Create a dual key by concatenating the first two 3 letter word fragments.

OXYPYR

- Use this dual key to look up exact matches on the word pair index.

Table 228: Sample results of a search for "PYRO* 1 OXYGEN*"

Dual key	***Description Identifier***	***Description***
OXYPYR	1969019	| 2,5-Dihydroxy-pyridine oxygenase |
OXYPYR	22565018	| pyrogallol 1,2-oxygenase |

Dual key	*Description Identifier*	*Description*
OXYPRY	104951019	| 2,5-Dihydroxy-pyridine oxygenase |

- *Descriptions* found by searching on the word pair index are screened, to see if they contain the complete words in the original search *string* :
  - *Description* 1969019 is eliminated since it does not contain the word "1";
  - *Description* 104951019 is eliminated, it does not contain the word "1" or any word beginning with the *string* "pyro".

7.6.1.3 Using word equivalents to enhance searches

In healthcare, there are many words with equivalent meanings. *Synonyms* provide alternative phrases referring to the *concept*. However, *synonyms* are not created automatically for every possible combination of words with an equivalent meaning. The success of simple searches using one or more *keywords* depends on the text of the available *descriptions*. Therefore searches will fail or will be incomplete where a different equivalent word is used in the search.

For example: "Kidney stone" and "Renal calculus" are synonymous *descriptions* in *SNOMED CT*. A search of *SNOMED CT* for the target phrase "kidney stone fragmentation" yields the result "Percutaneous nephrostomy with fragmentation of kidney stone," while a search for "Renal stone fragmentation" yields no results.

One way of addressing this problem is to maintain a table of *word equivalents*. A table of this type is a prerequisite for exhaustive *synonym* generation. An initial set of *word equivalents* is included in the *SNOMED CT* Developer Toolkit. Individual implementers will wish to add additional *word equivalents* to meet the requirements of their particular medical specialty or user needs. This table is an additional resource to assist searching and parsing of phrases. It need not be a comprehensive dictionary of words. Many searches can be completed without reference to this table so it need not contain every word or equivalent phrase used in *SNOMED CT*.

Several factors complicate the initial population and subsequent use of the *word equivalents table*:

- A phrase of two or more words may be equivalent to a single word.

Example:

"Endoscopic esophagus examination" is equivalent to "esophagoscopy"

- A word may have more than one meaning, and in this, only one meaning of a pair of words may be equivalent. Thus an apparent enhancement of a search may in practice lose some of the specificity of the intended search.

Example:

"Tap" and "aspiration" are equivalent in the context of *terms* such as "pleural tap", "pleural aspiration", but not in the context of a "patella tap", a physical "tap" on a bag or catheter, or the clinical disorder "neonatal aspiration syndrome".

- When searching using incomplete words and/or wildcards, use of *word equivalents* may impede effective searches by increasing the number of spurious potential matches. This either extends the processing required to filter the real matches from the potential matches or increases the length of the list of choices presented to the user.

A wise system developer will allow the user to customize their search options, enabling searches to be narrowed, or extended to meet the needs of varying circumstances.

7.6.1.3.1 Example: Using word equivalents table to extend a failed search

A system user enters the search *string* "Fragmentation of renal calculus;" the search returns no results. The search application that the user has been provided with has the option to extend the search by using the *word equivalents table*. The user selects this option and searches again using the same search *string*.

The *word equivalents table* contains the following relevant entries:

Table 229: Word Equivalents Table Example

WordBlockNumber	*WordText*	*WordType*
1021	KIDNEY	2 (*word equivalent*)
1021	RENAL	2 (*word equivalent*)
4430	CALCULUS	2 (*word equivalent*)
4430	CALCULI	1 (word form variant)
4430	STONE	2 (*word equivalent*)
9870	RENAL STONE	4 (equivalent phrase)
9870	KIDNEY STONE	4 (equivalent phrase)
9870	KIDNEY CALCULUS	4 (equivalent phrase)
9870	RENAL CALCULUS	4 (equivalent phrase)
9870	NEPHROLITH	2 (*word equivalent*)

The table is used to make substitutions in the search *string* to produce all possible unique search variants:

"Fragmentation of renal calculus"

"Fragmentation of renal stone"

"Fragmentation of kidney stone"

"Fragmentation of kidney calculus"

"Fragmentation of Nephrolith"

"Fragmentation of renal calculus"

"Fragmentation of renal calculi"

"Fragmentation of kidney calculi"

These 8 search *strings* are used as the target phrase for *keyword* searches on the word pair index. Results from all 8 searches are combined, and duplicate *concepts* are eliminated, giving the final list of search results.

7.6.1.4 Rationalizing searches that return duplicate hits

In the previous sections of this guide, we have considered methods of ensuring that searches on a target phrase maximize the possibility of finding the *concept* that the system user requires. It is equally important to prevent the search results from containing excessive matches, since these will require filtering by the user, imposing an additional burden. Some strategies for limiting the number of search results displayed are discussed in the following sub-sections.

7.6.1.4.1 Avoiding multiple hits on the same concept

In many instances several *synonyms* associated with the same *concept* contain the same *keyword*. The designer of search software may consider filtering the output of search results so that only the first matching *description* for a *concept* is displayed.

Example:

"Endoscopic examination of the stomach" and "endoscopy of the stomach" are *synonyms* of the same *concept*. A search for the target phrase "endo* stomach" would return the first phrase found during the search. The second would be excluded, since it has the same *concept identifier* as an existing match for this search.

7.6.1.4.2 Constraining and extending search parameters

User configurable options may be one way of limiting search results. Three possible methods of limiting search results through user configurable options are suggested here:

- Limiting searches to exact matches unless wild cards are used. A search on a single word may produce many matches if it is assumed that the user is searching for any phrase that contains the target word. Forcing the use of wild cards for this kind of search can help avoid this problem.
- Make searches that include use of " *word equivalents*" a user configurable option that can be used to extend or constrain a search.
- Display search results a few at a time, with most frequently used *descriptions* listed first. This option will require the application to track the frequency of *term* selection so that search results can be sorted in this way.

7.6.2 Hierarchical Navigation

This section of the guide describes the *Terminology services* that are likely to be required to navigate *SNOMED CT* hierarchies.

One of the key strengths of *SNOMED CT* | is a | rich set of *relationships* that connect the *concepts* within the terminology. The primary use of these *relationships* is to facilitate selective retrieval. However, some of these *relationships* are arranged in hierarchies that can be navigated using an appropriate user-interface control. For example, the *subtype hierarchy* formed by the | is a | *relationships* can be used to navigate from a selected *concept* to another *concept* that has a more specific or less specific meaning.

SNOMED CT also specifies standard ways to represent multiple *navigation* hierarchies that can be designed to meet different requirement. Unlike *relationship* based hierarchies, *navigation* hierarchies convey no semantic information but are intended to be used to enhance the user experience when navigating through the terminology.

7.6.2.1 Access to hierarchically related concepts

Terminology servers should enable client applications to access collections of *Concepts* that are related to a specified *Concept* as:

- *Subtype children*
- *Subtype descendants* (includes all generations of *children*);
- *Supertype parents*
- *Supertype ancestors* (includes all previous generations of parents).

7.6.2.2 Using | is a |Relationships for hierarchy navigation

7.6.2.2.1 The SNOMED CT hierarchy

The "*SNOMED CT hierarchy*" refers to the organization of *concepts* in *SNOMED CT* from the general, at the top of the *hierarchy*, to the more specific or "granular" at the bottom. The *concepts* that make up the very top level of the *hierarchy* are shown in *Table 230*. All other *SNOMED CT concepts* fall under one or more of these categories.

Table 230: Top Level *Concepts*

• *| Clinical finding |* • *| Procedure |* • *| Observable entity |* • *| Body structure |* • *| Organism |* • *| Substance |* • *| Pharmaceutical / biologic product |* • *| Specimen |* • *| Special concept |* • *| Linkage concept |*	• *| Physical force |* • *| Event |* • *| Environment or geographical location |* • *| Social context |* • *| Situation with explicit context |* • *| Staging and scales |* • *| Physical object |* • *| Qualifier value |* • *| Record artifact |*

Several levels of increasingly fine categorization may exist between the top level of the *hierarchy* and *concepts* that have sufficient detail to be recorded in a patient's medical record. *Figure 112* shows the levels of *hierarchy* that exist between the top-level *Concept* | Clinical finding | and the finding "Catatonic reaction."

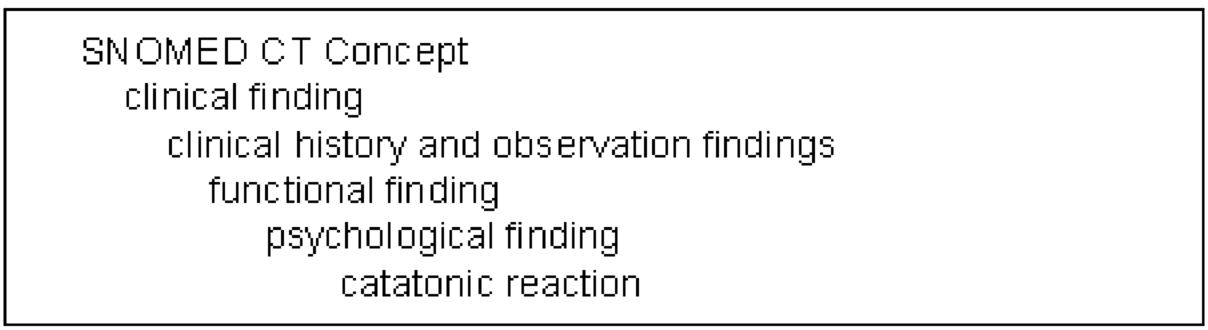


Figure 112: Hierarchy example: Catatonic Reaction

7.6.2.2.2 Hierarchy Representation in the relationships table

The *SNOMED CT Relationship* table represents *relationships* between one *SNOMED CT concept* and another by including a row in the table for each such *relationship*. The columns *sourceId*, *typeId* and *destinationId* define the source of the *relationship*, the kind of *relationship* that exists and the target of the *relationship* respectively. Each of these fields, contains a *SNOMED CT Concept Identifier*. Hierarchical *relationships* are expressed by linking the source *concept* to its "parents" (i.e. the *concept* or *concepts* immediately above it in the *hierarchy*). The *typeId* used to represent the *subtype hierarchy* is the | is a | *relationship*.

For example, we can say | catatonic reaction | | is a | | psychological finding |. This is expressed in the *Relationship* Table as follows:

Table 231: Subtype Relationship Example

sourceId	***typeId***	***destinationId***
102909009	116680003	116367006

Where:

- 102909009 is the *concept identifier* for | catatonic reaction |;
- 116680003 is the *concept identifier* for the | is a | *relationship*;
- 116367006 is the *concept identifier* for | psychological finding |.

Conversely, by inverting the | is a | *relationship* we can find the *children* of the target *Concept*, (i.e. the *Concept* or *Concepts* immediately below it in the *hierarchy*).

7.6.2.2.3 Using | is a |Relationships to enhance search capabilities

This section is concerned with the ways in which the *hierarchy* can be used to help a *SNOMED CT* user when they are searching or browsing the terminology.

Note: The primary use of the *SNOMED CT subtype hierarchy* is to support effective retrieval and aggregation of data. This is discussed in *Testing and traversing subtype relationships*.

It is possible to start at the top of *hierarchy* and navigate from parent to *child* in order to find a *Concept* or *term* in *SNOMED CT*. A more efficient approach, however, is to use the *hierarchy* to supplement a *keyword* search by enabling the user to look at related *Concepts* in order to consider them as alternative matches, or to check the context of a search result. The following examples illustrate these two uses of the *SNOMED CT hierarchy*.

Example:

1. Checking supertypes:

   - A user wishes to find a *description* that relates to the condition of a patient who is hypersensitive to an allergen. The user performs a search on the *keyword* "Hypersensitivity" and finds an exact match. Before the user selects the *description* for inclusion in the patient record, they check the *Fully Specified Name*, which is "Sensitivity (finding)." The user then checks the *hierarchy* and discovers that the selected *Concept* has "Psychological finding" as an *ancestor*, which indicates that this is not the correct *description* to use in this context.

2. Checking *subtypes*:

   - A user wishes to find a *description* that relates to the condition of a patient who is hypersensitive to an allergen. The user searches for the *keyword* "allergy," and finds one *Concept* having a *description* that is an exact match. The user then looks at the *children* of the *Concept* (i.e. those *concepts* immediately below it in the *hierarchy*). One of the *children* has the preferred *description* "Contact Hypersensitivity" which matches the user's intended meaning. The user selects this *Concept* for inclusion in the patient record.

7.6.2.2.4 Using | is a |Relationships to display hierarchical information in applications

Most visual application development tools contain a *component* designed to display hierarchical information as a tree in which branches can be expanded or collapsed. Tree views are well-suited to displaying *SNOMED CT* hierarchical *Relationships* (see *Figure 113*). These views are used in many different user-interfaces where information needs to be represented as a hierarchy (e.g. displaying a file-system as a hierarchy of folders or providing a collapsable outline of a document or help file). Therefore, most users will already be familiar this paradigm.

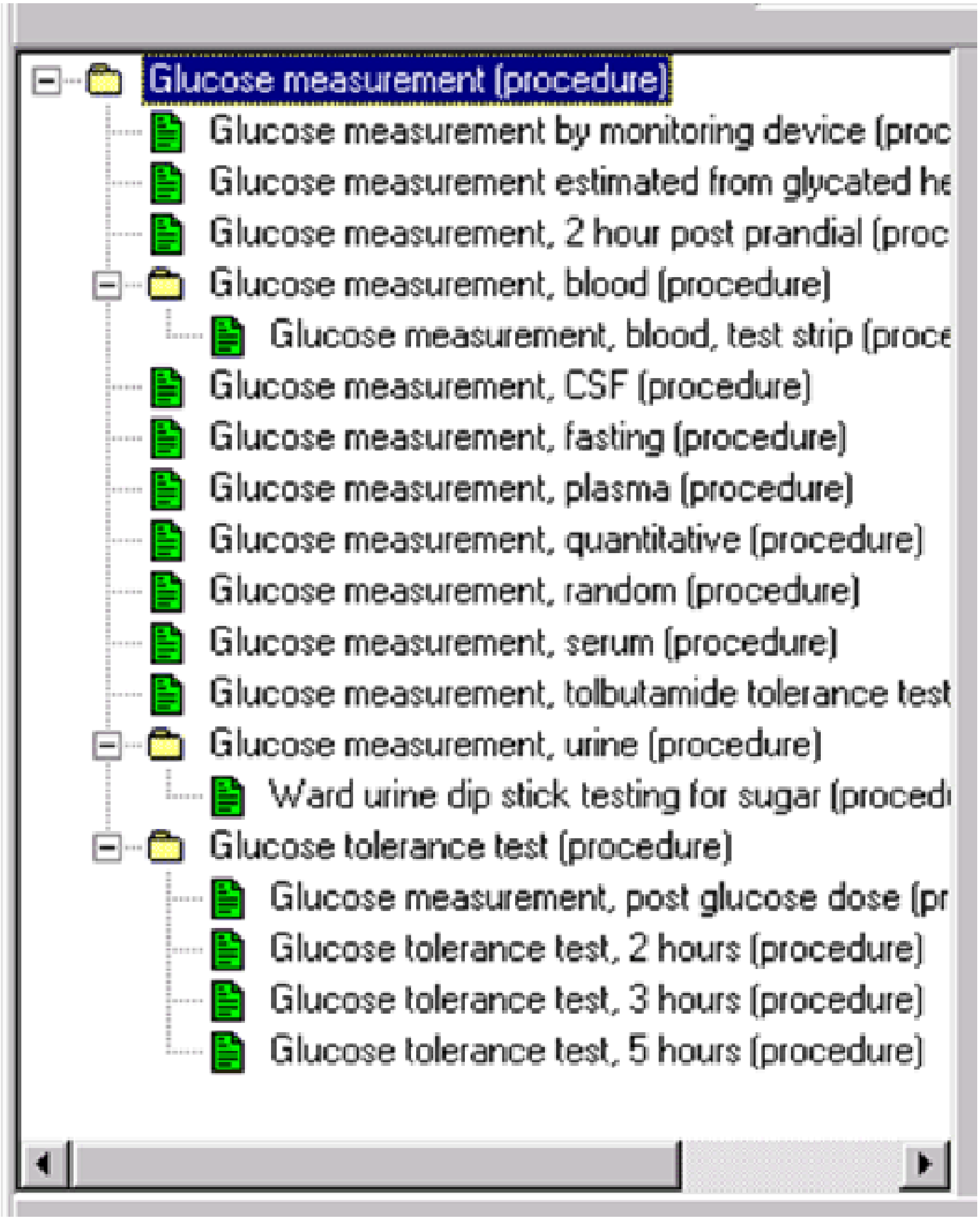


Figure 113: SNOMED CT hierarchy represented in a tree view

The process of creating a tree view from the *SNOMED CT Relationship* table is straightforward as long as a few simple ideas are mastered:

- Most standard tree-views controls start from a single root and require that higher level branches must be added before sub-branches. This means that when viewing part of the *hierarchy* from the bottom up, the tree must be compiled in temporary form before it can be displayed.

- Since the depth of the *hierarchy* is not known in any particular case, operations that iterate up or down the depth of the *hierarchy* must be done using a recursive algorithm. However, this recursion must usually be limited since placing the entirety of the *SNOMED CT* hierarchy in a single tree control is likely to create performance issues and may exceed physical limits on the capacity of the control.
- Standard tree view controls are not good at displaying the multiple parent nodes that occur in a *polyhierarchy* like *SNOMED CT*. Therefore, some compromises need to be made to present options for navigation up the hierarchy.
- Effective use of some tree controls requires unique keys for each node. Multiple parents and multiple roots through the hierarchy mean that the same *Concepts* will appear in multiple places in the hierarchy. Therefore, the *concept identifier* cannot be used to provide a key that is globally unique within the hierarchy.

7.6.2.3 Using | Part of |Relationships for hierarchy navigation

In addition to the *subtype hierarchy* represented by | is a | *relationships*, *SNOMED CT* also represents a partonomy hierarchy using | Part of | *relationship*. This creates an alternative hierarchy which can be also be used for navigation. The difference between these hierarchies is that:

- The *subtype hierarchy* relates *concepts* to supertypes that represent more general *concepts*. Each body structure *concept* has an | is a | *relationship* to one or more *concepts* that represents the *whole or any part of* the organ or other body part that contains it. *Concepts* that represent the *whole or any part of* an organ or body part are distinguished by their *fully specified names* which include the word 'structure'. These contrast with *concepts* that represent the entirety of an organ or body part which contain the word 'entire'.

  Example: | Right ventricular structure | | is a | | heart structure |

- The partonomy hierarchy relates body structure to *concepts* to *concept* that represent | the entirety of | or an organ or anatomical structure of which they form part

  Example: | Entire right ventricle | (is) | part of | | entire heart |

Note: In everyday speech the word "heart" may mean either | heart structure | or | entire heart | and the distinction between them is often overlooked. However, from a semantic perspective the difference is highly significant. The removal of some part of an organ does not imply the removal of the entire organ. Thus, while it is correct to state that | Right ventricular structure | | is a | | heart structure |, it would be wrong to state that | Entire right ventricle | | is a | | entire heart | or | Right ventricular structure | | is a | | entire heart |.

7.6.2.4 Using other Relationships to navigate SNOMED CT content

Many *SNOMED CT Concepts* have *relationships* with content in other areas of terminology. These *Relationships* are one of the ways in which *SNOMED CT* provides computer readable definitions for medical *concepts*. For example, diseases in *SNOMED CT* generally have a *Relationship* to the body site affected by the disorder and a *Relationship* to the morphology associated with the disease. Procedures in *SNOMED CT* might have *Relationships* to the *concept*, which defines the type of surgical action being carried and the procedure site, for example. Examples of *Relationships* for a disease and a procedure are shown below. A full list of the *Relationships* that can be used for each type of *Concept* can be found in *Table 103*.

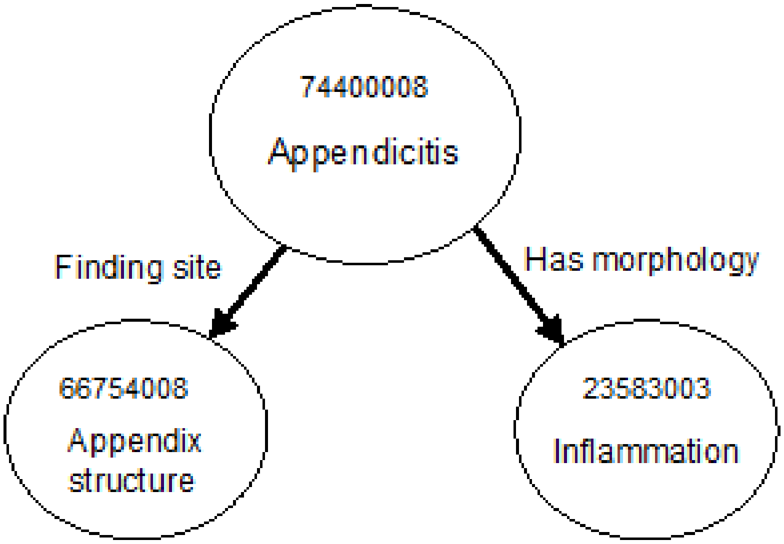


Figure 114: Relationship for disease appendicitis

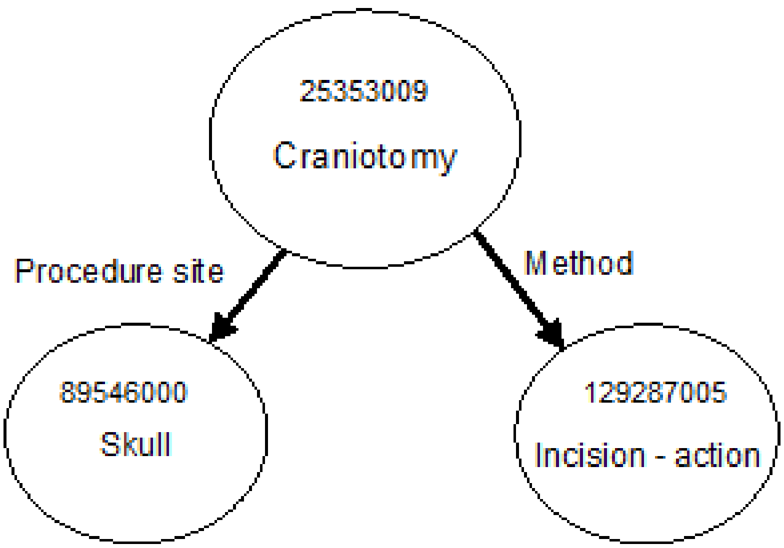


Figure 115: Relationships for procedure craniotomy

These *Relationships* are very useful in the context of data retrieval and analysis. The *Relationships* can also be used to aid in the search for specific *SNOMED CT Concepts* in cases where the *term* alone may not sufficiently distinguish between choices. For example, a search for all inflammatory diseases of the lung could be carried out as follows:

- Use the *hierarchy* to compile a list of all *Concepts* that are lung structures;
- Search for any *Concept* that has a row in the *relationship file* with a *Relationship Type* of "Disorder site," and with *ConceptId2* included in the list of lung structures;
- Now exclude any procedures from the list that do have the | Associated morphology | "Inflammation" in the *Relationship* table;
- Final product | is a | list of all lung disorders that involve inflammation.

To achieve these same results with a string search we would have to perform separate searches for | pneumonia |, bronchitis, | pleurisy | and many other conditions that cannot be linked via a sample string search.

7.6.2.5 Implementing Navigation Hierarchies

This section demonstrates how an *Ordered Reference Set* is used to specify and display a customized *navigation hierarchy*. A *navigation hierarchy* is a hierarchical view of *SNOMED CT concepts* which may differ from the strict *subtype hierarchy* (represented by | is a | *relationships*).

7.6.2.5.1 Navigation Hierarchy Example

To illustrate the way a *navigation hierarchy* is represent this section uses an example containing a set of *concepts* used to describe x-ray examinations of the upper and lower limbs. The resulting *navigation hierarchy* might usefully be extended to include other x-ray procedures but has been kept small for the purposes of the example.

Reference sets are created as explained in *how to create a new Reference Set using an existing pattern* and their *Identifier*, name and type are specified by a *concept*. The example *reference set* would be specified by a *concept* with the characteristics shown in *Table 232*.

Table 232: Concept specifying the Example Navigation Reference Set

Characteristic	Value	Comment
id	<RefsetId-A>	These symbolic values are used to avoid any potential confusion with released *reference sets*. The *moduleId* represents the module in which the *reference set* was developed.
moduleId	<ModuleId-A>	
preferredTerm	Example Navigation *Reference Set*	
| is a |	| Ordered type reference set |	

The *concepts* included in the *reference set* are shown with their *preferred terms* in *Table 233*.

Table 233: Concepts used in the Example Navigation Reference Set

Id	*Preferred Term*
1225002	| radiography of humerus |
1597004	| skeletal X-ray of ankle and foot |
168594001	| clavicle X-ray |
168619004	| plain X-ray head of humerus |
168620005	| plain X-ray shaft of humerus |
168623007	| X-ray shaft of radius/ulna |
168637003	| plain X-ray radius |
168655007	| instability views carpus |

Id	*Preferred Term*
168663008	| plain X-ray head of femur |
168664002	| femoral neck X-ray |
168665001	| plain X-ray shaft of femur |
168669007	| patella X-ray |
205115004	| radiologic examination of femur, anteroposterior and lateral views |
241063007	| bicipital groove X-ray |
241066004	| ulna groove X-ray |
241069006	| ulna X-ray |
241071006	| scaphoid X-ray |
241073009	| metacarpal X-ray |
241075002	| femur X-ray |
241076001	| tibia and/or fibula X-ray |
241077005	| tibia X-ray |
241078000	| fibula X-ray |
241079008	| metatarsal X-ray |
241080006	| tarsus X-ray |
268427003	| X-ray shaft of tibia/fibula |
271311001	| carpal bones X-ray |
302402006	| radius and/or ulna X-ray |
37815002	| diagnostic radiography of calcaneus |
40348008	| skeletal X-ray of pelvis and hip |
418687005	| fluoroscopy of humerus |

Id	*Preferred Term*
427961005	| x-ray of acetabulum |
432552002	| computed tomography of clavicle |
48966008	| skeletal X-ray of shoulder and upper limb |
5433008	| skeletal X-ray of lower limb |
70780000	| skeletal X-ray of elbow and forearm |
72872009	| skeletal X-ray of upper limb |
79082005	| diagnostic radiography of fibula, combined AP and lateral |
82420003	| radiologic examination of forearm, anteroposterior and lateral views |

The members of the example *reference set* would be distributed in a file with a name like:

- xder2_ciRefset_NavigationRefsetExampleFull_XX_20100731.txt.

The content of this file is shown in *Table 234* and the resulting hierarchical display is shown in *Figure 116*.

Table 234: Example Navigation Reference Set File

id	*effectiveTime*	active	*moduleId*	refsetId	*referencedComponentId*	order	linkedId
<uuid-0211>	20100731	1	<ModuleId-A>	<RefsetId-A>	<RefsetId-A>	1	5433008
<uuid-0212>	20100731	1	<ModuleId-A>	<RefsetId-A>	5433008	1	241080006
<uuid-0213>	20100731	1	<ModuleId-A>	<RefsetId-A>	241080006	1	37815002
<uuid-0214>	20100731	1	<ModuleId-A>	<RefsetId-A>	5433008	2	241079008
<uuid-0215>	20100731	1	<ModuleId-A>	<RefsetId-A>	5433008	3	241076001
<uuid-0216>	20100731	1	<ModuleId-A>	<RefsetId-A>	241076001	1	241078000
<uuid-0217>	20100731	1	<ModuleId-A>	<RefsetId-A>	241078000	1	268427003
<uuid-0218>	20100731	1	<ModuleId-A>	<RefsetId-A>	241078000	2	79082005
<uuid-0219>	20100731	1	<ModuleId-A>	<RefsetId-A>	241078000	3	241077005
<uuid-0220>	20100731	1	<ModuleId-A>	<RefsetId-A>	5433008	4	241075002

id	*effectiveTime*	active	*moduleId*	refsetId	*referencedComponentId*	order	linkedId
<uuid-0221>	20100731	1	<ModuleId-A>	<RefsetId-A>	241075002	1	205115004
<uuid-0222>	20100731	1	<ModuleId-A>	<RefsetId-A>	241075002	2	168665001
<uuid-0223>	20100731	1	<ModuleId-A>	<RefsetId-A>	241075002	3	168664002
<uuid-0224>	20100731	1	<ModuleId-A>	<RefsetId-A>	241075002	4	168663008
<uuid-0225>	20100731	1	<ModuleId-A>	<RefsetId-A>	5433008	5	168669007
<uuid-0226>	20100731	1	<ModuleId-A>	<RefsetId-A>	5433008	6	1597004
<uuid-0227>	20100731	1	<ModuleId-A>	<RefsetId-A>	5433008	7	40348008
<uuid-0228>	20100731	1	<ModuleId-A>	<RefsetId-A>	40348008	1	427961005
<uuid-0229>	20100731	1	<ModuleId-A>	<RefsetId-A>	<RefsetId-A>	2	72872009
<uuid-0230>	20100731	1	<ModuleId-A>	<RefsetId-A>	72872009	1	302402006
<uuid-0231>	20100731	1	<ModuleId-A>	<RefsetId-A>	302402006	1	241069006
<uuid-0232>	20100731	1	<ModuleId-A>	<RefsetId-A>	241069006	1	168623007
<uuid-0233>	20100731	1	<ModuleId-A>	<RefsetId-A>	302402006	2	168637003
<uuid-0234>	20100731	1	<ModuleId-A>	<RefsetId-A>	302402006	3	70780000
<uuid-0235>	20100731	1	<ModuleId-A>	<RefsetId-A>	70780000	1	241066004
<uuid-0236>	20100731	1	<ModuleId-A>	<RefsetId-A>	302402006	4	82420003
<uuid-0237>	20100731	1	<ModuleId-A>	<RefsetId-A>	72872009	2	168594001
<uuid-0238>	20100731	1	<ModuleId-A>	<RefsetId-A>	168594001	1	432552002
<uuid-0239>	20100731	1	<ModuleId-A>	<RefsetId-A>	72872009	3	1225002
<uuid-0240>	20100731	1	<ModuleId-A>	<RefsetId-A>	1225002	1	241063007
<uuid-0241>	20100731	1	<ModuleId-A>	<RefsetId-A>	1225002	2	168620005
<uuid-0242>	20100731	1	<ModuleId-A>	<RefsetId-A>	1225002	3	168619004
<uuid-0243>	20100731	1	<ModuleId-A>	<RefsetId-A>	1225002	4	418687005

id	*effectiveTime*	active	*moduleId*	refsetId	*referencedComponentId*	order	linkedId
<uuid-0244>	20100731	1	<ModuleId-A>	<RefsetId-A>	72872009	4	168655007
<uuid-0245>	20100731	1	<ModuleId-A>	<RefsetId-A>	72872009	5	271311001
<uuid-0246>	20100731	1	<ModuleId-A>	<RefsetId-A>	271311001	1	241071006
<uuid-0247>	20100731	1	<ModuleId-A>	<RefsetId-A>	72872009	6	241073009
<uuid-0248>	20100731	1	<ModuleId-A>	<RefsetId-A>	72872009	7	48966008

Note: Each of the symbolic names '<uuid-0211>' to '<uuid-0248>' in *Table 234* represents a unique 128-bit *UUID* generated by a standard algorithm. Use of the same symbolic names elsewhere in this example indicates a revised version of the same component with the same *Identifier*. Use of the same symbolic name in another example does not imply the same *Identifier*.

- 5433008 | skeletal X-ray of lower limb |
  - 241080006 | tarsus X-ray |
    - 37815002 | diagnostic radiography of calcaneus |
  - 241079008 | metatarsal X-ray |
  - 241076001 | tibia and/or fibula X-ray |
    - 241078000 | fibula X-ray |
      - 268427003 | X-ray shaft of tibia/fibula |
      - 79082005 | diagnostic radiography of fibula, combined AP and lateral |
      - 241077005 | tibia X-ray |
  - 241075002 | femur X-ray |
    - 205115004 | radiologic examination of femur, anteroposterior and lateral views |
    - 168665001 | plain X-ray shaft of femur |
    - 168664002 | femoral neck X-ray |
    - 168663008 | plain X-ray head of femur |
  - 168669007 | patella X-ray |
  - 1597004 | skeletal X-ray of ankle and foot |
  - 40348008 | skeletal X-ray of pelvis and hip |
    - 427961005 | x-ray of acetabulum |
- 72872009 | skeletal X-ray of upper limb |
  - 302402006 | radius and/or ulna X-ray |
    - 241069006 | ulna X-ray |
      - 168623007 | X-ray shaft of radius/ulna |
    - 168637003 | plain X-ray radius |
    - 70780000 | skeletal X-ray of elbow and forearm |

        - 241066004 | ulna groove X-ray |
    - 82420003 | radiologic examination of forearm, anteroposterior and lateral views |
- 168594001 | clavicle X-ray |
    - 432552002 | computed tomography of clavicle |
- 1225002 | radiography of humerus |
    - 241063007 | bicipital groove X-ray |
    - 168620005 | plain X-ray shaft of humerus |
    - 168619004 | plain X-ray head of humerus |
    - 418687005 | fluoroscopy of humerus |
- 168655007 | instability views carpus |
- 271311001 | carpal bones X-ray |
    - 241071006 | scaphoid X-ray |
- 241073009 | metacarpal X-ray |
- 48966008 | skeletal X-ray of shoulder and upper limb |

Figure 116: Example Navigation Reference Set - Hierarchy View

This *reference set* could be updated by addition of the rows in a subsequent release. If the three rows shown in *Table 235* are added in the next version, the results are as follows:

- 48966008 | skeletal X-ray of shoulder and upper limb | is removed from the *reference set* because the row with id=<uuid-0248> and the most recent *effectiveTime* is now inactive (active=0);
- The order of 241073009 | metacarpal X-ray | and 271311001 | carpal bones X-ray | are reversed as the most recent row for id=<uuid-0245> has order=6 while the most recent row for id=<uuid-0247> has order=5.

The changed part of the hierarchy is shown in *Figure 117*.

Table 235: Example Navigation Reference Set File - Updated Rows

id	*effectiveTime*	active	*moduleId*	refsetId	*referencedComponentId*	order	linkedId
<uuid-0245>	20110731	1	<ModuleId-A>	<RefsetId-A>	72872009	6	271311001
<uuid-0247>	20110131	1	<ModuleId-A>	<RefsetId-A>	72872009	5	241073009
<uuid-0248>	20110131	0	<ModuleId-A>	<RefsetId-A>	72872009	7	48966008

- 5433008 | skeletal X-ray of lower limb |
    - ... *unchanged* ...
- 72872009 | skeletal X-ray of upper limb |
    - ... *unchanged* ...
    - 168655007 | instability views carpus |
    - 241073009 | metacarpal X-ray |
    - 271311001 | carpal bones X-ray |

- 241071006 | scaphoid X-ray |

Figure 117: Example Navigation Reference Set - Updated Hierarchy View

7.6.2.5.2 Navigation Hierarchy Inheritance

A Navigation *Reference Set* may organize some *concepts* while allowing the *subtype hierarchy* (or another *navigation hierarchy*) to provide additional hierarchical links. In this case, a *concept* that has no children in the *navigation hierarchy* inherits the children specified in the *subtype hierarchy* or a specified default *navigation hierarchy*.

7.6.2.6 Using Tree View Components for Hierarchy Display

The two examples given below show the creation of a tree view from a small sample *hierarchy*.

The principals used can be extended to any size or depth of *hierarchy*.

7.6.2.6.1 Example 1: Show all descendants of Concept "A" in a tree view

Table 236: Example Relationships

sourceId	typeId	destinationId
B	| is a |	A
C	| is a |	A
D	| is a |	B
E	| is a |	B
E	| is a |	C
C	| is a |	F

We must process each *concept* in the *hierarchy*, starting at 'A'. Add a tree node for 'A', and then *query* to get the *children* of 'A'. Process each *child* recursively, i.e. add a node to the tree view for the *child*, then *query* for its *children*, etc.

Table 237: Child nodes

Node	*Child* Node
1	2
1	5
2	3
2	4
5	4

Node	*Child* Node
5	6

Table 238: Concept to node cross reference

Node	*Concept Identifier*
1	A
2	B
3	D
4	E
5	C
6	E
7	F

Now we have tree nodes and their *children* for each *Concept*. If the nodes have been added to a Windows tree view *component*, display will be automatic. If a text-based display is being used then the nodes can be output to the screen using the indent style display. Note that the *Concept* 'E' appears in the tree view twice, under each of its parents.

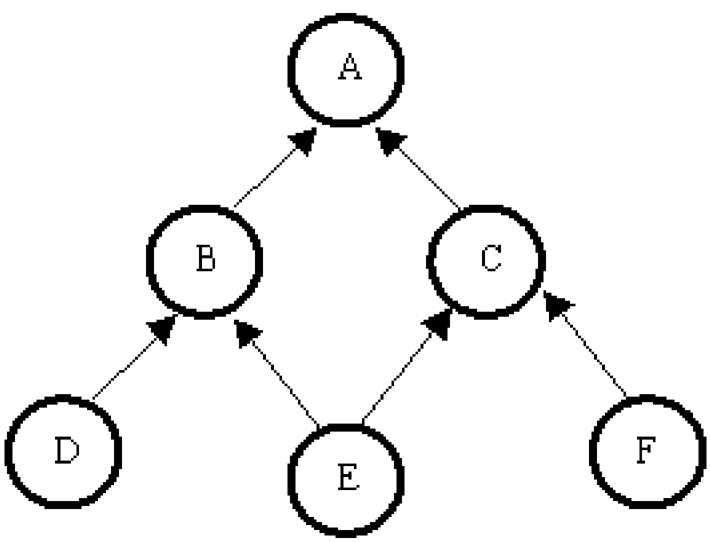


Figure 118: Tree view of sample hierarchy - descendants of "A"

7.6.2.6.2 Example 2 - Show all ancestors of Concept "E"

In order to construct the tree view, we must start from the top down, so we must create a temporary view of the *hierarchy* before we can add nodes to the tree view. Query to get the parents of 'E'. Process each parent recursively, i.e. add an entry to the temporary table, stating that 'E' I is a I *child* of each of its parents, then *query* to get its parent, etc. When the top of the tree is reached, a record is kept of the top-level *concept*, since this will be the starting point for building the tree view.

Table 239: Temporary view of the hierarchy

Concept	*Child Concept*
B	E
C	E
A	B
A	C

We can now use the temporary table information to build the tree view from the top down. Starting at A, add a node to the tree view. Work recursively from the information in the temporary view of the *hierarchy* to add the *descendants* of 'A' into the tree view.

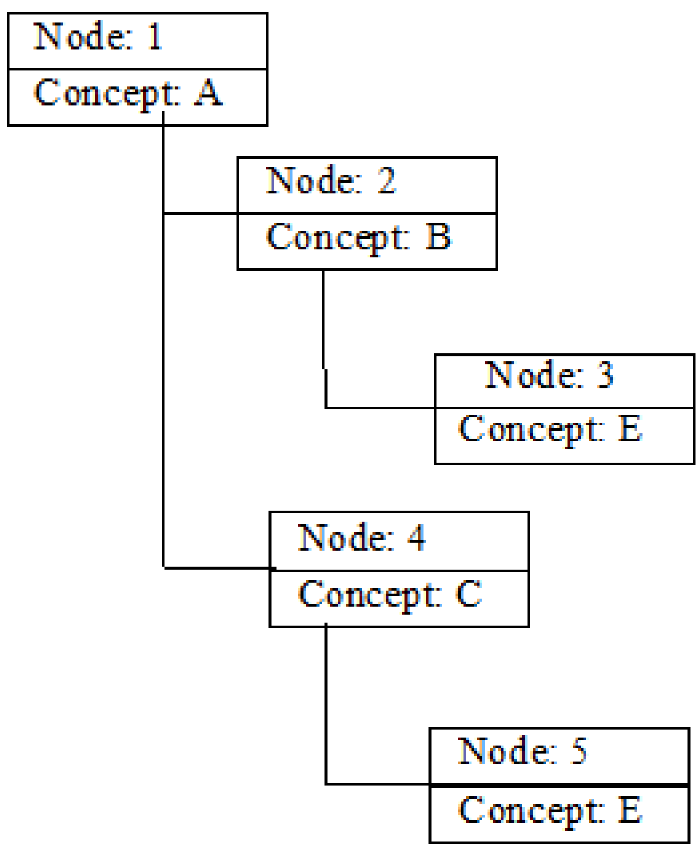


Figure 119: Tree view of sample hierarchy - ancestors of "E"

7.6.3 Applying Reference Sets

Refsets can be used for many different purposes. This section outlines some of the ways in which the *Refset* mechanism specified in this guide can be used to meet different practical requirements. The uses outlined are illustrative examples and do not represent all possible applications of *Refsets*.

7.6.3.1 Languages and Dialects

The *Language Reference Set* applicable to the local language, *dialect* and term preferences should be used to filter or prioritize the display of matching terms.

A *SNOMED CT enabled application* should be able to:

- Allow a selection of a particular language as a configuration option;
- Restrict access to *descriptions* so that are acceptable or preferred in the selected language;
- Treat the *synonym* marked as preferred in the selected language as the default *preferred term*;
- Treat the *Fully Specified Name* marked as preferred in the selected language as the *Fully Specified Name* to be displayed where relevant.

An application may support multiple *languages* or *dialects* allowing selection of combinations of Language *Reference Sets*. In this case, the primary selected language is supplemented by acceptable *Descriptions* in supplementary languages or *dialects*.

7.6.3.2 National requirements for specific Concepts

SNOMED CT is designed for use in many different countries and consequently includes a certain number of country-specific *Concepts*. Each country, may have specific requirements for the representation of *Concepts* that are not meaningful in other countries. These variations are particularly significant for the interfaces between clinical care, service administration and reimbursement. National laws and conventions may also create additional *refinements* of more general *Concepts*.

Reference Sets included in a National *Extension* release can be used to configure searches performed by *terminology services* to meet National requirements.

- *Simple Reference Sets* can be used to filter search contents;
- *Ordered Reference Sets* can be used to prioritize search result or provide a natural ordering of search results;
- *Attribute Value Reference Sets* can be used to filter information based on particular criteria represented by the *valueId* of particular members of the *Reference Set*.
- *Annotation Reference Sets* can be used to supplement the search results with information annotated to a *component* providing advice on intended use.

A *SNOMED CT enabled application* should be able to apply National *Reference Sets* to filter searches.

7.6.3.3 Regional variations in disease prevalence

There are substantial differences in the prevalence of diseases in different regions in which *SNOMED CT* may be used. Users will expect to find the conditions they commonly deal with, without being distracted by long lists of conditions they rarely see.

A *SNOMED CT enabled application* should be able to select *Reference Sets* that represent prevalence characteristics for a particular region. Based on the selected configuration searches should:

- Selectively include or exclude *concepts* or *descriptions* based on presence in or absence from a selected *Simple Reference Set*;
- Prioritize access to *concepts* or *descriptions* based on the order specified in a selected *Ordered Reference Set*.

The way in which access is prioritized depends on the nature of the application and its operating environment. However, examples of prioritization include:

- Showing *descriptions* associated with high priority *concepts* before those with lower priority when searching for word or phrases;
- Showing *concepts* with high priority before their less highly prioritized siblings in hierarchical displays;
- Initially listing *concepts* and associated *descriptions* with priority above a specified threshold and requiring an additional step to access those assigned lower priority.

7.6.3.4 Specialty and discipline-dependent variations in use of Concepts

SNOMED CT contains *Concepts* used by many different groups of health professionals. The frequency of use of these *Concepts* depends on the professional discipline and/or clinical specialty of the user. It is important to ensure that the user is able to access the *Concepts* that they use frequently, without being distracted by thousands of textually similar *Concepts* they rarely require.

A *SNOMED CT enabled application* should be able to select *Reference Sets* that represent the requirements of a particular speciality. Based on the selected configuration searches should:

- Selectively include or exclude *concepts* or *descriptions* based on presence in or absence from a selected *Simple Reference Set*;
- Prioritize access to *concepts* or *descriptions* based on the order specified in a selected *Ordered Reference Set*.

7.6.3.5 Local needs of organizations or individual users

The previous sections have dealt with requirements of countries, regions and specialties. Organizations and individual users may also have similar requirements for restricting or prioritizing access to particular *Concepts*.

A *SNOMED CT enabled application* should be able to rationally combine *Reference Sets* that represent National, Regional, specialty and local requirements. Based on the selected configuration searches should:

- Selectively include or exclude *concepts* or *descriptions* based on presence in or absence from selected *Simple Reference Sets*;
- Prioritize access to *concepts* or *descriptions* based on the order specified in the selected *Ordered Reference Sets*.

7.6.3.6 Supporting data entry protocols

Many clinical applications include facilities for data entry to be controlled or assisted by protocols, templates or structured data entry forms. Different sets of candidate *terms* or *concepts* may be appropriate to each data entry field. The sets of candidate *terms* or *concepts* for a field may be very large (e.g. any operative procedure) or very small (e.g. the possible observations from a particular examination).

Reference sets can be used to restrict the available options to match the requirements of a particular data entry protocol. *Reference Sets* provided by the author of the protocol can be applied to particular fields on a screen or particular data entry steps to configure relevant searches.

- *Simple Reference Sets* can be used to filter search contents;
- *Ordered Reference Sets* can be used to prioritize search result or provide a natural ordering of search results;
- *Attribute Value Reference Sets* can be used to filter information based on particular criteria represented by the *valueId* of particular members of the *Reference Set*.
- *Annotation Reference Sets* can be used to supplement the search results with information annotated to a *component* providing advice on intended use.

A *SNOMED CT enabled application* should be able to dynamically select a particular configuration based on identification of the data entry step or context. It should then be able to apply the relevant *Reference Sets* to provide an appropriate set of data entry options and/or to constrain text searches.

7.6.3.7 Managing the coded content of messages

A *Simple Reference Set* may be used to represent *value set* applicable to a particular field in a message. The entry of data to populate that field in the message can be constrained by filtering searches so that only *concept* in that *Reference Set* are returned.

Healthcare messages include fields that can be populated with codes from clinical coding schemes. *SNOMED CT* provides *concept identifiers* as a means of encoding *Concepts*. These *concept identifiers* are suitable for use in appropriate fields of many clinical messages.

Implementations of clinical messaging typically constrain the range of values that can be applied to particular fields. There are several reasons for this:

- To ensure that the information encoded is meaningful as a value for the specified field.

Example:

A field that is intended to describe the nature of investigation may contain a code that means "Serum glucose measurement" but should not contain a code that means "Hypoglycemia."

- To ensure that receiving application is able to process the message.

Example:

A locally added code value may be valid in a particular application but should not be used if the receiving application needs to retrieve, process or analyze the coded part of the message.

- To ensure adequate detail and specificity.

Example:

A field used to report an operative procedure could contain a code for "Abdominal procedure." However, this would not be adequate to meet the business purpose served by a message.

- To avoid unnecessary detail or diversity.

Example:

A biochemical investigation could be reported using a code that represents various detailed aspects of the method used to perform the investigation. Such details may be unnecessary to a clinician and may complicate the analysis, charting and graphing of a series of results reported at different levels of detail.

7.6.4 Access to qualifiers and refinable characteristics

A *terminology server* should enable an application to review the refinable *defining characteristics* and the specified set of *qualifying characteristics* for any selected *Concept*.

7.7 Testing and traversing subtype Relationships

The *subtype hierarchy* represented by I is a I *relationships* is an essential element in the structure and semantics of *SNOMED CT*. All *SNOMED CT enabled terminology servers* need to provide functions that test and traverse these *relationships* to navigate the hierarchy and to determine whether a *concept* is a *subtype* of another specified *concept*.

7.7.1 Top-level ancestor checking

Terminology servers should allow client applications to rapidly determine the top-level *Concept* that is the *supertype ancestor* for any specified *Concept*.

Each *Concept* has only one top-level supertype and this represents the semantic-type of the *Concept*.

7.7.2 Navigation concept checking

Terminology servers allow client applications to determine whether a specified *Concept* | is a | *navigation Concept*.

7.7.3 Subtype descendant testing

Terminology servers should be able to test whether any specified *Concept* | is a | *descendant subtype* of another specified *Concept*.

7.7.4 Subtype search scope restriction

Terminology servers should be able to restrict searches so that they return only those *Concepts* and (or their associated *Descriptions*) that are *subtype descendants* of a specified *Concept*.

Subtype search scope restriction is particularly valuable with respect to top-level *Concepts*. For example, when searching for a procedure it is useful to be able to exclude disorders or findings that may contain similar words of phrases.

Generalizing *subtype* search scope restriction to other nodes in the *subtype hierarchy* may significantly enhance usability in some situations.

Example:

When undertaking an ophthalmologic examination, a search for findings could be constrained to findings related to the eye, increasing the specificity of results of searches for phrases containing the word "fundus."

7.7.5 Optimizing concept subsumption testing

Rapid and efficient computation of whether a *concept* | is a | *subtype descendant* of another *concept* is essential for effective *transformation* of *expressions* and for testing subsumption between *expressions*.

7.7.5.1 Approaches to concept subsumption testing

The *SNOMED CT* Technical Implementation Guide discusses several strategies for delivering efficient computation of subsumption between *concepts*. These are briefly summarized here with a brief evaluation of their suitability.

7.7.5.1.1 Recursive testing of subtype Relationships

It is possible to determine whether one *concept* subsumes another *concept* by recursively following every possible sequences of | is a | *Relationships* from a candidate *concept* until the predicate *concept* is reached or until all possible paths have been exhausted.

This approach is far too slow to deliver effective implementations in all environments in which it has been tested to date.

7.7.5.1.2 Semantic type Identifiers and hierarchy flags

Flags added to the internal representation of each *Concept* can be used to indicate the set of high-level *concept* nodes of which that *concept* is a *subtype*. A *concept* can only subsume *concepts* that include the same set of high-level *concept* flags. This approach can reduce the number of tests that need to be performed to recursively test the *subtype relationships*:

- If a candidate does not have all the high-level node flags that the predicate has, no further tests are needed. The candidate is not a *subtype* of the predicate.
- Even if a candidate shares the high-level node flags with the predicate, any path that reaches a *concept* that does not share those flags need not be further tested.

While faster than the unaided recursive testing approach, this is too slow to deliver effective implementations and is not scalable.

7.7.5.1.3 Use of proprietary database features

Some databases include additional features to support the recursive testing of a chain of hierarchical *relationships*. Other methods of optimization that may be applied to allow more rapid computation of *subtype descendant relationships* are outlined in the following subsections.

Current experiences of databases that support this type of approach indicate that (while easy to implement) the performance is substantially inferior to use of branch-numbering or *transitive closure*.

7.7.5.1.4 Branch numbering

The internal representation of each *Concept* can be extended to include a branch-number and a set of branch-number -ranges.

A branch-numbering algorithm can then be applied when each release of *SNOMED CT* is imported.

A typical branch-numbering algorithm processes the *subtype hierarchy* in the following way:

- A depth first tree walk is performed starting from the root *Concept* (branch-number 1) and an incrementing number is applied to each *Concept* when it is encountered for the first time.
- After the branch numbers have been computed a further tree walk allocates one or more branch-number ranges to each *Concept* with any *subtype descendants*:
  - Many *Concepts* will have a single branch number range containing all their *descendants*.
  - Some *Concepts* will have several non-contiguous ranges of *descendant Concept* branch numbers:
    - This is because a *Concept* may have multiple supertypes. Therefore, the *descendants* of a *Concept* may have branch numbers that were allocated as a result of their *relationship* to another *ancestor Concept*. However, the path from any *Concept* to the root *Concept* always converges at or before the top-level *Concept*. Therefore, multiple ranges coalesce when reaching more general common *supertype ancestors*.
- At run time, rather than needing to traverse many *subtype Relationships*, the branch number of each *Concept* is tested for inclusion in the branch number range of the putative *ancestor*.

This approach removes the need for exhaustive testing of *subtype Relationships*. The disadvantages are a relatively complex build process that must be repeated for each release or update and a requirement for the internal *Concept* representation to accommodate a variable length representation of branch number ranges.

7.7.5.1.5 Precomputed Transitive Closure table

The *transitive closure* table is a comprehensive view of all the supertypes of every *concept*. It can be derived from current release data by traversing all | is a | *relationships* recursively and adding each inferred supertype *relationship* to a table.

The advantage of this type of view is that a *candidate - concept* can be tested for subsumption by *predicate - concept* by a simple SQL *query*. In addition, the table can be updated to take account of changes without requiring a complete rebuild. The disadvantage is the storage capacity required.

Note: The *transitive closure* table for the active content of the current version of the *International Release*, has about six million rows. The row count increases when *Extensions* are included. Typical database representations of the *transitive closure* table and associated indexes consume more than a Gigabyte of disk storage.

7.7.5.1.6 Recommendations

The *Transitive Closure* method is strongly recommended for use in any environment requiring high performance where disk capacity for storage and/or bandwidth for distribution are not a problem.

Where disk capacity and/or distribution bandwidth are limiting factors, Branch Numbering provides an efficient alternative approach.

7.7.5.2 Transitive closure implementation

Technology

The technology used to develop an *SNOMED CT enabled application* or used to query *SNOMED CT* data will affect the selection of the best implementation technique for the *transitive closure*.

If the *transitive closure* will be used to support SQL queries, a full *transitive closure* table needs to be created and stored *as a table in the relational database*.

In the cases where the transitive close will support actions in a software API, testing subsumption between in-memory objects, an *in-memory map* provides the best benefits.

7.7.5.2.1 Transitive closure distribution

It has been proposed that a *transitive closure* table should be released. This would support easier implementation and provide a reference against which to check alternative algorithms. The *transitive closure* table in a *full release* would contain a full history of the *transitive closure* since the first release of *SNOMED CT*. This would allow subsumption queries to be applied based on any release.

At present this table is not distributed and the format for such a distributed *transitive closure* table remains under discussion.

The following sub-sections provide basic advice on generating and using a simple and functional *transitive closure* table. Even if the *SNOMED CT International Edition transitive closure* is distributed, implementers may need to generate *transitive closures* including the content from one or more *Extensions*.

7.7.5.2.2 Transitive closure implementation in a relational database

7.7.5.2.2.1 Generating a transitive closure table

There are various ways in which a *transitive closure* table can be generated. The method illustrated here represents the smallest SQL *query* that might be used for this purpose. It may not be the most efficient *query* but on a typical Windows PC generates a snapshot *transitive closure* in about 5 minutes.

Table 240: MySQL script to Create a Snapshot Transitive Closure Table

```
-- --------------------------------------------------------------
-- SNOMED CT Transitive Closure for the Active Snapshot         --
-- Author: David Markwell 2010-2011                             --
-- Complete build of TC table for the most recent RFF2 version  --
-- Takes 5 minutes to run on typical system                     --
-- --------------------------------------------------------------

-- ---------------------------------------------------------------------------
-- ASSUMPTIONS
-- 1. Use of MySQL
-- 2. Database called `rf2` exists (or changed Initialize USE command)
-- 3. Database contains a table or view called `soa_relationship`
-- 4. The table `soa_relationship` contains an active snapshot (static or dynamic)
-- of the sct_relationship file(s) (including any extensions)
-- 5. The output table sct2_transitiveclosure contains the snapshot transitive
-- closure after completion.

-- Note: The soa_relationship view created by other sample scripts in this
--       document was used for testing this script.
-- ---------------------------------------------------------------------------
-- Initialize database connection
USE rf2;

-- Set delimiter to allow procedure creation
DELIMITER $$

-- Create procedure to make the TransitiveClosure
DROP PROCEDURE IF EXISTS `sct2_make_tc`$$

CREATE PROCEDURE `sct2_make_tc`()
BEGIN
-- Initialise by removing existing tables
 DROP TABLE IF EXISTS `sct2_transitiveclosure`;
 DROP TABLE IF EXISTS `tmp_tc1`;
 DROP TABLE IF EXISTS `batch_monitor`;

-- Create a table to allow batch process to be monitored (optional)
CREATE TABLE `batch_monitor` (
 `step` int(11) NOT NULL,
 `time` datetime DEFAULT NULL,
 `recs` int(11) DEFAULT NULL,
 `info` varchar(45) COLLATE latin1_general_cs DEFAULT NULL,
 PRIMARY KEY (`step`)
);

-- Set the snapshot version time
SET @effectiveTime=configTime();

-- Initialize step counter
SET @step=0;

-- Record progress in batch_monitor table
INSERT INTO `batch_monitor`
 (`step`,`time`,`recs`,`info`)
 VALUES(@step,NOW(),0,'start');

-- Create empty sct_transitive closure table
CREATE TABLE `sct2_transitiveclosure` (
 `subtypeId` BIGINT(20) NOT NULL ,
 `supertypeId` BIGINT(20) NOT NULL ,
 `effectiveTime` DATETIME,
 `active` BOOLEAN,
```

```
 PRIMARY KEY (`subtypeId`,`supertypeId`,`effectiveTime`),
 KEY `ix_tc_main` (`subtypeId`,`supertypeId`),
 KEY `ix_tc_inv` (`supertypeId`));

-- Create temporary first level transitive closure table
CREATE TEMPORARY TABLE `tmp_tc1` (
 `subtypeId` BIGINT(20) NOT NULL ,
 `supertypeId` BIGINT(20) NOT NULL ,
 PRIMARY KEY (`subtypeId`,`supertypeId`),
 KEY `ix_tc1` (`supertypeId`));

-- Insert Values into First Level TC
INSERT IGNORE INTO `tmp_tc1`(`supertypeId`, `subtypeId`)
 SELECT `destinationId`, `sourceId` FROM `soa_relationship`
  WHERE `active`=1 AND `typeId`= 116680003;

-- Create Level A temporary table for first iteration
DROP TABLE IF EXISTS `tmp_tcA`;

CREATE TEMPORARY TABLE `tmp_tcA` (
 `subtypeId` BIGINT(20) NOT NULL ,
 `supertypeId` BIGINT(20) NOT NULL ,
 PRIMARY KEY (`subtypeId`,`supertypeId`),
 KEY `ix_tc2` (`supertypeId`));

-- Copy Level 1 in to Level A for first iteration
INSERT IGNORE INTO `tmp_tcA`(`supertypeId`, `subtypeId`)
 SELECT `supertypeId`, `subtypeId` FROM `tmp_tc1`;

-- Start the Loop each pass adds 2 steps to the semantic distance
TcLoop:LOOP
  BEGIN
-- Increment the step count
    SET @step=@step+1;

-- Count records in Level A
    SET @rcount=(SELECT count(`supertypeId`) FROM `tmp_tcA`);

-- Batch monitor report (optional)
    INSERT INTO `batch_monitor`
        (`step`,`time`,`recs`,`info`)
        VALUES(@step,NOW(),@rcount,'tcA');

-- If Level A empty then quit here
    IF @rcount=0 THEN LEAVE TcLoop; END IF;

-- Append Level A records to final TC table
    INSERT IGNORE INTO `sct2_transitiveclosure`(`supertypeId`, `subtypeId`,`effectiveTime`,`active`)
    SELECT `supertypeId`, `subtypeId`,@effectiveTime,1 FROM `tmp_tcA`;

-- Create Level B temporary table for this iteration (adds 1 to semantic distance)
DROP TABLE IF EXISTS `tmp_tcB`;

CREATE TEMPORARY TABLE `tmp_tcB` (
 `subtypeId` BIGINT(20) NOT NULL ,
 `supertypeId` BIGINT(20) NOT NULL ,
 PRIMARY KEY (`subtypeId`,`supertypeId`),
 KEY `ix_tc3` (`supertypeId`));

-- Insert A+1 into B
INSERT IGNORE INTO `tmp_tcB` (`supertypeId`, `subtypeId`)
 SELECT `t`.`supertypeId`,`t1`.`subtypeId`
  FROM `tmp_tcA` `t` INNER JOIN `tmp_tc1` as `t1`
```

```
  ON `t`.`subtypeId`=`t1`.`supertypeId`
  LEFT OUTER JOIN `sct2_transitiveclosure` As `tc`
  ON `t`.`supertypeId`=`tc`.`supertypeId` AND `t1`.`subtypeId`=`tc`.`subtypeId`
  WHERE `tc`.`subtypeId` is null;

-- Level B empty then quit here
   SET @step=@step+1;

-- Level A empty then quit here
   SET @rcount=(SELECT count(`supertypeId`) FROM `tmp_tcB`);
   INSERT INTO `batch_monitor`
      (`step`,`time`,`recs`,`info`)
      VALUES(@step,NOW(),@rcount,'tcB');
   IF @rcount=0 THEN LEAVE TcLoop; END IF;

-- Append Level B to final TC table
INSERT IGNORE INTO `sct2_transitiveclosure`(`supertypeId`, `subtypeId`,`effectiveTime`,`active`)
   SELECT `supertypeId`, `subtypeId`,@effectiveTime,1 FROM `tmp_tcB`;

-- Create Level A temporary table for next iteration
DROP TABLE IF EXISTS `tmp_tcA`;
CREATE TEMPORARY TABLE `tmp_tcA` (
 `subtypeId` BIGINT(20) NOT NULL ,
 `supertypeId` BIGINT(20) NOT NULL ,
 PRIMARY KEY (`subtypeId`,`supertypeId`),
 KEY `ix_tc3` (`supertypeId`)
);

-- Insert B+1 into A
INSERT IGNORE INTO `tmp_tcA` (`supertypeId`, `subtypeId`)
 SELECT `t`.`supertypeId`,`t1`.`subtypeId`
  FROM `tmp_tcB` `t` INNER JOIN `tmp_tc1` AS `t1`
  ON `t`.`subtypeId`=`t1`.`supertypeId`
  LEFT OUTER JOIN `sct2_transitiveclosure` As `tc`
  ON `t`.`supertypeId`=`tc`.`supertypeId` AND `t1`.`subtypeId`=`tc`.`subtypeId`
  WHERE `tc`.`subtypeId` is null;

END;
END LOOP;

END$$
-- END OF PROCEDURE CREATION

-- RUN THE CREATED PROCEDURE
CALL `sct2_make_tc`()$$

-- Delete the procedure
DROP PROCEDURE IF EXISTS `sct2_make_tc`$$
```

7.7.5.2.2.2 Transitive closure table structure

The simplest form for a *transitive closure* table has two columns labeled "SubtypeId" and "SupertypeId". Each of these columns has a datatype that supports the *SNOMED CT Identifier* and is populated by *concept identifiers*.

This simple table requires one unique index "SubtypeId+SupertypeId" and a secondary non-unique index by "SupertypeId" to allow efficient reversed lookup.

Additional columns may be included to optimize some extended functionality. For example:

- A flag to indicate rows that represent links between a *concept* and its proximal *primitive* supertypes.

- If *inactive concepts* are included in the table, a flag to indicate the nature of any Historical Association traversed.
- A semantic distance count indicating the number of direct | is a | *relationship* between the *subtype* and supertype. Although such a number has not absolute meaning it may be useful as a relative measure of proximity.
- An *Identifier* of the *transitive closure* row. This may be of value for maintaining history of changes to *transitive closures* between releases.

7.7.5.2.2.3 Using the transitive closure table to check subsumption

The following SQL *queries* illustrate ways to use a *transitive closure* table to test subsumption.
The queries here use a MySQL database with the snapshot *transitive closure* table built as using the script documented in *generating a transitive closure table*. In practice , the SQL queries shown here will often be used as clauses in more complex *queries* allowing many candidates and predicates to be tested as a condition of retrieval in a single *query*.

```
SET @cptid=233604007;

SELECT `subtypeId`
FROM `sct2_transitiveclosure`
WHERE `supertypeId`=@cptid;
```

Figure 120: Return the Concept.id values of all subtype descendants of a specified concept

```
SET @cptid=233604007;

SELECT `supertypeId`
 FROM `sct2_transitiveclosure`
 WHERE `subtypeId`=@cptid;
```

Figure 121: Return the Concept.id values of all supertype ancestors of a specified concept

```
-- This illustration returns a text message indicating the semantic relationships between two concepts
-- Change the concept Id values here to test other concepts.
SET @cptidA=233604007;
SET @cptidB=422588002;

(SELECT CONCAT(CONVERT(@cptidB, CHAR), IF(count(`subtypeId`)
 ,' is ',' is NOT '),'a subtype of ',CONVERT(@cptidA, CHAR))
  FROM `sct2_transitiveclosure`
  WHERE `supertypeId`=@cptidA and `subtypeId`=@cptidB)
UNION
(SELECT CONCAT(CONVERT(@cptidA, CHAR), IF(count(`subtypeId`)
 ,' is ',' is NOT '),'a subtype of ',CONVERT(@cptidB, CHAR))
  FROM `sct2_transitiveclosure`
  WHERE `supertypeId`=@cptidB and `subtypeId`=@cptidA)
UNION
(SELECT CONCAT(CONVERT(@cptidA, CHAR), IF(@cptIdA=@cptidB,' is ',' is NOT ')
  ,'a the same as ',CONVERT(@cptidB, CHAR)))
;
```

Figure 122: Test whether concept is a subtype of another candidate concept

```
-- This query looks for concepts that:
--    a) are subtypes of a specified concept; and
--    b) contain a term with a string matching a specified pattern.
-- Note: This query requires that the sct2_description table has a FULLTEXT index on `term`.
--       The soa_description view is derived from that table.

SET @cptid=71388002;
SET @pattern='asthma';

SELECT `d`.`conceptId`,`d`.`term` FROM `soa_description` `d`
```

```
JOIN `sct2_transitiveclosure` `t` ON `d`.`conceptId`=`t`.`subtypeId`
 WHERE MATCH (`d`.`term`) AGAINST (@pattern) AND `t`.`supertypeId`=@cptid;
```

Figure 123: Matching terms for subtypes of a specified concept

7.7.5.2.3 Transitive closure implementation in memory

For real-time *subsumption tests*, an in-memory map performs better than a lookup on a persisted table in a relational database.

Map Structure:

- Map key: *Subtype concept Concept Identifier*
- Map value: Collection of direct parents *Concept Identifiers*

The high speed provided by in-memory structures allow us to have a simpler *transitive closure* representation, including only the direct parents of the *concept* (not all the ancestors, like in the relational database approach), and with only one appearance of the *subtype concept* in the map.

A recursive algorithm will check subsumption for any pair of candidate ids, navigating the map, looking for the parents of the *subtype* candidate and iteratively for all the parents of the parents, until it reaches the *root concept* (*concept* with an empty parents collection on the map value). If the parent candidate is found in any of the parents collections during the recursive map navigation, then iteration stops and the *subsumption test* returns true.

This approach provides a very compact representation, a full *transitive closure* map occupies around 12 megabytes,

The map creation process is straightforward, a single iteration of all *"Is a" Relationships* would retrieve all the necessary information for the map. In an editing environment the update of the map is also very simple, having only the direct parents represented in the map, changes in one *concept* affect only one value of the map. If the implementation uses a DL Classifier, the whole map should be updated after a classification run.

7.8 Supporting Selective Data Retrieval

This section addresses the types of *terminology service* that are required to enable effective use of the *SNOMED CT* hierarchies and definitions when retrieving data.

The actual process of data retrieval is a *record service*, rather than a *terminology service* because it involves interactions with a database containing instance data (e.g. an *electronic health record* or data warehouse repository). However, queries may use predicates that specify *subtypes* of particular *concepts* or specify *concepts* that have particular defining *Relationships*. In order to resolve these queries, the application will need to provide or use *Terminology services*, adapted to the implementation model in use. The use of either a local extension with a "Managed content addition" strategy, or *postcoordinated expressions* strategy with an *expressions* reference table, has an effect on the required set of *Terminology services*. (see *How to choose a SNOMED CT extension strategy*)

7.8.1 Creating queries

A *terminology server* should support the creation of queries that retrieve *SNOMED CT* encoded data by facilitating the generation of predicate statements.

For example, a *terminology server* may generate an SQL predicate list that includes the *Concept Identifiers* of all unique *subtype descendants* of a specified *Concept*. Some *constraints* on this functionality may be necessary as top-level or other general *Concepts* may generate extremely long lists of *descendant Concept Identifiers*.

7.8.2 Types of queries

There are different ways of representing a terminology query that can be sent to a *terminology server*:

- *Concept* and *Refset* references: lists of Ids of *concept* that can be retrieved from the server.
- Text based queries: text phrases that will be applied to *concept Descriptions* in order to retrieve results for the query.
- *Concept* definition queries: the query provides a *concept* definition, and the server returns all *concepts* that are subsumed by, or are equivalent to the definition.
- *Expression* retrieval: predicate terminology *expressions* that can be applied to all candidate *concepts* in the terminology to test for inclusion in the results, with filtering by hierarchy and attributes. These *expressions* are defined using a standard *expression* grammar that can be parsed and transformed in order to be evaluated against candidate *concepts*.
- Query languages: a query language combines any of the previous data retrieval techniques in the same syntax, including references to *concepts* by Ids, by text searches, by *refsets* or by hierarchy and attributes.

7.8.2.1 Concept and Refset references

The predicate for search for *Concepts* or *Refsets* includes references to one or more specific *concepts* by *Concept Identifier*, and additional instructions on how to retrieve related *concepts*.

Example use cases:

1. An application populates a combo box with a list of *concepts* pre-defined in a Simple Type *Refset*, identified by Id.
2. An HL7 message provides a reference to a *concept* by Id, the *preferred term* for the local implementation is retrieved based on the conceptId.
3. An application provides a tree view of the hierarchy, the *root concept* is referenced by Id and any level of children *concepts* are retrieved based on user selections.
4. A *concept* and all its *descendants* are retrieved in order to match with clinical records, where no *postcoordination* is used.

Example queries:

1. Retrieve the specified *Concept Identifier*
2. Retrieve direct *subtypes* (children) of the specified *Concept Identifier*
3. Retrieve *descendants* (children and they children recursively) of the specified *Concept Identifier*.
4. Retrieve members of the Simple Type *Refset* identified by the specified *Concept Identifier*

This kind of selective data retrieval can be easily implemented using SQL, *Concept Identifiers* are the primary key or foreign keys in all the necessary tables in the *SNOMED CT* model. Retrieval of *descendants* can be optimized as detailed in the *"Optimizing concept subsumption testing* chapter.

Implementations based primarily on *precoordinated* content will be able to support most of their use cases with this kind of query. *SNOMED CT* content is distributed with a pre-computed inferred view, that can be trusted to retrieve all related *concepts* by references to a *concept* in the terminology.

Implementations that have created local extensions, will require the availability of a *Descriptions* Logic classifier in order to periodically compute a new inferred view that will discover new *relationships* between *concepts* in the terminology.

In use cases where clinical data is recorded as *postcoordinated expressions*, or where *postcoordinated expressions* are used as the query, the techniques described in the *Expression retrieval section* should be applied.

7.8.2.2 Text based queries

Support for Text based queries is a fundamental component of the *terminology server*,

Example use cases:

1. A user enters a text string that is matched with *SNOMED CT* content in order to retrieve the most similar candidates
2. A mapping support tool finds closest matches in *SNOMED CT* for a local terminology or a classification, in order to provide lexical mapping suggestions to users.

This subject is discussed in the *User Interface Terminology Services section*.

7.8.2.3 Concept definition queries

Concept definition queries allow the user to submit a *concept* definition to the *Terminology Server*, and the server returns all the *concepts* that are either equivalent or subsumed by the input *concept* definition.

Example use cases: A Hospital gets an *concept* definition as part of a Decision Support Rule that is shared with another hospital. The hospital staff needs to find out what *concepts* in their terminology are equivalent or subsumed by that *concept* definition. They have implemented a Managed Content Additions (MCA) model, and their patient records include references to *precoordinated concepts* in their local terminology. They don't store *postcoordinated expressions expressions* in the clinical record.

Possible representation formats are covered in the *Representational Forms* chapter, and they include *SNOMED CT Release Formats*, OWL, KRSS, *SNOMED CT postcoordinated expressions* and others.

A *Description logic classifier* is used to classify the *concept* definition with the candidate terminology, being the official distribution of *SNOMED CT* or an extension. The classifier will output the list of equivalent *concepts*, and will create an updated set of inferred *Relationships* that allow the detection of all *descendants concepts*.

However, if the server needs to match a repository of *postcoordinated expressions* the *Expression retrieval* technique becomes the required approach, transforming the *concept* definition into an *expression*.

7.8.2.4 Expression retrieval and normal forms

A *terminology server* should support selective retrieval by facilitating testing of *expressions* against *query predicates*.

Using *expressions* is the most common approach for supporting *postcoordination*.

Example use cases:

1. Users define a *postcoordinated expression*, the server verifies if the *expression* is an exact match with an existing *precoordinated concept* or if it will be stored in a *postcoordinated expressions* repository as a new or existing *expression*.
2. For a epidemiological purposes a predicate *expression* is created as a query. The predicate *expression* is matched against candidate *expressions* and *concept* references stored in the clinical record, to retrieve all content equivalent or subsumed by the predicate *expression*.

To facilitate complete and accurate retrieval of *precoordinated* and *postcoordinated expressions* from clinical records or other resources it is necessary to compare an *expression* in a record with a *query predicate*. This comparison needs to determine if the candidate *expression* is subsumed by the predicate.

The same meaning can be represented in different *postcoordinated expressions* and to facilitate comparison *expressions* with the same meaning can be converted to a common *normal form*. This section describes the process of normalization and the approach to testing for subsumption between the resulting *normal form expressions*.

7.8.2.4.1 Candidate and predicate expressions

In a *subsumption test* there are two *expressions*, one of which is being tested for subsumption by the other. To distinguish these *expressions* the following definitions are used:

Candidate *expression* - An *expression* that is being tested to see if it is subsumed by another *expression*.

Predicate *expression* - An *expression* that is being tested to see if it subsumes another *expression*.

Table 241: Example Predicate and Candidate Expressions

Predicate	Candidate	Test result
| Fracture of femur |	| Fracture of neck of femur |	**True**
	| Fracture of bone |	**False**
| Fracture of bone |	| Fracture of femur |	**True**
	| Fracture of neck of femur |	**True**
| asthma | *(in patient)*	| FH: Asthma |	**False**
	| Severe asthma | *(in patient)*	**True**
| Family history of respiratory disease |	| FH: Asthma |	**True**

7.8.2.4.2 Expression parts

The figures in this section illustrate some terms used to describe different parts of an *expression* in the discussion of *normal forms*, the guidance on *transforming expressions to normal forms* and on *testing subsumption and equivalence between expressions*.

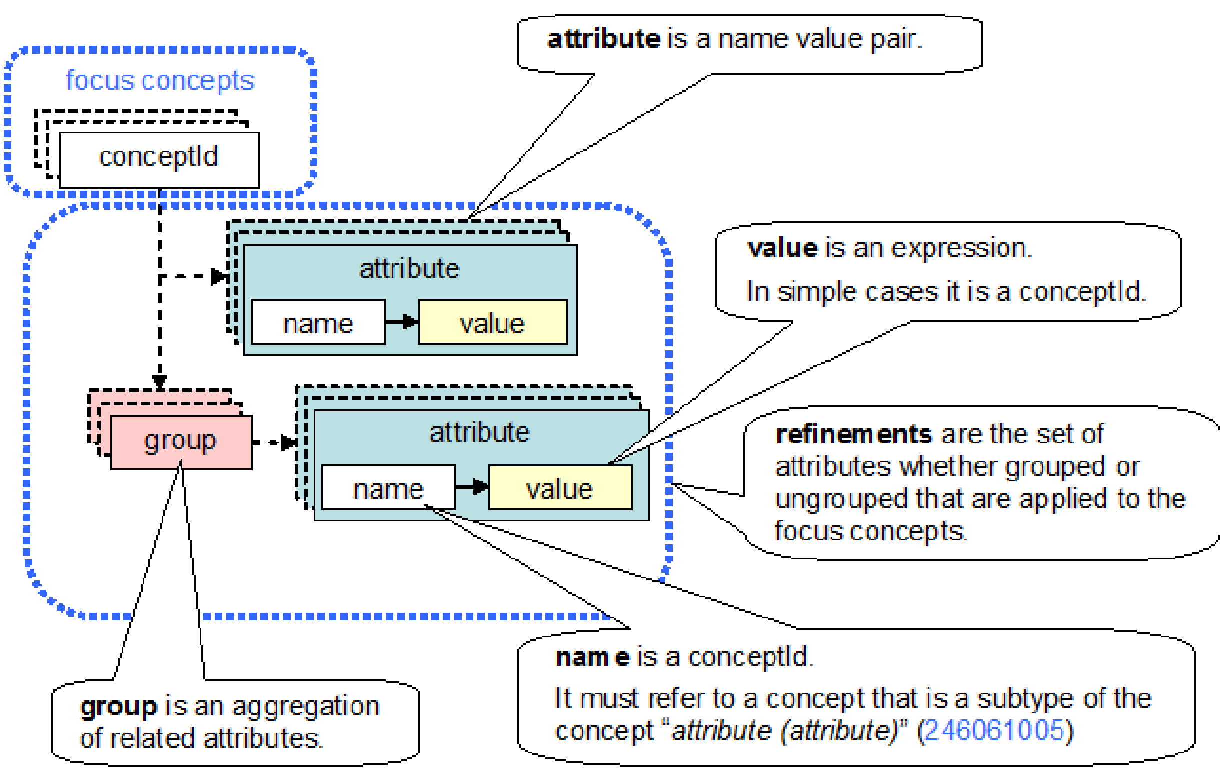


Figure 124: Focus concepts and refinements

As illustrated by *Figure 124*, an *expression* consists of one or more conceptIds plus optional *refinements*. The *refinements* may include any number of attributes. *Attributes* are expressed as name-value pairs and may apply independently or as part of a group.

The name part of the *attribute name* -value pair is a conceptId that refers to a *concept* that names the characteristic that is refined by this attribute. The value part of the *attribute name* -value pair is an *expression*. In simple cases, this is simply a conceptId referring to a *concept* that represents the appropriate value for this attribute. However, it may also be a nested *expression* as shown in *Figure 125*.

Figure 125 illustrates the potential for nesting of *expressions* and the naming conventions applied in this guide to distinguish different parts of an *expression* at different levels. The top level of an *expression* is referred to as the "focus *expression* ". It consists of a set of one or more "*focus concepts*" and a "focus *refinement* ". The values of the attributes in the focus *refinement* are "nested *expressions*" that consist of one or more "value *concepts*" optionally refined by a "nested *refinement* ".

Expressions may be nested recursively so there may be further levels of "nested *expressions*" with "nested *refinements* ". If it is necessary to distinguish the level of nesting, the following naming convention is applied.

Table 242: Expression Nesting

Level number	*Description*
level 0 *expression*	**Focus *expression***
level 1 *expression*	**Nested *expression***
level *N expression*	**An *expression* nested inside a level (*N* - 1) *expression***

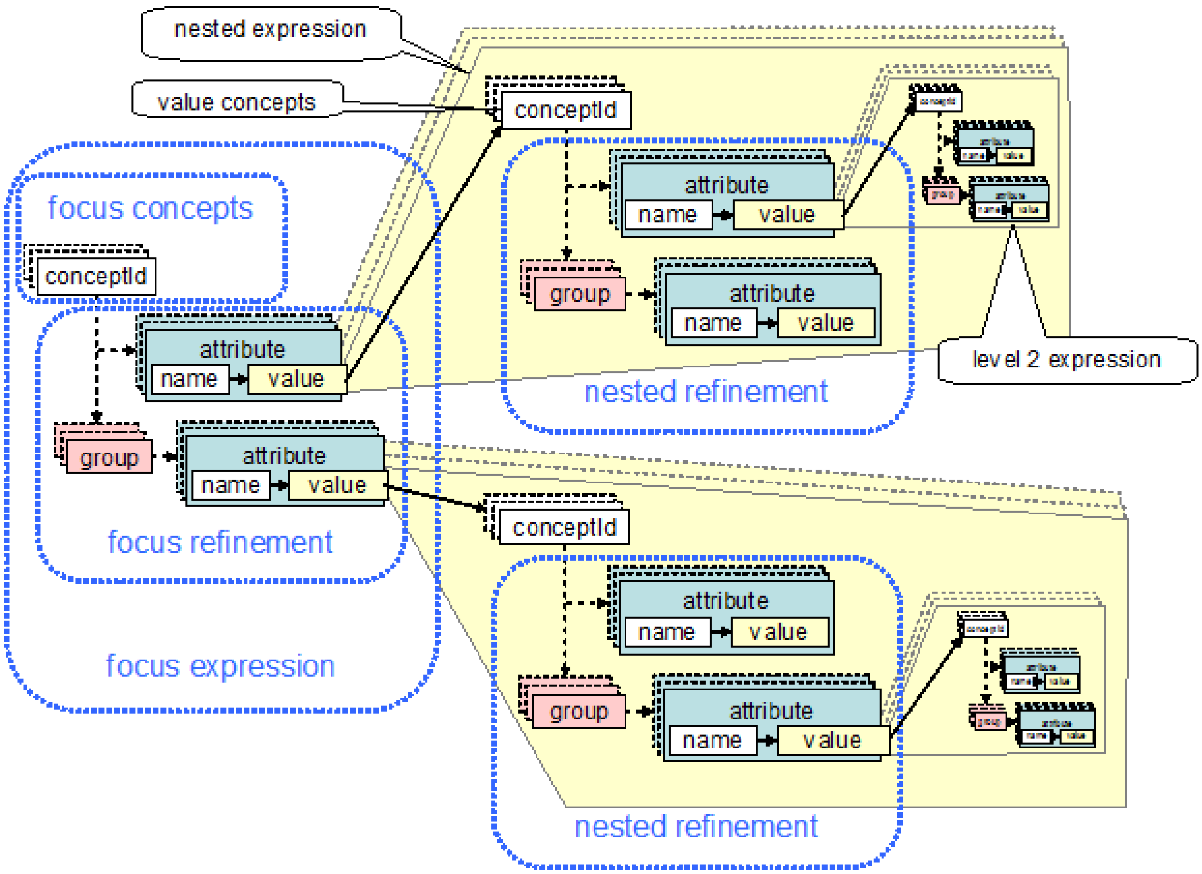


Figure 125: Illustration of the names used to refer to parts of a nested expression

The general pattern shown in *Figure 125* applies to all *expressions* whether or not they include *SNOMED CT* context information. *Figure 126* illustrates the specific features of an *expression* that includes a representation of *SNOMED CT* context.

The "focus *expression*" of a context containing *expression* is the "*context wrapper*" and may include a "context *refinement*" consisting of a set of context attributes:

- | associated finding | or | associated procedure | ;
- | finding context | or | procedure context | ;
- *408732007 | subject relationship context |* ;
- | temporal context | .

In a normalized context *expression*, all context attributes are grouped. Each group in a normalized *context wrapper* contains a complete set of four context attributes [25].

The value of the | associated finding | or | associated procedure | is a "nested *expression*" which is referred to as the "clinical kernel".

During some stages of processing, the "clinical kernel" is separated from the " *context wrapper*". When separated from its context the "clinical kernel" is the "focus *expression*" of a context-free *expression*.

[25] Usually a single group is present in a context expression. Theoretical cases exist for multiple groups where different contexts apply to different aspects of a concept but these cases are beyond the scope of the normalization rules in this guide.

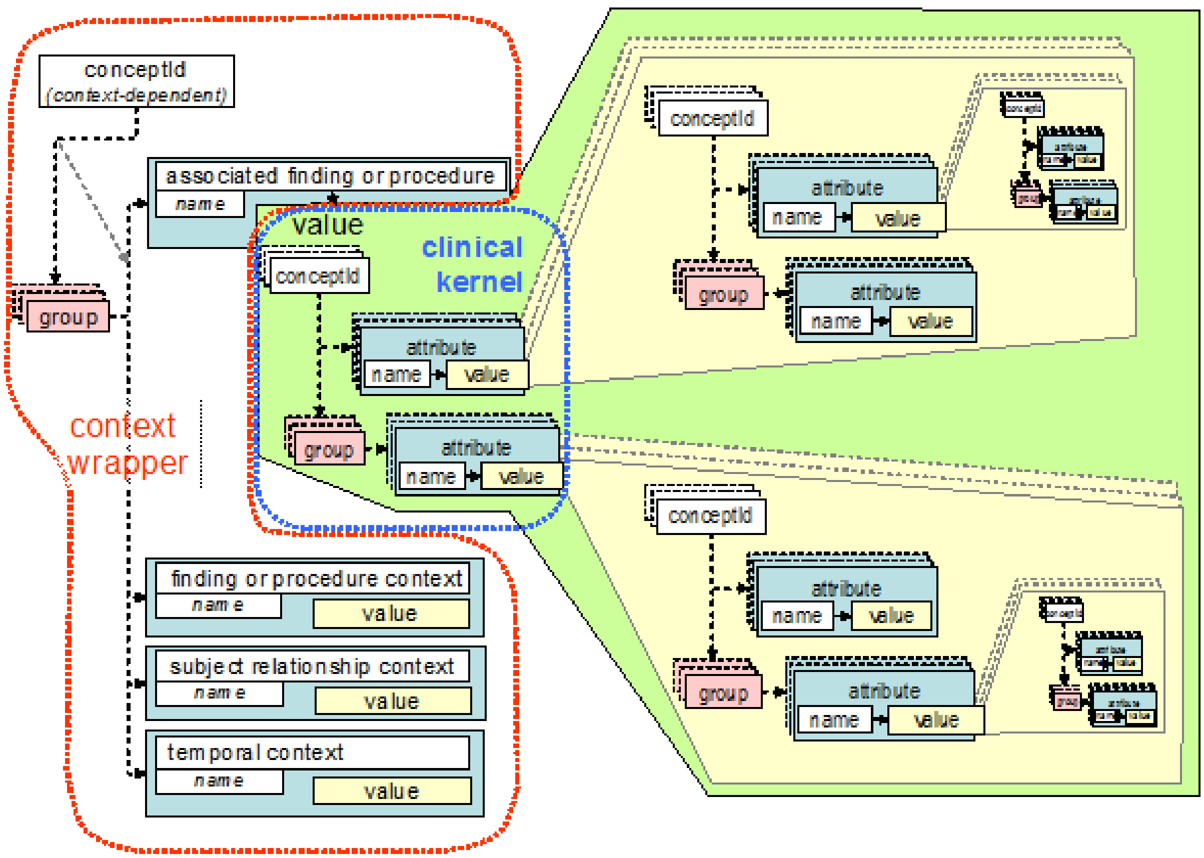


Figure 126: Illustration of the names used to refer to parts of an expression that represent context

7.8.2.4.3 Normal forms

A *normal form* is a view that can be generated for any valid *expression* by applying a set of logical *transformation* rules. Once converted to their *normal forms*, *expressions* can be more easily tested for subsumption by one another.

7.8.2.4.3.1 General characteristics of normal forms

All the conceptIds present in a *normal form expression* refer to *primitive concepts*. When normalizing an *expression*, every conceptId is replaced with the *normal form expression* that represents the definition of the referenced *concept*.

Normalization is recursive so that any element of a *concept* definition that refers to another *fully defined concept* is also replaced by the *normal form* of that *concept*.

One test of normalization is that applying the rules to an alreadynormalized *expression* should return an identical *expression*.

7.8.2.4.3.2 Rationale for long and short normal forms

There are two distinct *normal forms* that are of value when computing subsumption.

The long *normal form* is appropriate for a candidate *expression* because it explicitly states all the attributes can be inferred from *concepts* referenced by the *expression*. This makes it easier to test whether the candidate fulfills a set of predicate conditions.

The short *normal form* is more appropriate for predicate *expressions*. It enables more efficient retrieval testing because there are fewer conditions to test. However, there is no loss of specificity because any candidate that fulfills the conditions of the short *normal form* inevitably fulfills the conditions of the long *normal form*.

7.8.2.4.3.3 Building long and short normal forms

The most effective approach to building either *normal form* is to start by generating the long *normal form*. If the short *normal form* is required this can then be derived by removing redundant defining *relationships*.

Generating a long form to derive short form may appear counterintuitive. However, there are three reasons why this approach is strongly recommended.

- The process of generating either *normal form* includes steps that test subsumption between different parts of an *expression*. The long *normal form* is required as the predicate for these tests.
- A single approach requires only one algorithm to be specified and implemented. This eases maintenance and reduces the risks of inconsistencies developing between the two *transforms*.
- The short form is needed less frequently than the long form because it is used in predicates (e.g. queries) rather than in candidate instances (e.g. *expression* in a record). An approach that optimizes long form generation is therefore advantageous.

The next section sets out the general approach to generation of the long and short *normal forms* for a *concept* definition. References to individual *concepts* in the source *expression* are replaced by *normal form concept* definitions when generating the *normal form* of an *expression*.

7.8.2.4.3.4 Concept definitions in normal forms

7.8.2.4.3.4.1 Long Normal Form

A form which when applied to a candidate *expression* allows effective computation of whether it is subsumed by a predicate *expression*.

Supertype view: **Proximal *Primitive* Supertypes**

- For *fully defined concepts* compute the proximal *primitives*
- For *primitive concepts* treat the *concept* itself as the proximal *primitive* supertype:
  - Rationale: This *primitive concept* must be present to enable the candidate *expression* to be subsumed by a predicate *expression* that includes this particular *primitive concept*.

Attribute view: **All Defining *Relationships***

- For all *concepts* (whether *fully defined* or *primitive*) include all non - *subtype* defining *relationships*, irrespective of whether these are also present in the *union* of the definitions of the *primitive* supertypes:
  - Rationale: An *expression* may be subsumed by a *concept* that does not share all its proximal *primitive* supertypes. Some of the characteristics specified as part of other *primitives* in the candidate *expression* may also be present in the candidate *expression*.

7.8.2.4.3.4.2 Short Normal Form

A form which when applied to a predicate *expression* allows effective computation of whether a candidate *expression* is one of its *subtypes*.

Supertype view: **Proximal *Primitive* Supertypes**

- For *fully defined concepts* compute the proximal *primitives*
- For *primitive concepts* treat the *concept* itself as the proximal *primitive* supertype:
  - Rationale: As for long form see *Long Normal Form* on page 560.

Attribute view: **Differential Defining *Relationships*** (compared to supertype view)

- For *primitive concepts* there are no differential defining *relationships* because the *primitive concept* is its own proximal *primitive* supertype. Therefore in predicate *normal form* the attribute view is empty for *primitive concepts*.

- For *fully defined concepts* the differential form only includes defining *relationships*, and *relationship groups*, that are more specific than those present in the *union* of the definitions of the *primitive* supertypes:
  - Rationale: Each element in the predicate specifies an additional test to be applied to candidate *expressions*. However these additional tests are superfluous because:
    - The candidate *expression* cannot be subsumed by the predicate unless every candidate *primitive* supertype is subsumed by at least one predicate *primitive* supertype;
    - If this condition is met, then all defining *relationships* or *relationship groups* or the candidate *primitive* supertypes are inevitably also shared by the candidate *expression*.

7.8.2.4.3.4.3 Examples of normal form concept definitions

7.8.2.4.3.4.3.1 Normal form of a fully-defined concept with no intermediate primitives

The *concept* | fracture of femur | is *fully defined* and its proximal *primitive* supertype is a high-level *primitive* [26]. This proximal *primitive* does not share any of the defining *relationships* of the *concept* itself. Therefore, the long and short *normal forms* of | fracture of femur | are identical because all its defining *relationships* differ from those of its *primitive* supertype.

Table 243: Normal form of a fully-defined concept with no intermediate primitives

Concept	**71620000 | fracture of femur |**
Definition *(distributed)*	116680003 | is a | = 64572001 | disease | {116676008 | associated morphology | = 72704001 | fracture | ,363698007 | finding site | = 71341001 | bone structure of femur | }
Long NF *(candidate)*	64572001 | disease | : {116676008 | associated morphology | = 72704001 | fracture | ,363698007 | finding site | = 71341001 | bone structure of femur | }
Short NF *(predicate)*	64572001 | disease | : {116676008 | associated morphology | = 72704001 | fracture | ,363698007 | finding site | = 71341001 | bone structure of femur | }

Using the long *normal form*

To test if | fracture of femur | is subsumed by another *expression* the long *normal form* is used as the candidate. If all the conditions of a predicate *expression* are satisfied by this candidate then | fracture of femur | is subsumed by this predicate.

- The *concept* | fracture of femur | is subsumed by any *normal form* predicate *expression* with a *focus concept* | disease | (or a supertype of "diseases" such as | clinical finding |) unless the predicate *expression* also has conditions that do not subsume | morphology | = | fracture | and | finding site | = | bone structure of femur |.

Using the short *normal form*

To test if | fracture of femur | subsumes another *expression* the short *normal form* is used as the predicate. Any candidate *expression* that satisfies all the conditions of this candidate is subsumed by | fracture of femur |

[26] A high-level primitive is a concept that is primitive and has no fully defined supertypes.

- The *concept* | fracture of femur | subsumes any *concept* that is a | disease | with a morphology subsumed by "fracture" and a | finding site | subsumed by | bone structure of femur |:
  - The candidate *expression* | disease | with | morphology | = | fracture, open | and | finding site | = | structure of neck of femur | is thus subsumed.

7.8.2.4.3.4.3.2 Normal forms of a Primitive concept

The *concept* "asthma" is *primitive* so it is its own proximal *primitive* supertype. The long *normal form* therefore consists of the *concept* itself and all its defining *relationships*. The short *normal form* is simply the *concept* itself.

Table 244: Normal forms of a Primitive concept

Concept	**195967001 | asthma | [*Primitive*]**
Definition *(distributed)*	116680003 | is a | = 41427001 | disorder of bronchus | ,116680003 | is a | = 79688008 | respiratory obstruction | {116676008 | associated morphology | = 26036001 | obstruction | ,363698007 | finding site | = 955009 | bronchial structure | }
Long NF *(candidate)*	195967001 | asthma | : {116676008 | associated morphology | = 26036001 | obstruction | ,363698007 | finding site | = 955009 | bronchial structure | }
Short NF *(predicate)*	195967001 | asthma |

Using the long *normal form*

To test if "asthma" is subsumed by another *expression* the long *normal form* is used as the candidate. If all the conditions of a predicate *expression* are satisfied by this candidate then "asthma" is subsumed by this predicate.

- The *concept* "asthma" is subsumed by any *normal form* predicate *expression* with a *focus concept* that is "asthma" (or a supertype of "asthma" such as | disease | or | clinical finding |) unless the predicate *expression* also has conditions that do not subsume | morphology | = | obstruction | and | finding site | = | bronchial structure |.

Using the short *normal form*

To test if "asthma" subsumes another *expression* the short *normal form* is used as the predicate. Any candidate *expression* that satisfies all the conditions of this candidate is subsumed by | asthma |.

- The *concept* "asthma", only subsumes *expressions* that explicitly include a *focus concept* that is either "asthma" or a *subtype* of | asthma |:
  - The candidate *expression* | disease | with | morphology | = | obstruction | and | finding site | = | bronchial obstruction | is not subsumed by | asthma |.

7.8.2.4.3.4.3.3 Normal form of a fully-defined concept with an intermediate primitive

The *concept* | allergic asthma | is *fully defined* but its proximal *primitive* supertype ("asthma") is an intermediate *primitive*[27]. The long *normal form* consists of the proximal *primitive* supertype and all the defining *relationships* of | allergic asthma |. The short *normal form* is the same proximal *primitive* but the only *relationship* included is | due to | = | allergic reaction | as this is its only difference from the definition of the *primitive*.

Table 245: Normal form of a fully-defined concept with an intermediate primitive

<table>
<tr><th>Concept</th><th>389145006 | allergic asthma |</th></tr>
<tr><td>Definition
(distributed)</td><td>116680003 | is a | = 195967001 | asthma |
,116680003 | is a | = 418168000 | disorder due to allergic reaction |
,42752001 | due to | = 419076005 | allergic reaction |
{116676008 | associated morphology | = 26036001 | obstruction |
,363698007 | finding site | = 955009 | bronchial structure | }</td></tr>
<tr><td>Long NF
(candidate)</td><td>195967001 | asthma | :
42752001 | due to | = 419076005 | allergic reaction |
{116676008 | associated morphology | = 26036001 | obstruction |
,363698007 | finding site | = 955009 | bronchial structure | }</td></tr>
<tr><td>Short NF
(predicate)</td><td>195967001 | asthma | :
42752001 | due to | = 419076005 | allergic reaction |</td></tr>
</table>

Using the long *normal form*

To test if | allergic asthma | is subsumed by another *expression* the long *normal form* is used as the candidate. If all the conditions of a predicate *expression* are satisfied by this candidate then | allergic asthma | is subsumed by this predicate.

- The *concept* | allergic asthma | is subsumed by any *normal form* predicate *expression* with a *focus concept* that is "asthma" (or a supertype of "asthma" such as | disease | or | clinical finding |) unless the predicate *expression* also has conditions that do not subsume | morphology | = | obstruction | and | finding site | = | bronchial structure | and | due to | = | allergic reaction |

Using the short *normal form*

To test if | allergic asthma | subsumes another *expression* the short *normal form* is used as the predicate. Any candidate *expression* that satisfies all the conditions of this candidate is subsumed by | allergic asthma |.

- The *concept* | allergic asthma |, only subsumes *expressions* that explicitly include a *focus concept* that is either "asthma" or a *subtype* of "asthma" and the attribute | due to | = | allergic reaction |:
  - The candidate *expression* | disease | with | morphology | = | obstruction | and | finding site | = | bronchial obstruction | and | due to | = | allergic reaction | is <u>not</u> subsumed by | allergic asthma |.

[27] An intermediate primitive is a concept that is primitive but which has fully-defined supertypes and subtypes.

7.8.2.4.3.4.3.4 Normal form of a fully-defined concept with fully-defined attribute values

The *concept* | neoplasm of right lower lobe of lung | is *fully defined* with a high-level proximal *primitive* (| disease |). The long *normal form* consists of the proximal *primitive* supertype and all the defining *relationships* of | neoplasm of right lower lobe of lung |. However, the value of the | finding site | attribute (| structure of right lower lobe of lung |) is itself *fully defined*. Therefore, this value is also transformed to *normal form* (| structure of lower lobe of lung | with | laterality | = | right |).The short *normal form* is the same because all of the defining *relationships* differ from those of the proximal *primitive* supertype.

The standard *SNOMED CT distribution format* does not support explicit nesting of definitions. The *normal forms* described in this require nesting and this is supported by the *SNOMED CT expression* model. As a result, the *normal forms* shown here differ from the distributed *relationship file* and from the *canonical table*.

Table 246: Normal form of a fully-defined concept with fully-defined attribute values

Concept	**126716006 | neoplasm of right lower lobe of lung |**
Definition *(distributed)*	116680003 | is a | = 126713003 | neoplasm of lung | {116676008 | associated morphology | = 108369006 | neoplasm | ,363698007 | finding site | = 266005 | structure of right lower lobe of lung | }
Long NF *(candidate)*	64572001 | disease | : {116676008 | associated morphology | = 108369006 | neoplasm | ,363698007 | finding site | = (90572001 | structure of lower lobe of lung | : 272741003 | laterality | = 24028007 | right |)}
Short NF *(predicate)*	*Same as long form*

Using the long *normal form*

To test if | neoplasm of right lower lobe of lung | is subsumed by another *expression* the long *normal form* is used as the candidate. If all the conditions of a predicate *expression* are satisfied by this candidate then | neoplasm of right lower lobe of lung | is subsumed by this predicate.

- The *concept* | neoplasm of right lower lobe of lung | is subsumed by any *normal form* predicate *expression* with a *focus concept* that is | disease | (or a supertype of | disease | such as | clinical finding |) unless the predicate *expression* also has conditions that do not subsume | morphology | = | neoplasm | and | finding site | = | structure of lower lobe of the lung | with | laterality | = | right |.

Using the short *normal form*

To test if | neoplasm of right lower lobe of lung | subsumes another *expression* the short *normal form* is used as the predicate. Any candidate *expression* that satisfies all the conditions of this candidate is subsumed by | neoplasm of right lower lobe of lung |.

- The *concept* | neoplasm of right lower lobe of lung |, only subsumes *expressions* that have a | finding site | that is the "right lower lobe of the lung". However, because this site is normalized an *expression* that postcoordinates the laterality and the site will also be subsumed.

7.8.2.4.3.5 Applying normal forms to expressions

The previous section described the manner in which *normal form expression transformations* are applied to *concept* definitions. In this section this approach is extended to cover *expressions* which may contain *refinements* or qualifications of the released *concepts*.

7.8.2.4.3.5.1 Normal form of a simple expression

The simplest *expression* consists of a reference to a single *concept* (i.e. a single conceptId with no *refinements*). The *normal forms* for this are the same as those for the *concept* definition. *Table 247* illustrates this using one of the examples used in the previous section.

Table 247: A simple expression with no refinements

***Expression* view**	***Expression***
Close-to-user	71620000 | fracture of femur |
Normal-form (short or long)	64572001 | disease | : {116676008 | associated morphology | = 72704001 | fracture | ,363698007 | finding site | = 71341001 | bone structure of femur | }

7.8.2.4.3.5.2 Normal forms of expressions with refinements

If a *refinement* specifies a more specific (*subtype*) value for one of the defining *relationships* of the *focus concept*, the refined value simply replaces the value in the definition. The examples in *Table 248* and *Table 249* illustrate this for *refinements* to either site or the morphology.

Table 250 extends this example to include *refinements* to both the morphology and site. In all these examples while the *refinements* were not grouped in the close-to-user form, the *transformation* groups the site and morphology. This occurs because the refined values are replacing the defining values that were grouped.

Table 248: An expression with a refinement to finding site

***Expression* view**	***Expression***
Close-to-user	71620000 | fracture of femur | : 363698007 | finding site |29627003 | structure of neck of femur |
Normal-form (short or long)	64572001 | disease | : {116676008 | associated morphology | = 72704001 | fracture | ,363698007 | finding site | = 29627003 | structure of neck of femur | }

Table 249: An expression with a refinement to the nature of the morphology

***Expression* view**	***Expression***
Close-to-user	71620000 | fracture of femur | : 116676008 | associated morphology | = 134341006 | displaced fracture |

Expression **view**	***Expression***
Normal-form (short or long)	64572001 | disease | : {116676008 | associated morphology | =134341006 | displaced fracture | ,363698007 | finding site | = 71341001 | bone structure of femur | }

Table 250: An expression with a refinement to the morphology and site

Expression **view**	***Expression***
Close-to-user	71620000 | fracture of femur | : 116676008 | associated morphology | = 134341006 | displaced fracture | ,363698007 | finding site | = 71341001 | bone structure of femur |
Normal-form (short or long)	64572001 | disease | : {116676008 | associated morphology | =134341006 | displaced fracture | ,363698007 | finding site | = 71341001 | bone structure of femur | }

7.8.2.4.3.5.3 Normal forms of expressions with qualifiers

Table 251 shows the effect of applying a *qualifier Attribute* to a *concept*. As this is a *qualifier* it is not present in the definition and there is no indication whether this *qualifier* should be grouped with the site and morphology. Therefore, the *qualifier* remains ungrouped in the *normal form*.

Table 251: An expression with a qualifier applied to specify severity

Expression **view**	***Expression***
Close-to-user	71620000 | fracture of femur | : 246112005 | severity | = 24484000 | severe |
Normal-form (short or long)	64572001 | disease | : 246112005 | severity | = 24484000 | severe | {116676008 | associated morphology | = 72704001 | fracture | ,363698007 | finding site | = 71341001 | bone structure of femur | }

7.8.2.4.3.5.4 Normal forms of expressions with nested refinements

A *refinement* may be applied to a value in the *expression* rather than directly to the *focus concept*. Laterality is the most obvious example of a nested *refinement* and is used for all the illustrations in this section. However, nesting also occurs with other *expressions*, notably *expressions* that include explicit representations of context (see *Expressions that include context* and *Normal forms and the context model*).

Table 252 illustrates this by showing the effect of applying laterality to *refinement* to the finding site. The resulting *normal form* groups the finding site and its nested laterality *refinement*, with the morphology because this sub-*expression* is a valid *refinement* of the defined site.

Table 252: An expression with refinement of the laterality (nested with body structure)

***Expression* view**	***Expression***
Nested refinement of laterality	71620000 | fracture of femur | : 363698007 | finding site | = (71341001 | bone structure of femur | : 272741003 | laterality | = 7771000 | left |)
Normal-form (short or long)	64572001 | disease | : {116676008 | associated morphology | = 72704001 | fracture | ,363698007 | finding site | = (71341001 | bone structure of femur | : 272741003 | laterality | = 7771000 | left |)}

7.8.2.4.3.5.5 Normal forms representations of laterality

Table 252 used laterality as an illustration of nested refinement. However, lateralized findings or procedures may be represented in several different ways.

As shown in *Table 253* applying laterality as a nested *refinement* to finding that has defined site requires restatement of the finding site (even though this value is unchanged). This redundancy is removed when the *expression* is transformed to its *normal form*.

Table 253: Nested laterality refinement

***Expression* view**	***Expression***
Close-to-user	47933007 | foot pain | : 363698007 | finding site | = (56459004 | foot structure | : 272741003 | laterality | = 7771000 | left |)
Normal-form (short or long)	22253000 | pain | : 363698007 | finding site | = (56459004 | foot structure | : 272741003 | laterality | = 7771000 | left |)

Table 254 shows an alternative that is available for sites where there is a *concept* that is specific for lateralized body structure. In this example, the value | left foot | is a valid *refinement* of finding site because it is a *subtype* of | foot structure |.

The *concept* | left foot | is *fully defined* and includes the defining *relationship* | laterality | = | left |. Therefore, the *normal form* of this *expression* is identical to the nested laterality example.

Table 254: Alternative expression refinements representing lateralization

***Expression* view**	***Expression***
Lateralized body structure value	47933007 | foot pain | : 363698007 | finding site | = 22335008 | structure of left foot |

***Expression* view**	***Expression***
Normal-form (short or long)	22253000 | pain | : 363698007 | finding site | = (56459004 | foot structure | : 272741003 | laterality | = 7771000 | left |)

Table 255 illustrates an alternative that is available in a limited number of cases where a *concept* exists that precoordinates a finding with a lateralized finding site. The example shown here is artificial because the *concept* | pain in left foot | is not present in *SNOMED CT* and there are no plans to add such *concepts*. However, some *concepts* of this nature do exist and their definitions when transformed result in the same *normal form* as the *expression* shown in the earlier example.

Table 255: Laterality precoordinated in a finding

***Expression* view**	***Expression***
precoordinated expression	*<some-id>* | pain in left foot | :
Normal-form (short or long)	22253000 | pain | : 363698007 | finding site | = (56459004 | foot structure | : 272741003 | laterality | = 7771000 | left |)

Table 256 shows a close-to-user form in which laterality has been applied directly to a finding. For the purposes of computing *equivalence* and subsumption the *concept model* always treats laterality as applying to body structures rather than directly to findings or procedures. However, a simple *transform* rule allows a close-to-user *expression* consisting of a finding with a direct laterality *refinement* to be normalized . This normalization rule specifies that the laterality *refinement* is applied to all lateralizable sites in the normalized *expression*. The end result of this *transform* is exactly the same *normal form* as results from other approaches. The same approach can be used for procedures.

Table 256: Laterality applied directly to a finding

***Expression* view**	***Expression***
Close-to-user *expression*	47933007 | foot pain | : 272741003 | laterality | = 7771000 | left |
Normal-form (short or long)	22253000 | pain | : 363698007 | finding site | = (56459004 | foot structure | : 272741003 | laterality | = 7771000 | left |)

Conclusions on approaches to laterality

In principle, all four of the representations shown in *Table 257* are acceptable and all of them can be transformed to the same *normal form*. However, laterality is only *precoordinated* with a limited number of findings, procedures and body structures. Therefore, the only representations that provide comprehensive coverage are the direct form (3) and the nested *normal form* (4). Superficially the *normal form* seem most appropriate but based on a more detailed the direct close-to-user form (3) is recommended for recording, storage and communication.

Table 257: Alternative representations of laterality

***Expression* view**	***Expression***	**Applicability**
1. *precoordinated*	*<some-id>* | pain in left foot | :	Limited Only available is a *concept* exists to represent the finding or procedure at a specific lateralized site.
2. Lateralized body structure value	47933007 | foot pain | : 363698007 | finding site | = 22335008 | structure of left foot |	Limited Only if available if a *concept* exists to represent the lateralized body structure.
3. Laterality applied directly to finding	47933007 | foot pain | : 272741003 | laterality | = 7771000 | left |	Simple general purpose close-to-user form.
4. Normal-form (short or long)	22253000 | pain | : 363698007 | finding site | = (56459004 | foot structure | : 272741003 | laterality | = 7771000 | left |)	Common form derived by transformation.

The direct close-to-user form (3) has three significant advantages when compared to the *normal form*:

- Where multiple sites are involved (see *Table 261*) or where multiple separately grouped actions apply to the same site, this approach avoids the need to specify laterality separately for each site[28]. Routinely presenting users with a choice of which sites are to be lateralize is likely to hinder acceptance.
- The nested approach "locks-in" the site value(s) and groupings present in the definition at the time the *expression* is authored. If a future release enhances or corrects that definition, instances of the same *refinement* before and after a change will not compute as equivalent. However, if laterality is applied directly, the derived *normal forms* will be identical irrespective of when the *expression* was created.
- The resulting *expressions* are simpler and more compact and the *transform* rules mean no information is lost.

7.8.2.4.3.5.6 Normal forms of expressions including refinements of a Primitive concept

When the *focus concept* is *primitive* or has an intermediate *primitive* supertype the normal and close to user forms are less likely to be the same as one another.

Table 258 illustrates the effects of a *refinement* applied to *primitive concepts*. The same general rules apply but after the *transformation* process the same *primitive focus concept* remains. The short *normal form expression* is identical to the close-to-user form because the *refinement* represents the only difference between the long *normal form* and the definition of the *focus concept*.

[28] Only on very rare occasions will a single finding or procedure require separate lateralization of different sites in its definition. However, support for the direct approach does not preclude the nested approach if it is necessary to associate different laterality refinement with different structures.

Table 258: An expression that refines a Primitive concept

Expression **view**	***Expression***
Close-to-user	12529006 I expiratory crackles I : 363698007 I finding site I = 303549000 I entire lower lobe of lung I
Normal-form (long)	12529006 I expiratory crackles I : 363698007 I finding site I = 303549000 I entire lower lobe of lung I ,363714003 I interprets I = 78064003 I respiratory function I ,418775008 I finding method I = (315306007 I examination by method I : {260686004 I method I = 129436005 I auscultation - action I ,363704007 I procedure site I = 257728006 I anatomical concepts I})
Normal-form (short)	12529006 I expiratory crackles I : 363698007 I finding site I = 303549000 I entire lower lobe of lung I

Table 259 illustrates the effects of a *refinement* applied to a *concept* with an intermediate *primitive* supertype. The short *normal form* contains the "due to" *Attribute* because this differs between the *focus concept* (allergic asthma) and the *primitive* supertype (asthma) as when as the I causative agent I specified in the *refinement*.

Table 259: An expression that refines a concept with an intermediate primitive supertype

Expression **view**	***Expression***
Close-to-user	389145006 I allergic asthma I : 246075003 I causative agent I =260147004 I house dust mite I
Normal-form (long)	195967001 I asthma I : 246075003 I causative agent I = 260147004 I house dust mite I ,42752001 I due to I = 419076005 I allergic reaction I {116676008 I associated morphology I = 26036001 I obstruction I ,363698007 I finding site I = 955009 I bronchial structure I }
Normal-form (short)	195967001 I asthma I : 246075003 I causative agent I = 260147004 I house dust mite I ,42752001 I due to I = 419076005 I allergic reaction I

7.8.2.4.3.5.7 Normal forms for refinements of concepts with more complex definitions

Some *concept* definitions include multiple instances of the same defining attribute. Usually these are grouped separately, for example to represent a procedure that examines one body structure and removes another. When *refinements* are applied to these *concepts* a question arises as to which value is to be refined. In most cases, the *transform* rules allow this to be determined without requiring the close-to-user *expression* to explicitly state the instance that is being refined. The *transform* rule states that an ungrouped *refinement* applies to any instance of the appropriate attribute that subsumes it.

Table 260 shows the effect of this *transform* rule when the *refinement* of procedure site is a *subtype* of one of the defining site attributes but not of the other one. The appropriate value is refined and the other value is unchanged.

Table 260: An expression that refines one of two sites

***Expression* view**	***Expression***
Close-to-user	116028008 | salpingo-oophorectomy | : 363704007 | procedure site | = 280107002 | entire left fallopian tube |
Normal-form (short or long)	71388002 | procedure | : {260686004 | method | = 129304002 | excision - action | ,363704007 | procedure site | = (181463001 | entire fallopian tube | : 272741003 | laterality | = 7771000 | left |)} {260686004 | method | = 129304002 | excision - action | ,363704007 | procedure site | = 15497006 | ovarian structure | }

Table 261 shows a case in which a *refinement* of laterality is applicable to both the sites in the definition of a procedure (i.e. both | ovarian structure | and | fallopian tube structure | are lateralizable). Therefore, the resulting *normal form* shows this lateralization applied to both structures.

Table 261: An expression that lateralizes multiple sites

***Expression* view**	***Expression***
Close-to-user	116028008 | salpingo-oophorectomy | : 272741003 | laterality | = 7771000 | left |
Normal-form (short or long)	71388002 | procedure | : {260686004 | method | = 129304002 | excision - action | ,363704007 | procedure site | = (15497006 | ovarian structure | : 272741003 | laterality | = 7771000 | left |)} {260686004 | method | = 129304002 | excision - action | ,363704007 | procedure site | = (31435000 | fallopian tube structure | : 272741003 | laterality | = 7771000 | left |)}

In a few cases, a *refinement* that is valid for more than one attribute may need to be applied specifically to just one of those attributes. In these cases, the close to user form should include an *attribute group* with at least one other attribute from the appropriate group in the *concept* definition. This allows the distinction to be made by the *transform* rules for *attribute group* merging.

Otherwise, the close-to-user form should not repeat groups or additional attributes that are unchanged from the definition. These attributes and groups are derivable by the *normal form transformation*. However, if they are included in the stored or communicated close-to-user form they are "locked-in", which may impair *equivalence* testing across releases.

7.8.2.4.3.5.8 Normal forms and the context model

The *SNOMED CT* context model is designed to allow *equivalence* and subsumption testing to take account of difference in the context in which a finding or procedure *concept* is used. The same general *transform* rules apply to *concepts* that include explicit statements of context. However, in addition to these rules the default context to a finding or procedure *expression* that has no explicitly stated context. This additional step allows the *equivalence* and *subsumption tests* to be applied in exactly the same way to *expressions* without stated context and to those with a stated context.

Table 262 shows an *expression* in which the *focus concept* | family history of disorder | has a definition that includes stated context. The disorder | allergy to nuts | is stated as the | associated finding |. Both these *concepts* are transformed to their respective *normal forms*

- The *normal form* of | family history of disorder | is a context wrapped in which "person in the family" is the value of "subject *relationships* context";
- The *normal form* of | allergy to nuts | (106190000 | allergy | with | causative agent | = | nut |) becomes the nested value of the | associated finding | *Attribute* of the *context wrapper*.

Table 262: An expression that includes specific context information

***Expression* view**	***Expression***
Close-to-user	281666001 | family history of disorder | : 246090004 | associated finding | = 91934008 | allergy to nuts |
Normal-form (short or long)	243796009 | context-dependent category | : {246090004 | associated finding | = (106190000 | allergy | : 246075003 | causative agent | = 13577000 | nut |) ,408729009 | finding context | = 410515003 | known present | ,408731000 | temporal context | = 410512000 | current or specified | ,408732007 | subject relationship context | = 303071001 | person in the family | }

Table 263 shows an *expression* that does not state its context. Applying the *transform* rules the *normal form expression* is generated as in previous examples. When the additional step to apply the default context is carried out a default context-wrapper (| known present | in "the subject of the record" at "current or specified time |") is created and the clinical *expression* becomes the clinical-kernel within this wrapper.

In this case, the morphology and finding site *Attributes* are omitted as the values of these attributes are the same as those defined for the *Primitive concept* 195967001 | asthma |.

Table 263: Applying default context to an expression

<table>
<tr><th>Expression view</th><th>Expression</th></tr>
<tr><td>Close-to-user</td><td>195967001 | asthma | : 246112005 | severity | = 255604002 | mild |</td></tr>
<tr><td>Normal-form
with context
(long)</td><td>243796009 | situation with explicit context | :
{246090004 | associated finding | =
(195967001 | asthma | :
246112005 | severity | = 255604002 | mild |
{116676008 | associated morphology | = 26036001 | obstruction |
,363698007 | finding site | = 955009 | bronchial structure | })
,408729009 | finding context | = 410515003 | known present |
,408731000 | temporal context | = 410512000 | current or specified |
,408732007 | subject relationship context | = 410604004 | subject of record | }</td></tr>
<tr><td>Normal-form
with context
(short)</td><td>243796009 | situation with explicit context | :
{246090004 | associated finding | =
(195967001 | asthma | :
246112005 | severity | = 255604002 | mild |)
,408729009 | finding context | = 410515003 | known present |
,408731000 | temporal context | = 410512000 | current or specified |
,408732007 | subject relationship context | = 410604004 | subject of record | }</td></tr>
</table>

7.8.2.4.3.5.9 Normal forms that take account of the information model

When *expressions* are used in record systems or electronic communication there is often some surrounding contextual information that may affect the way in which the meaning of the *expression* should be interpreted. For example, a reference to a disease within a part of a record dedicated to "family history" should not be interpreted as a diagnosis of the patient. This is a complex area because many different information models and conventions may apply in different systems. However, some general rules have been identified and can be defined in relation to standard reference models (e.g. the *HL7* Version 3 *Reference Information Model*).

The general rules are that contextual information apparent in the surrounding information model should be separated from the *SNOMED CT expression* before it is normalized . The *expression* is then transformed to a *normal form expression*. If the resulting *normal form* contains a *context wrapper*, this is separated from the clinical-kernel. A new *context wrapper* is derived by merging the information model context and any context stated in any original *context wrapper*. The *SNOMED CT* default context is only applied to fill in the gaps where neither the information model nor the original wrapper provides a definitive value for a context *Attribute*. The clinical-kernel is then nested in the new *context wrapper*.

The examples in this section cover two of the most common areas in which context from the information model affects the meaning of a contained *expression* in a predictable and processable way.

Table 264 illustrates the fact that when an *expression* representing a procedure exists in an information construct that represents a request, the default | procedure context | value | done | is overridden by the information model. Thus the resulting *normal form expressions* show the | procedure context | value "requested".

Table 264: Information model representation of context affecting default context

***Expression* view**	***Expression***
Information model	<u>Request</u> • Represented by the information model for example *HL7* Observation.moodCode="RQO". 113075003 I creatinine measurement, serum I
Close-to-user *with context*	400999005 I procedure requested I : 363589002 I associated procedure I =113075003 I creatinine measurement, serum I
Normal-form *with context* (long)	243796009 I context-dependent category I : 363589002 I associated procedure I = (252144003 252144003 : 116686009 I has specimen I = (123038009 I specimen I : 370133003 I specimen substance I = 67922002 I serum I) ,246093002 I component I = 15373003 I creatinine I) ,408730004 I procedure context I = 385644000 I requested I ,408731000 I temporal context I = 410512000 I current or specified I ,408732007 I subject relationship context I = 410604004 Isubject of recordI
Normal-form *with context* (short)	243796009 I context-dependent category I : 363589002 I associated procedure I =113075003 I creatinine measurement, serum I 408730004 I procedure context I = 385644000 I requested I ,408731000 I temporal context I = 410512000 I current or specified I ,408732007 I subject relationship context I = 410604004 Isubject of recordI

Table 265 illustrates that when the information model applies a value to a measurement procedure the resulting statement expresses a finding (i.e. the finding of a specific value as a result of that procedure).

If the requirement is to distinguish between requested and completed procedures, the default procedure context-wrapper could be applied. This would (correctly) assert that the procedure had been done (I procedure context I = I done I).

However, if the requirement is to distinguish between goals and actual measured values the default finding context-wrapper is more appropriate. This would indicate that this was a finding that was known to be present (I finding context I = I known present I).

Table 265: Measurement procedures with values assigned in the information model

***Expression* view**	***Expression***
Information model	Report • Represented by the information model for example *HL7* Observation.moodCode="EVN". 113075003 | creatinine measurement, serum | **Value = 16 g/L** • Represented by the information model for example by *HL7* Observation.value.
Normal-form *with context* (short)	243796009 | situation with explicit context | : {246090004 | associated finding | = 113075003 | creatinine measurement, serum | ,408729009 | finding context | = 410515003 | known present | ,408731000 | temporal context | = 410512000 | current or specified | ,408732007 | subject relationship context | = 410604004 |subject of record| } **Value = 16 g/L** • Represented by the information model for example by *HL7* Observation.value.

7.8.2.4.4 Transforming expressions to normal forms

The process of transforming an *expression* to a *normal form* is based on the *description logic* definitions of the *concepts* referenced by the *expression*. Using this approach, *expressions* that are authored, stored and/or communicated in a relatively informal close-to-user form are logically transformed into a common normalized form. In this normalized form it is possible to apply simple rules to test subsumption between *expressions*.

The simplest case of a valid close-to-user *expression* is a single conceptId, and the approach described can be applied to these simple *precoordinated expressions*, as well as to more complex *expressions* that include multiple conceptIds and *refinements* (*qualifiers*).

The approach to normalization may be applied to specific *expressions* but may also be extended to take account of contextual information derived from the information model in which the *expression* is situated. Therefore, the *normal form* may include *SNOMED CT* context information, even if this is not present in the initial *SNOMED CT expression*.

The algorithm extends earlier work on *canonical forms* as follow:

- Normalizes *fully defined* values within definitions or *expressions* producing nested *expressions* that are fully normalized .
- Merges *refinements* stated in an *expression* with definitional *relationships* present in the definitions of the *concepts* referenced by the *expression*:
    - The merge process takes account of *refinements* that may not be grouped or nested in a manner that precisely reflects the structure of a current (or future) *concept* definition;
    - This avoids the need to add, store and communicate potentially spurious detail from current definitions to the *expression* recorded by a user or software application.
- Takes account of context rules including default context and a preliminary approach to moodCode mapping and handling of procedures with values (present in algorithm but not yet easily visible in test environment).
- Supports *subsumption tests* that take account of finding specified with | known absent | finding context.

*Figure 127*illustrates an overview of the process of normalization of an *expression*. Subsequent sections describe the processes shown in this diagram.

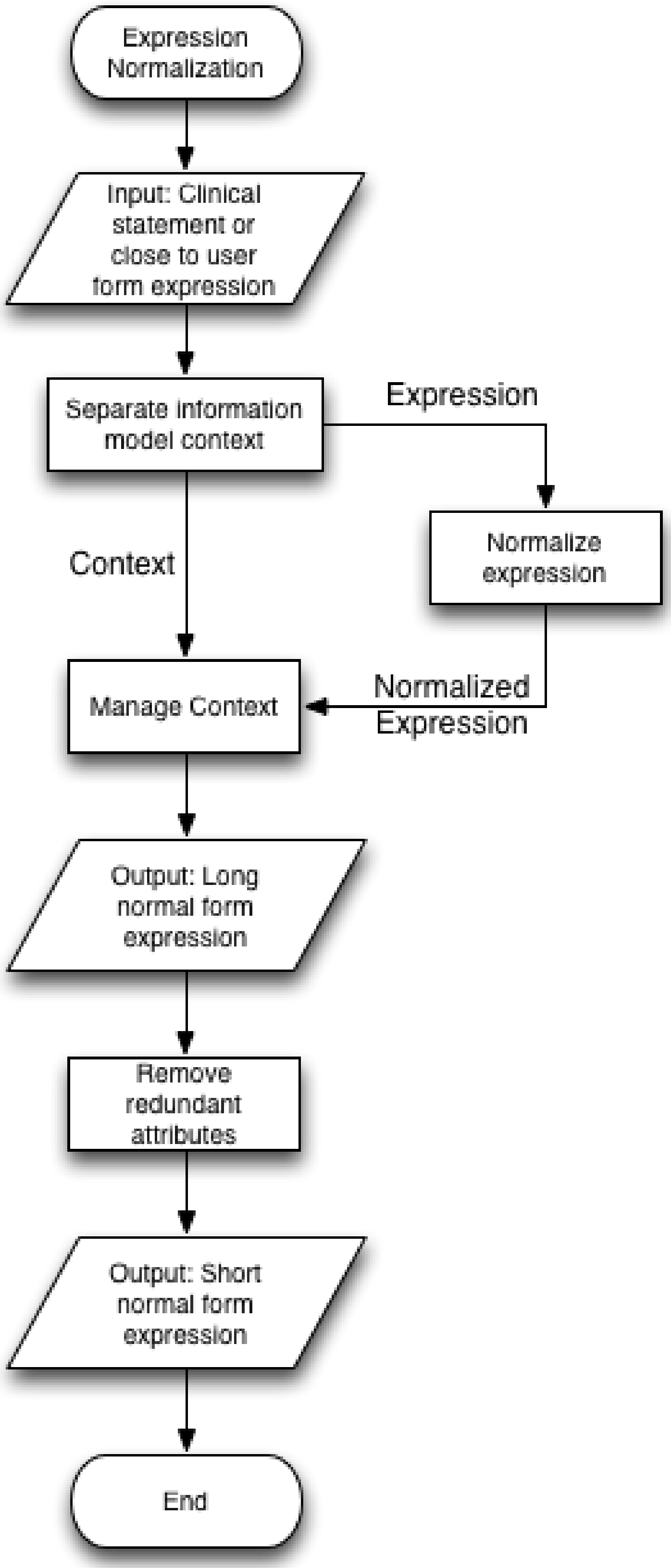


Figure 127: Overview of expression normalization process

7.8.2.4.4.1 Separate information model context

The objective of this process is to separate information associated with an *expression* from the *expression* itself.

Information that is not part of the *expression* itself, may influence its interpretation.

For example, a *expression* used in an *HL7* Observation in goal mood (moodCode="GOL") implies that the finding context I goal I applies to the *expression* rather than the default value I known present I.

If the input is a *expression* without any information about its use within a specific information model:

- The *expression* is passed unchanged to the "Normalize *expression*" process;
- No information model context is passed to the "Manage context" process.

If the input is an *HL7* clinical statement (or a similar structure that conveys additional contextual information):

- The *expression* is separated from the surrounding information model information and is passed to the "Normalize *expression*" process;
- Relevant surrounding information model information is passed to the "Manage context" process.

The items of surrounding information that are relevant vary according to the information model and the guidelines on its use. For example, if the *HL7* clinical statement model is used, any of the following attributes and related classes that are present are relevant to normalization of the *expression* in context:

- Act.moodCode;
- Observation.value;
- Act.negationInd;
- Act.uncertaintyCode;
- participation associations or an Act (especially "subject").

The way in which these attributes may affect *SNOMED CT* context is discussed in *Manage context*.

Normalize expression

Figure 128 illustrates the detailed steps in the process of normalizing an *expression*. This process takes place after separating the *expression* from any surrounding information model context and before managing context representation in the *expression*.

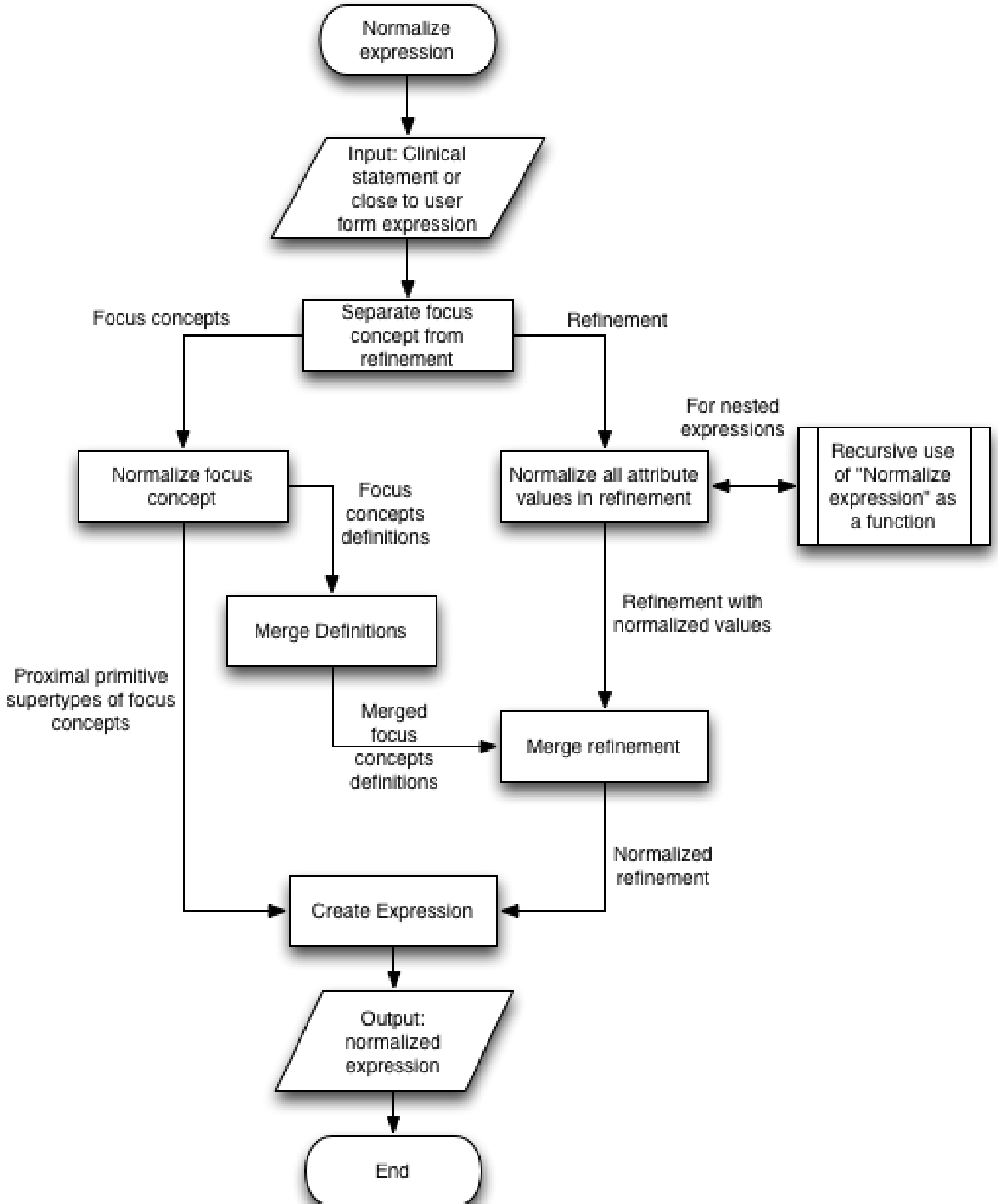


Figure 128: Expression normalization processes

7.8.2.4.4.2.1 Separate focus concepts from refinement

The set of *focus concepts* in the *expression* is passed to the *Normalize focus concepts* process.

If the *expression* contains a *refinement*, this is passed to the *Normalize attribute values in refinement* process.

Table 266: Separate focus concept from refinement

	Expression
Original *expression*	12676007 | fracture of radius |+ 397181002 | open fracture |: 272741003 | laterality |= 7771000 | left | 42752001 | due to |= 297186008 | motorcycle accident |
Focus Concept	12676007 | fracture of radius |+ 397181002 | open fracture |
Refinement	272741003 | laterality |= 7771000 | left | 42752001 | due to |= 297186008 | motorcycle accident |

Normalize attribute values in refinement

The value of every attribute specified in the *expression refinement* (including grouped and ungrouped attributes) is treated as an *expression* and normalized according to the full set of rules in *Normalize Normalise expression*. To ensure depth-first processing, this recursive process is carried out before any other processing of the *expression refinement*.

Recursive normalization should be applied to all values even if they are represented by single conceptIds.

When all *attribute values* in the *expression refinement* have been processed, the *refinement* is passed to the *Merge refinement* process.

Normalize focus concepts

The set of *focus concepts* is normalized to generate two separate outputs described in the following sections.

7.8.2.4.4.2.3.1 The set of normalized definitions of each focus concept

The set of normalized definitions includes a separate normalized definition for each *focus concept*,

- The normalized definition includes:
  - All ungrouped *relationships*
  - All *relationship groups* complete with contained *relationships*
- All *relationship* values are normalized by recursively following the full set of rules described in *Concept definitions in normal forms*.

  Note: Storage of pre-computed normalized form of *concept* definitions simplifies this process as it removes the requirement for recursive processing of definitions at run time.

The set of normalized definitions is passed to the *Merge definitions* process.

Table 267: Normalize focus concepts definitions

	Expression
Original *expression*	12676007 | fracture of radius |+ 397181002 | open fracture |: 272741003 | laterality |= 7771000 | left | 42752001 | due to |= 297186008 | motorcycle accident |

	Expression
Focus Concepts	12676007 | fracture of radius |+ 397181002 | open fracture |
Set of normalized *focus concept* definitions	76069003 | disorder of bone |:{ 116676008 | associated morphology |= 72704001 | fracture |, 363698007 | finding site |= 23416004 | bone structure of radius |}+ 64572001 | disease |: 116676008 | associated morphology |= 52329006 | open fracture |

7.8.2.4.4.2.3.2 The non-redundant proximal primitive supertypes of the focus concepts

The non-redundant proximal *primitive* supertypes of the *focus concepts* is the set of all *primitive* supertypes of all the *focus concepts* with redundant *concepts* removed.

- A *concept* is redundant if it is:
  - A duplicate of another member of the set;
  - A supertype of another *concept* in the set.

The set of proximal *primitive* supertypes generated by this process is passed to the *Create expression* process as the *focus concepts* for the output *expression*.

Table 268: Normalize focus concepts definitions

	Expression
Original *expression*	12676007 | fracture of radius |+ 397181002 | open fracture |: 272741003 | laterality |= 7771000 | left | 42752001 | due to |= 297186008 | motorcycle accident |
Focus Concepts	12676007 | fracture of radius |+ 397181002 | open fracture |
Set of normalized *focus concept* definitions	76069003 | disorder of bone |:{ 116676008 | associated morphology |= 72704001 | fracture |, 363698007 | finding site |= 23416004 | bone structure of radius |}+ 64572001 | disease |: 116676008 | associated morphology |= 52329006 | open fracture |
List of all proximal *primitive* supertypes	76069003 | disorder of bone | 64572001 | disease |
List of non-redundant proximal *primitive* supertypes	64572001 | disease |

7.8.2.4.4.2.4 Merge definitions

The set of normalized definitions derived from the *Normalize focus concepts* process are merged with one another to remove redundancy. Then the normalized *refinement* is merged with the pre-merged definition to create a single *refinement* which expresses the full set of definitions and *refinements* without unnecessary redundancy.

The rules applied to the merger are described below for grouped and ungrouped attributes.

Group merging is completed before applying any ungrouped *relationships*. This ensures that, where appropriate, ungrouped attributes are applied to the correct groups in the output.

Redundant attributes are not removed until the merger process is complete. This ensures that the full set of attributes is available to allow matching throughout the process of merging.

7.8.2.4.4.2.4.1 Attribute names and attribute hierarchies

The following sections on merging groups and attributes refer to "name-matched" attributes. Two or more attributes in a definition or *expression* are "name-matched" if they have the same *attribute name*[29].

- For example, the attribute | procedure site | = | appendix structure | is name-matched by the attribute | procedure site | = | entire femur |.

However, consideration also needs to be given to hierarchical *relationships* between different "*attribute names*". For example, | procedure site - direct | and | procedure site - indirect | are *subtypes* of | procedure site |.

The simplest approach that can be consistently applied is to treat attributes that have subsumed names as name-matched for the purposes of group and value merging. The more specific *attribute name* is then applied to the merged attribute in the target definition. This means that the same rules apply for merging the values of | procedure site | and | procedure site - direct | as apply to mergers of attributes with identical names and that the name | procedure site - direct | would then be applied to any values that were merged in this way.

<u>Progress note</u>

Review of a number of practical examples suggests that there may be some unexpected consequences of this approach. For this reason, while the issues that arise are studied further, implementers are recommended only to merge literal name-matched attributes.

<u>Some potential issues are noted here</u>

As definitions are refined over time there will be more use of the specific | procedure site - indirect | and | procedure site - direct |. Should pre-existing *refinements* to the more general | procedure site | be assigned to whichever of the more specific attributes has a value that subsumes the refined value?

If this rule is applied to some combined procedures then the merger collapses some existing definitions that contain both a | procedure site | and a | procedure site - direct | so that only one of these attributes remains. This will become less of an issue as | procedure site - indirect | is applied more widely.

7.8.2.4.4.2.4.2 Merging groups

- If a group in one definition meets the following criteria in relation to a group in the other definition then the groups are merged:
  - At least one *attribute* in one of the groups is name-matched by an *attribute* in the other group.

and

[29] The words "attribute" and " attribute name " are used here as documented in the SNOMED CT guide to the "Abstract Logical Models and Representation Forms". In SNOMED CT distribution files a "defining relationship " is equivalent to this use of the word "attribute" and a " relationship type" represents an "attribute name".

- For each name-matched pair of *attributes*, the value of that *attribute* in one group either subsumes or is identical to the value of the name-matched *attribute* in the other group;
- Groups that meet the criteria for merging are merged by adding all *attributes* present in both source groups to the same group in the merged target definition;
- Groups that cannot be merged are created as separate groups in the target definition.

Note that these conditions allow additional *attributes* that are not name-matched to be present in either of the candidate groups. They also allow values of name-matched *attributes* to be subsumed in different directions between the two groups (i.e. do not require the entire of one group to be subsumed the other group).

Table 269: Merging groups examples

Groups to merge	Result
Group 1 363704007 | procedure site |= 421235005 | structure of femur |, 363700003 | direct morphology |= 72704001 | fracture | *Group 2* 260686004 | method |= 129371009 | fixation - action |, 424226004 | using device |= 31031000 | Orthopedic internal fixation system, device |	**No match, no merge** *Group 1* 363704007 | procedure site |= 421235005 | structure of femur |, 363700003 | direct morphology |= 72704001 | fracture | *Group 2* 260686004 | method |= 129371009 | fixation - action |, 424226004 | using device |= 31031000 | Orthopedic internal fixation system, device |
Group 1 363698007 | finding site |= 62413002 | radius |, 116676008 | associated morphology |= 72704001 | fracture | *Group 2* 363698007 | finding site |= 87342007 | fibula |, 116676008 | associated morphology |= 72704001 | fracture |	***Attribute name* match, but values don't match and are not subsumed in any direction ('radius' and 'fibula'), no merge** *Group 1* 363698007 | finding site |= 62413002 | radius |, 116676008 | associated morphology |= 72704001 | fracture | *Group 2* 363698007 | finding site |= 87342007 | fibula |, 116676008 | associated morphology |= 72704001 | fracture |

Groups to merge	Result
Group 1 363698007 | finding site |= 62413002 | radius |, 116676008 | associated morphology |= 72704001 | fracture | *Group 2* 363698007 | finding site |= 87342007 | distal radius |, 116676008 | associated morphology |= 72704001 | fracture |	***Attribute name* match, 'distal radius' is subsumed by 'radius', merged groups.** *Group 1* 363698007 | finding site |= 62413002 | radius |, 116676008 | associated morphology |= 72704001 | fracture | 363698007 | finding site |= 87342007 | distal radius |, 116676008 | associated morphology |= 72704001 | fracture |

* Redundant elements will be removed in later in the process

7.8.2.4.4.2.4.3 Merging ungrouped attributes

- If an ungrouped attribute in one definition is name-matched by a grouped attribute in the other definition, this attribute is merged according to the following rules:
  - If the value of the ungrouped attribute subsumes the value of the name-matched grouped attribute:
    - omit the ungrouped attribute from the target definition.
  - If the value of the grouped attribute subsumes the value of the name-matched grouped attribute:
    - add the ungrouped attribute to the group containing the matching grouped attribute in the target definition.
    - if this condition is met by multiple groups:
      - add the ungrouped attribute to all groups that meet this condition.
  - If the value of the name-matched grouped and ungrouped attributes are disjoint:
    - add the ungrouped attribute as an ungrouped attribute in the target *expression*.
- If an ungrouped attribute is name-matched with an ungrouped attribute in the other definition this attribute is merged according to the following rules:
  - If the value of one of the name-matched attributes subsumes the other value:
    - include the attribute with the most specific value (not grouped);
    - omit the attributed with the less specific value.
  - If the value of the name-matched attributes are identical:
    - Include one and omit the other.
  - If neither of the of the two preceding conditions apply:
    - include both attributes (not grouped).
- If an attribute is ungrouped in one *expression* and there is no name-matched attribute in the other definition:
  - include the attribute (not grouped).

Table 270: Merging ungrouped attributes

	Result
Group 1 363698007 I finding site I= 62413002 I radius I, 116676008 I associated morphology I= 72704001 I fracture I *Group 2* 363698007 I finding site I= 87342007 I distal radius I, 116676008 I associated morphology I= 72704001 I fracture I *Ungrouped* 42752001 I due to I= 297186008 I motorcycle accident I	***Attribute name* match, 'distal radius' is subsumed by 'radius', merged groups. Ungrouped attribute does not match. Not merged.** *Group 1* 363698007 I finding site I= 62413002 I radius I, 116676008 I associated morphology I= 72704001 I fracture I 363698007 I finding site I= 87342007 I distal radius I, 116676008 I associated morphology I= 72704001 I fracture I *Ungrouped* 42752001 I due to I= 297186008 I motorcycle accident I
Group 1 363698007 I finding site I= 62413002 I radius I, 116676008 I associated morphology I= 72704001 I fracture I *Group 2* 363698007 I finding site I= 87342007 I distal radius I, 116676008 I associated morphology I= 72704001 I fracture I *Ungrouped* 116676008 I associated morphology I= 72704001 I fracture I	***Attribute name* match, 'distal radius' is subsumed by 'radius', merged groups. Ungrouped attribute matches name and value. Merged.** *Group 1* 363698007 I finding site I= 62413002 I radius I, 116676008 I associated morphology I= 72704001 I fracture I 363698007 I finding site I= 87342007 I distal radius I, 116676008 I associated morphology I= 72704001 I fracture I 116676008 I associated morphology I= 72704001 I fracture I

* Redundant elements will be removed in later in the process

7.8.2.4.4.2.4.4 Remove redundant elements from the merged definition

Check each group in the target definition and, within that group, compare the values of any name-matched attributes.

- If an attribute in the group has a value that subsumes the value of another name-matched attribute in the same group, remove that attribute from this group in the target definition.

Check the ungrouped set of attributes.

- If any ungrouped attribute has a value that subsumes the value of a name-matched attribute, remove this ungrouped attribute from the target definition.

Note

The removal of redundancies described only applies to name-matched pairs of attributes. It does *not* affect attributes that are redundant *only* because they are present in the definitions of the *primitive focus concepts*. Supertype (| is a |) *relationships* are ignored during this stage of processing.

Table 271: Removing redundant elements

Merged definitions with redundancy	Redundancy removed
Group 1 363698007 | finding site |= 62413002 | radius |, 116676008 | associated morphology |= 72704001 | fracture | 363698007 | finding site |= 87342007 | distal radius |, 116676008 | associated morphology |= 72704001 | fracture | 116676008 | associated morphology |= 72704001 | fracture |	*Group 1* 363698007 | finding site |= 87342007 | distal radius |, 116676008 | associated morphology |= 72704001 | fracture |

7.8.2.4.4.2.4.5 Completion of the definition merging

Once the *focus concept* definitions have been merged, the target definition is passed to the *Merge refinement* process.

7.8.2.4.4.2.5 Merge refinement

The normalized *expression refinement* from the *Normalize attribute values in refinement* process is merged with the combined definition from the *Merge definitions* process. The rules for this process are the same as those for merging definitions.

Normalization of laterality

If an attribute representing a value for 272741003 | laterality | is present in the *refinement* and is applied to a *focus concept* that is not subsumed by 123037004 | body structure |, the laterality attribute should be applied to any and every lateralizable | body structure | specified in the resulting *refinement*.

Table 272: Normalization of laterality

	Expression
Original *expression*	12676007 | fracture of radius |: 272741003 | laterality |= 7771000 | left |
Normalized *expression* before laterality normalization	76069003 | disorder of bone |:{ 116676008 | associated morphology |= 72704001 | fracture |, 363698007 | finding site |= 23416004 | bone structure of radius |}, 272741003 | laterality |= 7771000 | left |

	Expression
Normalized *expression* after laterality normalization	76069003 | disorder of bone |:{ 116676008 | associated morphology |= 72704001 | fracture |, 363698007 | finding site |=(23416004 | bone structure of radius |, 272741003 | laterality |= 7771000 | left |)}

Normalization of non-context attributes applied in a context wrapper

If the *focus concept* is subsumed by 243796009 | situation with explicit context | and any attributes other than valid context attributes [30] are present in the *refinement*, these attributes are applied as additional *refinement* of the value of the 246090004 | associated finding | or 363589002 | associated procedure | attribute.

7.8.2.4.4.2.5.3 Completion of the definition merging

Once the *refinement* has been merged the resulting final *refinement* is passed to the *Create expression* process.

7.8.2.4.4.2.6 Create expression

The create *expression* process combines the proximal *primitive* supertypes from the *Normalize focus concepts* process (as the new *focus concepts*) - with the *refinement* derived from the *Merge refinement* process.

The resulting *expression* is now fully normalized but context information may need to be adjusted or applied by the *Manage context* process.

Table 273: Normalize focus concepts definitions

	Expression
List of non-redundant proximal *primitive* supertypes	64572001 | disease |
Normalized refinement without redundancy	*Group 1* 363698007 | finding site |= 87342007 | distal radius |, 116676008 | associated morphology |= 72704001 | fracture |
Resulting normalized *expression*	64572001 | disease |:{ 363698007 | finding site |= 87342007 | distal radius |, 116676008 | associated morphology |= 72704001 | fracture |}

7.8.2.4.4.3 Manage context

Figure 129 illustrates the steps involved in managing context information extracted from the input statement or *expression*.

30 The only valid context attributes are: 246090004 | associated finding |, 363589002 | associated procedure |, 2470590016 | finding context |, 2470591017 | procedure context |, 2470592012 | temporal context | and 2470593019 | subject relationship context |.

The input to this process consists of the information model context, derived from the *Separate information model context* process, and the normalized *expression*, generated by the *Create Expression* step at the end of the *Normalize Normalise expression* process.

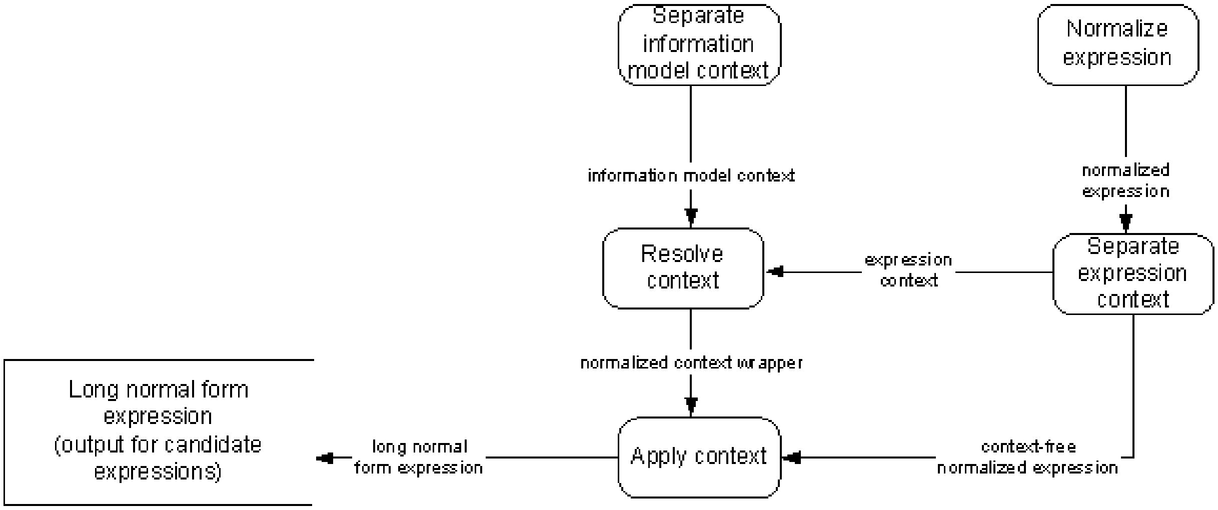


Figure 129: Managing context in normalized expressions

7.8.2.4.4.3.1 Separate expression context

The normalized *expression* (generated by the *Normalise expression* process) may or may not contain any context information. If it does, this context is separated from the *expression* so that it can be validated and reconciled with any information model context.

If the *focus concept* is *subtype* of 243796009 | situation with explicit context |

- The *expression* that represents the value of the | associated finding | or | associated procedure | attribute is passed as the context-free *expression* to the *Apply context* process;
- The focus *expression* without the | associated finding | or | associated procedure | attribute is passed to the *Resolve context*.

If the *focus concept* is not a *subtype* of 243796009 | situation with explicit context | but its *refinement* contains values for one or more of the following context attributes:2470590016 | finding context |, 2470591017 | procedure context |, 2470592012 | temporal context | *or* 2470593019 | subject relationship context |.

- These attributes are passed to the *Resolve context*:
    - If the attributes present do not include a | finding context | or | procedure context | value, then an indication of the top level supertype of the *focus concept* is also passed with these context attributes.
- The focus *expression*, with the context attributes removed, is passed as the context-free *expression* to the *Apply context* process.

If neither of the above conditions apply then

- An indication of the top level supertype of the *focus concept* is passed to the *Resolve context* process;
- The entire *expression* is passed as the context-free *expression* to the *Apply context* process.

7.8.2.4.4.3.2 Resolve context

The resolve context process takes the information model context derived from the *Separate information model context* process and the *expression* context derived from *Separate expression context* process and attempts to resolve them to generate a single consistent context.

The context information in the *expression* or information model may unequivocally indicate that:

- Finding context applies:

    - *Subtypes* of "finding" or "linkage *concept*";
    - *Subtypes* of | procedure | or "observable" with an associated "value" in the information model;
    - Finding context *attribute value* present in the *expression* context information.

- Procedure context applies:

    - *Subtypes* of | procedure | or "observable" without an associated "value" in the information model;
    - Procedure context *attribute value* present in the *expression* context information.

The appropriate default context applies unless modified

- By specific context attributes in the *expression* context information.
- By rules associated with particular information model context information:

    - For example, rules that in a reference file such as the MoodMap.xml (see *Figure 130*).

The output is one of the following

- A single *context wrapper* that is passed to the *Apply context* process.
- An indication that context is not relevant to the *expression* and should not be applied. This is also passed to the *Apply context* process allowing it to return a context-free *expression*.
- A report of errors arising from incompatibilities in the context information from the two sources.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- MoodMap.xml by David Markwell - Version 2005-03-26 -->
<!-- Copyright 2005 The Clinical Information Consultancy Ltd (www.clininfo.co.uk)
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

     http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
 -->

<!-- Suggested maps from HL7 Version 3 Mood Codes to default values for SNOMED Context attribute -->
<moodMap>
 <moodCode code="-" term="Any">
  <context id="408729009" term="finding context" defaultId="410515003" defaultTerm="known present"
permitted="410514004" permittedTerm="finding context value"/>
  <context id="408730004" term="procedure context" defaultId="385658003" defaultTerm="done"
permitted="288532009" permittedTerm="context values for actions"/>
  <context id="408731000" term="temporal context" defaultId="410512000" defaultTerm="current or specified"
 permitted="410510008" permittedTerm="temporal context value"/>
  <context id="408732007" term="subject relationship context" defaultId="410604004" defaultTerm="subject
 of record" permitted="125676002" permittedTerm="person - add other permitted values in future"/>
 </moodCode>
 <moodCode code="EVN" term="Event">
  <context id="408729009" term="finding context" defaultId="410515003" defaultTerm="known present"
permitted="410514004" permittedTerm="finding context value"/>
  <context id="408730004" term="procedure context" defaultId="385658003" defaultTerm="done"
permitted="410523001" permittedTerm="post-starting action status"/>
  <context id="408731000" term="temporal context" defaultId="410512000" defaultTerm="current or specified"
 permitted="410510008" permittedTerm="temporal context value"/>
  <context id="408732007" term="subject relationship context" defaultId="410604004" defaultTerm="subject
 of record" permitted="125676002" permittedTerm="person - add other permitted values in future"/>
 </moodCode>
 <moodCode code="GOL" term="Goal">
```

```
  <context id="408729009" term="finding context" defaultId="410518001" defaultTerm="goal"
permitted="410518001" permittedTerm="goal"/>
  <context id="408731000" term="temporal context" defaultId="410512000" defaultTerm="current or specified"
 permitted="410510008" permittedTerm="temporal context value"/>
  <context id="408732007" term="subject relationship context" defaultId="410604004" defaultTerm="subject
 of record" permitted="125676002" permittedTerm="person - add other permitted values in future"/>
 </moodCode>
 <moodCode code="INT" term="Intent">
  <context id="408730004" term="procedure context" defaultId="410522006" defaultTerm="pre-starting action
 status" permitted="410522006" permittedTerm="pre-starting action status"/>
  <context id="408731000" term="temporal context" defaultId="410512000" defaultTerm="current or specified"
 permitted="410510008" permittedTerm="temporal context value"/>
  <context id="408732007" term="subject relationship context" defaultId="410604004" defaultTerm="subject
 of record" permitted="125676002" permittedTerm="person - add other permitted values in future"/>
 </moodCode>
 <moodCode code="RQO" term="Request">
  <context id="408730004" term="procedure context" defaultId="385644000" defaultTerm="requested"
permitted="385644000" permittedTerm="requested"/>
  <context id="408731000" term="temporal context" defaultId="410512000" defaultTerm="current or specified"
 permitted="410510008" permittedTerm="temporal context value"/>
  <context id="408732007" term="subject relationship context" defaultId="410604004" defaultTerm="subject
 of record" permitted="125676002" permittedTerm="person - add other permitted values in future"/>
 </moodCode>
 <moodCode code="PRP" term="Proposal">
  <context id="408730004" term="procedure context" defaultId="385643006" defaultTerm="to be done"
permitted="385649005 385643006" permittedTerm="being organised / to be done"/>
  <context id="408731000" term="temporal context" defaultId="410512000" defaultTerm="current or specified"
 permitted="410510008" permittedTerm="temporal context value"/>
  <context id="408732007" term="subject relationship context" defaultId="410604004" defaultTerm="subject
 of record" permitted="125676002" permittedTerm="person - add other permitted values in future"/>
 </moodCode>
 <moodCode code="PRMS" term="Promise">
  <context id="408730004" term="procedure context" defaultId="385645004" defaultTerm="accepted"
permitted="385649005 385645004" permittedTerm="being organised / accepted"/>
  <context id="408731000" term="temporal context" defaultId="410512000" defaultTerm="current or specified"
 permitted="410510008" permittedTerm="temporal context value"/>
  <context id="408732007" term="subject relationship context" defaultId="410604004" defaultTerm="subject
 of record" permitted="125676002" permittedTerm="person - add other permitted values in future"/>
 </moodCode>
 <moodCode code="ARQ" term="Appointment request">
  <context id="408730004" term="procedure context" defaultId="385644000" defaultTerm="requested"
permitted="385644000" permittedTerm="requested"/>
  <context id="408731000" term="temporal context" defaultId="410512000" defaultTerm="current or specified"
 permitted="410510008" permittedTerm="temporal context value"/>
  <context id="408732007" term="subject relationship context" defaultId="410604004" defaultTerm="subject
 of record" permitted="125676002" permittedTerm="person - add other permitted values in future"/>
 </moodCode>
 <moodCode code="APT" term="Appointment">
  <context id="408730004" term="procedure context" defaultId="60304008" defaultTerm="scheduled"
permitted="60304008" permittedTerm="scheduled"/>
  <context id="408731000" term="temporal context" defaultId="410512000" defaultTerm="current or specified"
 permitted="410510008" permittedTerm="temporal context value"/>
  <context id="408732007" term="subject relationship context" defaultId="410604004" defaultTerm="subject
 of record" permitted="125676002" permittedTerm="person - add other permitted values in future"/>
 </moodCode>
 <moodCode code="DEF" term="Definition" notApplicable="true"/>
 <moodCode code="SLOT" term="Resource slot" notApplicable="true"/>
 <moodCode code="EVN.CRT" term="Event criterion" notApplicable="true"/>
 <moodCode code="OPT" term="Option" notApplicable="true"/>
</moodMap>
```

Figure 130: MoodMap.xml example

7.8.2.4.4.3.3 Apply context

If no *context wrapper* is provided by the *Resolve context* process, the context-free *expression* from the *Separate expression context* process is returned as the fully normalized *expression*.

If the *Resolve context* process provides a *context wrapper* including a 2470590016 | finding context |attribute, the context-free *expression* from the *Separate expression context* process is applied to this as the value of the 246090004 | associated finding |attribute. The resulting context-dependent *expression* is returned as the fully normalized *expression*.

If the *Resolve context* process provides a *context wrapper* including a 2470591017 | procedure context | attribute, the context-free *expression* from the *Separate expression context* process is applied to this as the value of the 363589002 | associated procedure | attribute. The resulting context-dependent *expression* is returned as the fully normalized *expression*.

moodMap

moodCode (13)

	code	term	context		id	term	defaultId	defaultTerm	permitted	permittedTerm
1	-	Any	context (4)							
				1	408729009	finding context	410515003	known present	410514004	finding context value
				2	408730004	procedure context	385658003	done	288532009	context values for actions
				3	408731000	temporal context	410512000	current or specified	410510008	temporal context value
				4	408732007	subject relationship context	410604004	subject of record	125676002	person - add other permitted values in future
2	EVN	Event	context (4)							
				1	408729009	finding context	410515003	known present	410514004	finding context value
				2	408730004	procedure context	385658003	done	410523001	post-starting action status
				3	408731000	temporal context	410512000	current or specified	410510008	temporal context value
				4	408732007	subject relationship context	410604004	subject of record	125676002	person - add other permitted values in future
3	GOL	Goal	context (3)							
				1	408729009	finding context	410518001	goal	410518001	goal
				2	408731000	temporal context	410512000	current or specified	410510008	temporal context value
				3	408732007	subject relationship context	410604004	subject of record	125676002	person - add other permitted values in future
4	INT	Intent	context (3)							
				1	408730004	procedure context	410522006	pre-starting action status	410522006	pre-starting action status
				2	408731000	temporal context	410512000	current or specified	410510008	temporal context value
				3	408732007	subject relationship context	410604004	subject of record	125676002	person - add other permitted values in future
5	RQO	Request	context (3)							
				1	408730004	procedure context	385644000	requested	385644000	requested
				2	408731000	temporal context	410512000	current or specified	410510008	temporal context value
				3	408732007	subject relationship context	410604004	subject of record	125676002	person - add other permitted values in future
6	PRP	Proposal	context (3)							
				1	408730004	procedure context	385643006	to be done	385649005 385643006	being organised / to be done
				2	408731000	temporal context	410512000	current or specified	410510008	temporal context value
				3	408732007	subject relationship context	410604004	subject of record	125676002	person - add other permitted values in future
7	PRMS	Promise	context (3)							
				1	408730004	procedure context	385645004	accepted	385649005 385645004	being organised / accepted
				2	408731000	temporal context	410512000	current or specified	410510008	temporal context value
				3	408732007	subject relationship context	410604004	subject of record	125676002	person - add other permitted values in future
8	ARQ	Appointment request	context (3)							
				1	408730004	procedure context	385644000	requested	385644000	requested
				2	408731000	temporal context	410512000	current or specified	410510008	temporal context value
				3	408732007	subject relationship context	410604004	subject of record	125676002	person - add other permitted values in future
9	APT	Appointment	context (3)							
				1	408730004	procedure context	60304008	scheduled	60304008	scheduled
				2	408731000	temporal context	410512000	current or specified	410510008	temporal context value
				3	408732007	subject relationship context	410604004	subject of record	125676002	person - add other permitted values in future
10	DEF	Definition								
11	SLOT	Resource slot								
12	EVN.CRT	Event criterion								
13	OPT	Option								

Figure 131: MoodMap.xml file image

7.8.2.4.4.4 Additional steps for alternative forms

The processes described in the preceding sections generate the "long normal view". This is the most general *normal form*. It can be directly applied to meet key requirements related to subsumption testing. It can also be used as the source from which to derive other useful forms that are optimized for particular purposes. The following sections outline some of these.

7.8.2.4.4.4.1 Deriving the short normal view

The short *normal form* can be derived from the long *normal form* by the following steps.

7.8.2.4.4.4.1.1 Generate the set of normalized definitions of each primitive focus concepts

A normalized *expression* includes the normalized definition for *concept* in the original *expression*.

- The normalized definition includes. In the long *normal form* this process is extended to include the definitions of the *Primitive concepts* in the normalized *expression*.

- The values of all defining *relationship* are expanded and normalized by recursively following the full set of rules described in *Normalize focus concepts* on page 580.

Note: Storage of pre-computed normalized forms of *concept* definitions simplifies this process as it removes the requirement for recursive processing of definitions at run time.

7.8.2.4.4.4.1.2 Merge the generated definition sets

This process follows exactly the steps described in *Merge definitions*.

7.8.2.4.4.4.1.3 Removed redundant attributes and groups

Attributes and groups shared with the merged definition are removed from the *refinement*. Only groups and ungrouped attributes that are identical can be removed from the *refinement*. If a group is not identical the parts that are similar cannot be removed.

7.8.2.4.4.4.1.4 Recursive removal of redundancy

The process described in this section is recursively applied to any nested *expressions* that remain after the top-level process to remove redundant attributes and groups.

Unlike the process of normalization , this process is done breadth first at each level in the *hierarchy*. If long normalized forms at nested levels are shortened before checking for redundancy, the *expression* will not match those in the merged definition even if they are semantically identical.

7.8.2.4.4.4.2 Canonical representations

The idea of a canonical representation is that it generates a predictable *string* rendering. The missing element to deliver this in the *description* of the "long *normal form*", is a specified sort *order* within the collections elements in an *expression*. A standard sort *order* is not essential for general purpose use but it is very useful to enable fast matching of logically identical *expressions* (which might otherwise be obscured by differences in order that have no semantic relevance).

The *canonical form* for an *expression* is regarded as being the long *normal form* ordered according to the following sorting rules.

- The *expression* is rendered in the form specified by the *SNOMED CT compositional grammar*. For canonical representation a restricted version of the *compositional grammar* is used:
  - No whitespace characters may be included in the *canonical form*
  - No pipe characters "|" and thus no *term* text shall be included in the *canonical form*.
  - Thus the only permitted characters are:
    - Digits [0-9] - for conceptId values;
    - Plus [+] - to combine *focus concepts*;
    - Colon [:] - to represent the start of a *refinement*;
    - Equals [=] - to link an *attribute name* to it value;
    - Comma [,] - to separate attributes within a *refinement*
    - Round brackets [()] - to represent nesting;
    - Curly brackets [{}] - to represent grouping.
- The syntax determines the general *order* of elements within an *expression* as follows:
  - Focus conceptIds;
  - *Attributes* (expressed as name-value pairs);
  - Groups (containing attributes).
- Within a set of focus conceptIds:
  - *Concept Identifiers* are sorted alphabetically based on their normal *string* rendering (i.e. digits with no leading zeros):

    - The reason for alphabetic sorting rather than numeric sorting is that it is complex to sort attributes and groups which consist of an arbitrary number of conceptIds using numeric keys.

- Within a set of ungrouped attributes or a set of attributes within a group:
    - *Attributes* are sorted alphabetically based on the *string* concatenation of the name and value conceptIds separated by an "=" sign;
    - If a value contains nested *refinements*, the value is enclosed in round brackets (which may influence the sort *order*) and the elements of the nested *expression* are sorted by applying the general canonical sorting rules.
- Within a set of groups:
    - Groups are sorted by alphabetical *order* of the combined set of previously sorted attributes.

7.8.2.4.5 Testing subsumption and equivalence between expressions

The main reason for generating *normal form expressions* is to enable testing for *equivalence* and subsumption between different post coordinated *expressions*. This section describes how these processes are carried out.

The process of generating *normal form* for an *expression* also requires testing of subsumption between subsidiary elements within the *expression*.

7.8.2.4.5.1 Testing for equivalence

The following steps can be applied to test for *equivalence* between any two valid *expressions*.

1. *Transform* both *expression* to long *normal form* (see *Transforming expressions to normal forms*).
2. Render these *normal forms* according the canonical representation (see *Canonical representations*.
3. Perform a simple *string* comparison between the two long *normal forms* in canonical representation:
    a. If the *strings* are identical then the *expressions* being tested are equivalent;
    b. If the *strings* are not identical the two *expressions* being tested are not logically equivalent.

.

Note that this does not prove that the *expressions* are not equivalent. This limitation applies for the following reasons:

- One or more of the *concepts* referenced by the original *expressions* may be *primitive* (i.e. not *fully defined*) and this may obscure the *equivalence* [31].
- Two different *expressions* may include an alternative "sufficient set" of *attributes* that imply the same meaning (see *Nature of the definition*).

7.8.2.4.5.2 Testing expression subsumption

The following steps can be applied to test for subsumption of any *candidate expression* by a *predicate expression*.

1. *Transform* the predicate *expression* to short *normal form* [32](see *Deriving the short normal view* on page 591):
    - The resulting "predicate short *normal form expression*" is referred in subsequent steps as the *normalized-predicate*.

[31] This issue will gradually diminish in significance as more concepts are fully defined through addition of new defining relationships.

[32] The *predicate long normal form* can be used instead of the *predicate short normal form* . However, the short form is preferred as it reduces the number of steps required in testing each candidate expression.

2. *Transform* the candidate *expression* to long *normal form* (see *Transforming expressions to normal forms*) :
   - The resulting "candidate long *normal form expression*" is referred in subsequent steps as the *normalized -candidate*.
3. Test for subsumption between the *normalized -predicate* and the *normalized -candidate* by applying the tests described in *Testing subsumption between two normal form expressions* on page 594:
   - The *predicate expression* subsumes the *candidate expression* if the *normalized -predicate* subsumes the *normalized -candidate*.

7.8.2.4.5.2.1 Testing subsumption between two normal form expressions

The following steps are applied to test if a *normalized -predicate* subsumes a *normalized -candidate*. This assumes that these *normal form expressions* have been generated as outlined in *Transforming expressions to normal forms*.

1. Test that each *focus concept* referenced in the *normalized -predicate* subsumes at least one *focus concept* in the *normalized -candidate*.:
   - If not, the *normalized -predicate* does not subsume the *normalized -candidate*. No further testing is required:
     - Exit with result *false*.
   - The approach to testing *concept* subsumption is described in section *Testing concept subsumption* on page 596.
2. Test that each *attribute group* in the *normalized -predicate* subsumes at least one *attribute group* in the *normalized -candidate*:
   - If not, the *normalized -predicate* does not subsume the *normalized -candidate*. No further testing is required:
     - Exit with result *false*.
   - The approach to testing *attribute group* subsumption is described in *Testing subsumption between two attribute groups* on page 594
3. Test that each ungrouped *attribute* in the *normalized -predicate* subsumes at least one *attribute* (either grouped or ungrouped) in the *normalized -candidate*:
   - If not, the *normalized -predicate* does not subsume the *normalized -candidate*:
     - Exit with result *false*.
   - The approach to testing *attribute* subsumption is described in *Testing attribute subsumption* on page 596.
4. If all these tests succeed, the *normalized-predicate* subsumes the *normalized-candidate*:
   - Exit with result *true*.

7.8.2.4.5.2.2 Testing subsumption between two attribute groups

The following steps test if a *predicate -attribute group* subsumes *candidate -attribute group*.

1. Check the *predicate -attribute group* for the presence of the *attribute*: 408729009 | finding context |:
   - If the group does not contain this *attribute*, apply the normal *attribute group* tests specified in *Testing a normal attribute group* on page 595.

2. If the *predicate -attribute group* contains the 408729009 | finding context | *attribute*, check whether its value is one of the following: 410516002 | known absent | or 410594000 | definitely not present |:
   - If the *attribute* exists and has one of these values, apply the tests for a *context attribute group with absent finding*, as specified in *Testing a context attribute group with absent finding* on page 595;
   - If the *attribute* exists and has any other value, apply the tests for a normal *attribute group*, as specified in *Testing a normal attribute group* on page 595.

7.8.2.4.5.2.2.1 Testing a normal attribute group

The following step tests most *attribute groups*. However, a modified approach (see *Testing a context attribute group with absent finding* on page 595) is required in the case of *attribute groups* that indicate the absence of a finding.

1. Test that each *attribute* in the *predicate -attribute group* subsumes at least one *attribute* in the *candidate -attribute group*:
   - If not, the *predicate -attribute group* does not subsume the *candidate -attribute group*:
     - Exit with result *false*.
   - The approach to testing *attribute* subsumption is described in *Testing attribute subsumption* on page 596
2. If all *attributes* in the group pass this test, the *predicate -attribute group* subsumes the *candidate -attribute group*:
   - Exit with result *true*.

7.8.2.4.5.2.2.2 Testing a context attribute group with absent finding

The following steps test most *attribute groups* that indicate the absence of a finding. This approach differs from the general tests applicable to other *attribute groups* because of the way in which assertions of absence affect the direction of subsumption. This is discussed in detail in *Recording and retrieving absent findings*.

1. Attempt to match each *attribute* in the *predicate -attribute group* with an *Attribute* which has the same name in the *candidate -attribute group*:
   - If any *attribute* in the *predicate -attribute group* is not matched by an *Attribute* with same name in the *candidate -attribute group,*the *predicate -attribute group* does not subsume the *candidate -attribute group*:
     - Exit with result *false*.
2. For each of the matched *attributes* identified in the previous step, compare the value of the *attribute* in the *predicate -attribute group* with the value of the same *attribute* in the *candidate -attribute group*:
   - If the *attribute name* is 408729009 | finding context | or 408731000 | temporal context |, the *candidate-value* must be equivalent to or subsumed by the *predicate-value*.
   - However, if the *attribute name* is 246090004 | associated finding | or 408732007 | subject relationship context |, the direction of the test is inverted. In these cases, the *predicate-value* must be equivalent to or subsumed by the *candidate-value*.
   - If any of these tests fail, the *predicate -attribute group* does not subsume the *candidate -attribute group*:
     - Exit with result *false*.
   - *Attribute values* are *expressions* and are tested in the same way as any other *expressions* (see *Testing subsumption between two normal form expressions*):
     - *Expression* subsumption testing is recursive where *expressions* include nested *qualifiers*.

3. If all the tests above are successful, the *predicate -attribute group* subsumes the *candidate -attribute group*:
    - Exit with result *true*.

7.8.2.4.5.2.3 Testing attribute subsumption

The following steps test if a *predicate -Attribute* subsumes a *candidate -attribute*.

1. Test that the candidate *attribute name* is either the same as or subsumed by the predicate *attribute name*:
    - If not, the *predicate -attribute* does not subsume the *candidate -attribute*:
        - Exit with result *false*.
    - The approach to testing *concept* subsumption is described in *Testing concept subsumption*.
2. Test that the *candidate -attribute* value is equivalent to or subsumed by the *predicate -attribute* value:
    - If not, the *predicate -attribute* does not subsume the *candidate -attribute*:
        - Exit with result *false*.
    - *Attribute values* are *expressions* and are tested in the same way as any other *expression* (see *Testing subsumption between two normal form expressions*):
        - *Expression* subsumption testing is recursive where *expressions* include nested *qualifiers*.
3. If both the above tests are successful, the *predicate -attribute* subsumes the *candidate -attribute*:
    - Exit with result *true*.

7.8.2.4.5.2.4 Testing concept subsumption

The following steps test if a *predicate - concept* subsumes a *candidate - concept*.

1. Test if candidate - *concept* is an *inactive concept*:
    - Check is the *candidate-concept* is in one of the *Historical Association Reference Sets*. Some *Historical Association Reference Sets* point to *active concept* with have replaced duplicate, erroneous or ambiguous *concepts*.
        - This can allow a replacement *active concept* to be used in subsequent steps[33]
2. Test if the *candidate - concept* is identical to the *predicate - concept*. :
    - If *candidate - concept*.id == *predicate - concept*.id the *concepts* are identical:
        - Exit with result *true* (accept equivalent).
3. Test if the *predicate - concept* is one of the *supertype ancestors* of the *candidate - concept*:
    - This is true if a sequence of I is a I *relationships* leads from the *candidate - concept* (sourceId) to the *predicate - concept* (destinationId):
        - Exit returning the result of this test.

[33] Active concepts that are related to ambiguous candidate -concepts should only be tested after deciding whether the prime objective of retrieval is "completeness" (in which case include these possible related concepts) or "precision" in which case they should be excluded unless the ambiguity can be resolved by the term selected with the original concept.

- Various approaches to optimization of this test are described in *Optimizing concept subsumption testing*. The recommended approach is to use a *transitive closure* table (see *Transitive Closure Implementation*.

7.8.2.4.6 Optimization of normalization and expression subsumption testing

The steps in the normal *transform* and subsumption testing processes are not particularly onerous. However, queries require thousands or millions of such tests to be carried out. It is therefore likely that most practical implementations will require some type of optimization to support tests for subsumption between *expressions*.

The method described in this section is one approach to optimization . The central idea is the use of a repository to store *expressions* and *relationships* between *expressions*.

The advantages of this include the following:

- All transformcomputations can be done off-line rather than at run-time;
- Less *transforms* are done as each distinct candidate *expression* need only be transformed once when created and once more each time a new release alters the definitions on which it is based:
  - Other approaches either require:
    - real - time *transformation* each time a candidate *expression* is considered for retrieval;

    or

    - storage of *normal forms* in each record entry and updating of each *normal form* instance whenever a new release affects the definitions on which it is based.

Neither of these approaches appears to be scalable over time, as record volumes increase. In contrast the proposed optimization is not affected by the total number or records but only by the total number of distinct *expressions* encountered.

- Additional optimization is possible by pre-classifying the repository so that individual queries can test an *expression* with a single join to a table representing the *transitive closure* of all used *expressions*.

The approach described in the following section is only one way of implementing the central idea of optimization using an *expression* repository. There several ways to harness the same general technique and some of these may be better suited to particular requirements or technical environments.

7.8.2.4.6.1 Expression repository design

The primary requirements for an *expression* repository are:

- Allocation of a fixed length, unique *Identifier* for every *expression* used in an operational environment:
  - The size of an operational environment may range from an individual application at a particular site to a large multi-site organization using multiple applications that use the same *expression* repository.
  - The easiest way to deliver unique *Identifiers* within this range of organizational scales is the use of a *UUID* (also referred to a GUID). The *UUID* /GUID allocation algorithm provides an industry standard approach to allocation of universally unique 128-bit *Identifiers* and is readily available in all widely used operating systems.
- Linking every close-to-user *expression* with its long *normal form*:
  - It is necessary to update this link each time a new release of *SNOMED CT* changes an underlying *relationship* that may affect the *normal form*.
- The *expressions* themselves need to be searchable:
  - The canonical version of the *SNOMED CT compositional grammar* is recommended for this purpose because it has a minimum of syntactic noise and a specified sort *order* for the elements within the *expression*.

- Despite these advantages some *normal form expressions* can be quite long. Currently the longest *normal forms* seen are up to 300 characters long. For a degree of future proofing it is suggested that the longest indexable variable length character *string* should be used (e.g. in MS SQL Server a length of up to 900 characters).

Two possible designs that meet this goal are suggested in next two sections.

- The first option uses two tables - one to identify *expressions* and the other to link them to indicate the results of *transformation* (see *Dual table expression repository*).
- The second approach is less flexible and in its current form it lacks many of the features of the first option. It is included in this guide because it follows an original suggested design and indicates an alternative for consideration and discussion. (see *Single table expression repository*).

Whichever of these designs is used the general steps in using the tables is similar (see *Using the expressions repository*)

7.8.2.4.6.1.1 Dual table expression repository

A dual table design is more flexible and potentially more compact than a *single table expression repository*. The compactness is achieved because each distinct *normal form* only occurs once in one row of the table. The flexibility results from the ability to make multiple links between *expressions* to specify the results of different *transforms* without repeating the *expression*.

Each discrete *expression* (whether close-to-user or one of the *normal forms*) is allocated a unique ExpressionId when it is first used or generated. From this point on, this ExpressionId is immutably linked to the *expression* and the *expression* must not be altered in anyway.

The ExpressionLink table represents the linkage between an *expression* and its *normal forms*. ExpressionLinks can be updated as necessary (i.e. when a new release is received) without any effect on the existing content of the *Expression* table. However, an additional row will be added to the *expression* table whenever a *transform* results in a new *expression* (i.e. any *expression* that is not in the *Expression* table).

Table 274: Suggested structure for the Expression table in a dual table expression repository

Each row in the *Expression* table represents and identifies an *expression*. All *expressions* used are identified in this way (i.e. close-to-user and *normal forms*) and the links between an *expression* and a transformed version of that *expression* are specified by the ExpressionLink table.

Primary Key	**Field Type**	**Permitted characters**	**Length**	***Description***
ExpressionId	*GUID*	binary-or - *string* version	*16/36*	Unique *Identifier for this expression*.

Data Fields	**Field Type**	**Permitted characters**	**Length**	***Description***
Expression	*String*	0 *to* 9 *and { } ( ) - , + = :*	*6 -900*	Canonical rendering of an *expression* in *SNOMED CT compositional grammar* ***Indexed***
DateAdded	*IsoDateTime*	binary-or - *string* version	*8/20*	Date time of addition of this *Expression* to the *expressions* table. YYYYMMDDhhmmss+ZZ.zz

Table 275: Suggested structure for the ExpressionLink table in a dual table expression repository

Each row in the ExpressionLink table links a source *Expression* with the result of transforming that *expression*. The DateIn and DateOut columns allow *active* and *inactive* links to be stored in the same table - easing historical review of changes. The primary key ExpressionLinkId allows multiple rows to link the same pair of *expressions* where a link is valid in one release, not valid in the next and restore in a subsequent release.

Primary Key	**Field Type**	**Permitted characters**	**Length**	***Description***
ExpressionLinkId	*GUID*	binary-or - *string* version	*16/36*	
Unique *Identifier for this expression link*				

Data Fields	**Field Type**	**Permitted characters**	**Length**	***Description***
SourceExpressionId	*GUID*	binary-or - *string* version	*16/36*	Foreign key link to the *Expression* table row for the source *expression* which is linked to a *transform* by this link. ***Indexed***
ResultExpressionId	*GUID*	binary-or - *string* version	*16/36*	Foreign key link to the *Expression* table row for an *expression* representing the result of the *transform* applied to the source *expression* ***Indexed***
TransformType	*Enum*	digits [0-9]	*2*	An enumerated value representing the nature of the *transform* between the source and result *expressions*. Values might include: 0=Single *concept expression* Long *normal form* 1=Other *expression* Long *normal form* 2=Long *normal form* Short *normal form* A direct *transform* for each *expression* to short *normal form* could be added buy is not essential this can be achieved by traversing a type 0 or 1 link followed by a type 2 link.

Data Fields	Field Type	Permitted characters	Length	***Description***
DateIn	*IsoDateTime*	binary-or - *string* version	*8/20*	Date time of addition of this CtuExpression to the *expressions* table. YYYYMMDDhhmmss+ZZ.zz
DateOut	*IsoDateTime*	binary-or - *string* version	*8/20*	Date time at which this ExpressionLink was rendered obsolete by replacement. YYYYMMDDhhmmss+ZZ.zz

7.8.2.4.6.1.2 Single table expression repository

The single table approach provides direct mapping between a close to user *expression* and a *normal form*. This was the original suggested design for an *expressions* table. However, a dual table approach provides a more efficient and more flexible solution (see *Dual table expression repository*).

Table 276: Suggested structure for a single table expression repository

CtuExpression Table				
Each row in the *Expression* table represents a close-to-user form *expression* and it *relationship* to a *normal form expression*.				
Primary Key	**Field Type**	**Permitted characters**	**Length**	***Description***
ExpressionId	*GUID*	0 *to* 9 *and A to F and { } -*	*16/38*	Unique *Identifier for this close-to-user expression.*
Data Key	**Field Type**	**Permitted characters**	**Length**	***Description***
CtuExpression	*String*	0 *to* 9 *and { } ( ) - , + = :*	*6 - 300*	Canonical rendering of close to user *expression* in *SNOMED CT compositional grammar* ***Indexed***
LongNormalExpression	*String*	0 *to* 9 *and { } ( ) - , + = :*	*6 - 900*	Canonical rendering of long *normal form expression* in *SNOMED CT compositional grammar* ***Indexed***
DateUpdated	*IsoDateTime*	0 *to* 9	*20*	Date time of last update to the LongNormalExpression for this CtuExpression. YYYYMMDDhhmmss+ZZ.zz
DateAdded	*IsoDateTime*	0 *to* 9	*20*	Date time of addition of this CtuExpression to the *expressions* table. YYYYMMDDhhmmss+ZZ.zz

7.8.2.4.6.2 Using the expressions repository

Whichever approach is taken to the design of the repository the way in which it is used is similar.

7.8.2.4.6.2.1 Run-time data entry and inbound communications

Each time an *expression* is recorded (either directly or in an inbound communication) the *expression* is looked up in the repository. The *expression* is rendered using the canonical version of the *SNOMED CT compositional grammar* and an *expression* matching this *string* is looked for in the repository.

If the *expression* is not found a new row is added to the *expression* repository.

Whether a row is found or added the unique *Identifier* of the *expression* is added to the record entry or other resource in which the information encoded by the *expression* is to be stored. Depending on authentication and other requirements, the original form of the *expression* may also be stored.

This step is often referred to a "just-in - time *precoordination*" because a *precoordinated Identifier* for the *postcoordinated expression* is generated at the time it is required. It is also possible to prime the repository with a range of *expressions* that are anticipated (e.g. because they are generated by a particular set of forms or protocols).

7.8.2.4.6.2.2 Run-time display or outbound communications

Although an *expression* repository may be shared across a large multi-site organization , there are advantages in requiring communication to adhere to standards that are not limited to bounds of that organization and which are not dependent on real - time communication with the repository. Therefore, when there is a requirement to display or communicate the information represented by an *expression Identifier*, the *expression* should be looked up in the repository and added to a communication in its original form.

7.8.2.4.6.2.3 Support for normal form transformation of new expressions

The *transforms* described in this guide are applied to all new *expressions* in the repository. Where a *transforms* results in a new *expression*, this *expression* is added to the repository. The appropriate reference between the original *expression* and *normal form expression* is created, in a manner determined by the repository design.

7.8.2.4.6.2.4 Support for normal form transformation after updates to SNOMED CT definitions

After a *SNOMED CT* update, the repository is refreshed to check for consequent changes in the *normal forms* for existing *expression*. Where changes are required the appropriate rows in the repository are added or updated in accordance with the repository design.

7.8.2.4.6.2.5 Supporting retrieval with an expression repository - basic

Retrieval requests can be dealt with by using the repository to locate the appropriate *normal forms* for the predicate *expression* and for candidate *expressions*. Instead of requiring processing to *transform expressions* in real-time a simple SQL *query* can immediately return the appropriate *expression*.

The general process of process testing subsumption between *expression* is then applied as documented in this guide (see *Testing subsumption and equivalence between expressions*).

7.8.2.4.6.2.6 Supporting retrieval with an expression repository - advanced

Further optimization is possible if the *expressions* in the repository are classified to generate an extended *transitive closure* table. This process is similar in effect to testing for subsumption between every pair of *expressions* in the repository and recording the results of each successful test as a row in a table that identifies the *relationship* between the subsuming and subsumed *expression*. This process may appear to be unscalable because it requires many millions of tests to be carried out. However, fortunately algorithms are available that can optimize this process and classify hundreds of thousands of *concepts* in a little over an hour on what would today be considered a fairly modest system.

If this approach is followed, any predicate *expression* can be tested against any candidate *expression* by a simple SQL *query* using this extended *transitive closure* table. The result of this is that testing subsumption of an *expression* will perform practically as fast as testing the subsumption between two *precoordinated concepts*.

7.8.2.4.6.2.7 Is storing an expression the same as creating a new concept?

In one sense adding an *expression* to a repository and giving it an *Identifier* is the same as creating a new *concept*. However, there are some subtle but highly significant differences between storing, identifying and reusing an *expression* in the way suggested in the guide and a *SNOMED CT concept*. These are summarized in *Differences between concepts and stored expressions*.

Table 277: Differences between concepts and stored expressions

SNOMED CT* released *concept	**Stored *expression***
The defining *relationships* of a *SNOMED CT concept* are intended to represent the meaning of words or phrases as they are used in clinical practice.	An *expression* is the collection of references to a set of *SNOMED CT concepts*.
The meaning of a *SNOMED CT concept* is represented by the *Fully Specified Name* (and sometimes by an associated textual definition).	An *expression* is not associated with a specific text *string*. It may be rendered in different human readable forms but its only source of meaning is the meaning of the *concepts* it references.
Because a *concept* definition attempts to express the human understood meaning of a word or phrase the logical definition expressed by its defining *relationship* may not be sufficient to fully define the *concept*. In these cases the *concept* definition is marked as "*primitive* ".	Because an *expression* has no specific *term* or source of meaning other than the *focus concept* and *Attribute* it is inherently "*fully defined*" in that those *Attributes* fully define what the *expression* may be used to represent.
A *SNOMED CT Concept* can be bound to various *terms* that are deemed to be *synonyms* in a given *language*. The *SNOMED CT* design provides a framework for managing these bindings, correcting errors, supporting translations and tracking the history of changes.	It may be tempting to associate particular words or phrases with an *expression*. This will inevitably occur in instances in individual record entries. However, *terms* should not be bound to *expressions* in a way that suggests a formal persisting association between that *term* and the class represented by the *expression*. If such a binding is required a *SNOMED CT concept* should be requested (or created in an *extension*) to provide a proper framework for managing that binding.

7.8.2.4.7 Retrieving absent findings

This part of the guide is based on a white paper produced in May 2006 to consider the impact of "negatives" on *transformation*, normalization and *subtype* testing rules. The outcome of this was revision of the rules on subsumption testing in relation to context *attribute groups* that include finding context values indicating that a finding is known to be absent.

7.8.2.4.7.1 Rationale

The wider issue of different types of negation cannot be completely resolved in a short space of time - as has been demonstrated during numerous previous discussions. However, there are some aspects of current advice on computation of subsumption that appear to be misleading in relation to *concepts* that express the absence of a finding.

For example, current *subtype* testing rules on the following *expression*:

373572006 | clinical finding absent | :

246090004 | associated finding | = (125605004 | fracture of bone | :

363698007 | finding site | = 71341001 | bone structure of femur |)

normalizes as follows

243796009 | context-dependent category | :

{246090004 | associated finding | =

(64572001 | disease | :

{116676008 | associated morphology | = 72704001 | fracture |

,363698007 | finding site | = 71341001 | bone structure of femur | })

,408729009 | finding context | = 410516002 | known absent |

,408731000 | temporal context | = 410512000 | current or specified |

,408732007 | subject relationship context | = 410604004 |subject of record| }

The result of applying "normal" subsumption testing rules is that | no fracture of femur | is subsumed by | no fracture of bone |. Superficially this may seem reasonable, but it will incorrectly cause the inference that a person with a record of | no fracture of femur | has | no fracture of a bone |. This is true if | no fracture of a bone | meant one bone that it not fractured, but the generally understood meaning would be that the patient had no fractured bones.

Thus the objective was to revise the *transformation* and/or *subtype* testing rules to appropriately handle *expressions* that represent absent findings.

7.8.2.4.7.2 Overview

The approach specified in this guide deals with the computational issue of *subtype* testing based in the current *concept model*, *classifier* logic and distribution format of *SNOMED CT*. The approach has been tested and produces reasonable results with current data. It also works appropriate with combined presence and absence finding (e.g. "head injury without skull fracture") provided these are modeled using separate context *attribute groups*.

The positive statement in the previous paragraph must be tempered by the knowledge that a logical technical approach is only a part of the solution. Human factors are an important issue when considering the proper processing of *concepts* of absence and other forms of negation. Therefore, it is also necessary to consider what people may mean when they explicitly state the "absence of a finding"; and what other people may intend when they query records to determine the presence or absence of a finding.

The technical approach suggested for subsumption testing *expressions* that involve absence of a finding are valuable only if applied appropriately. Human interpretation may be required to determine the clinical relevance of the results of absent *subtype* tests for a particular purpose.

7.8.2.4.7.3 Testing subsumption of absence of a finding

7.8.2.4.7.3.1 Initial assumptions

The general rules for computation of subsumption of *expressions* and *transformation* to *normal forms* are stated in detail in the *SNOMED CT* document on *transformation* to *normal forms*. They can be summarized as follows:

When two *expressions* are tested for subsumption, tests are performed recursively on the following elements within the *normal form* of those *expressions*:

- Groups of *attribute value* pairs;
- *Attribute value* pairs;
- Nested *expressions* use to represent values within an *attribute value* pair.

The *normal form* of an *expression* is derived by a set of rules which retain the full semantic meaning of the original *expression* while *transformation* it to a form in which:

- Every referenced *focus concept* is a *primitive concept*
- Every *attribute value* is a normalized *expression*
- Grouping and nesting of *Attributes* is aligned with the *concept model*

- Default context or context derived from the information model is made explicit using *SNOMED CT* context *Attributes*

7.8.2.4.7.3.2 Identifying expressions that include absence

The *normal form* of any *expression* that represents absence of finding includes the following standard context *Attributes*:

243796009 | situation with explicit context |:

408729009 | finding context | = 410516002 | known absent | *(or a subtype)*

,408731000 | temporal context | =*<temporal context value>*

,408732007 | subject relationship context | = *<subject Relationship context value>*

,246090004 | associated finding | =*<a clinical finding or event>*

When the value of | finding context | is | known absent | (or one of its *subtypes*) then it may be appropriate to apply *subtype* testing rules based on absence. However, as discussed in *recording and retrieving absent findings*, the way in which rules are applied depends on the intended results of the *query* and rationale behind recording a negative finding.

7.8.2.4.7.3.3 Testing groups rather than expressions

The relevant information in an *expression* can be regarded as a group of *attributes* as follows:

{ 408729009 | finding context | = 410516002 | known absent | *(or a subtype)*

, 408731000 | temporal context | = *<temporal context value>*

, 408732007 | subject relationship context | =

, 246090004 | associated finding | = }

Considering absence at the group level, rather than at the *expression* level, allows account to be taken of *expressions* that refer to presence of one finding and absence of another.

The following style of *expression* represents the presence of "first clinical finding" and the absence of "second clinical finding".

243796009 | situation with explicit context | :

{ 408729009 | finding context | = 410515003 | known present | *(or a subtype)*

, 408731000 | temporal context | = *<temporal context value>*

, 408732007 | subject relationship context | =

, 246090004 | associated finding | = }

{ 408729009 | finding context | = 410516002 | known absent | *(or a subtype)*

, 408731000 | temporal context | = *<temporal context value>*

, 408732007 | subject relationship context | =

, 246090004 | associated finding | = }

In this case, the first group is tested according to the general subsumption testing rules and the approach to absence may be appropriate to the second group (i.e. the group that includes | finding context | = | known absent |).

The overall *expression*, containing both these groups, is then tested in the general way according to whether the two groups separately pass the relevant test. The general subsumption testing rules allow groups not present in the predicate *expression* to be present in the candidate *expression*. Therefore both of the following predicate *expressions* subsume the candidate *expression* above irrespective of the special rules for handling absence.

<u>Predicate 1 - "first clinical finding present"</u>

243796009 | situation with explicit context | :

{ 408729009 | finding context | = 410515003 | known present | *(or a subtype)*

, 408731000 | temporal context | = *<temporal context value>*

, 408732007 | subject relationship context | =

, 246090004 | associated finding | = }

<u>Predicate 2 - "second clinical finding absent"</u>

243796009 | situation with explicit context | :

{ 408729009 | finding context | = 410516002 | known absent | *(or a subtype)*

, 408731000 | temporal context | = *<temporal context value>*

, 408732007 | subject relationship context | =

, 246090004 | associated finding | = }

7.8.2.4.7.3.4 Testing | associated finding | in groups containing | known absent |

If a group contains | finding context | = | known absent | then the test applied to the value of the | associated finding | *Attribute* is changed.

The general purpose test for the value of an *Attribute* is:

- "is the candidate value identical to or a *subtype* of the predicate value".

The alternative test when the group contains | known absent | is:

- "is the **predicate** value identical to or a *subtype* of the **candidate** value".

7.8.2.4.7.3.5 Testing | subject relationship context | in groups containing | known absent |

If a group contains | finding context | = | known absent | then the test applied to the value of the 408732007 | subject relationship context | *Attribute* should also be changed to the alternative form.

- "is the **predicate** value identical to or a *subtype* of the **candidate** value".

Thus

- | family history of heart disease in father | implies | FH: Cardiac disorder |; but;
- "no family history of heart disease" implies "no family history of heart disease in father".

7.8.2.4.7.3.6 Testing | temporal context | in groups containing | known absent |

If a group contains | finding context | = | known absent | then the test applied to the value of the | temporal context | *attribute* needs to be carefully considered depending on the intended result of the *query*.

In some cases it may also be changed to the alternative form.

- "is the **predicate** value identical to or a *subtype* of the **candidate** value".

Thus:

- "*currently has asthma*" implies "*has, or at some time in past had, asthma*"; but;
- "*did not have headache recently*" does not imply "*did not have headache in the past*".

However, since the value "all times past" is specified for expressing *concepts* like | never had a headache | the standard *subsumption test* rules may work in some cases.

Since the time aspect in the record is relative to the time of recording while the intended result of a *query* may be relative to a specified time (or the time of the *query*) the use of temporal context in queries requires careful consideration on a *query* by *query* basis.

7.8.2.4.7.3.7 Differences between subject Relationship and temporal context

The difference between the handling of "408732007 | subject relationship context |" and | temporal context | noted in *Testing |subject relationship context| in groups containing |known absent|* and *Testing |temporal context| in groups containing |known absent|* may result from a significant difference in value hierarchies.

Thus "no family history of asthma" literally means something like:

"As far as is known, at all times in the past , the disorder asthma was absent from, all members of the subjects family "

The temporal context value *hierarchy* includes the value "all times past" to capture one part of this. However, for the "all *Members* of the subject's family" we use the same | Person in the family |*concept* as is used for asserting "at least one *Member* of the family".

An argument can be made for aligning the approach in both these hierarchies in one of two ways:

1. Removing the value "all times past" from | temporal context | and using "current or past" in its place. Then the *subtype* testing of temporal context would invert in the same way as for the other *Attributes* (i.e. in absence mode the "current or past" would imply all other temporal context ... aka "all times past").
2. Adding "all *Members* of family" to the subject *relationship* value *hierarchy* and carefully applying this in all negation *expressions*. In this case, the alternative *subtype* testing would only apply to | associated finding |.

While approach (b) may appear more rational it does seem to have two disadvantages:

- It requires more disciplined use in modeling and in *postcoordination*
- Several new "all" values would be needed - "all *Members* of paternal family", "all male *Members* of family", "all known contacts", etc. to allow negatives to be expressed clearly.

7.8.2.4.7.3.8 Impact of nesting context expressions

Currently, the *concept model* does not allow a *subtype* of | Situation with explicit context | to be the value of a defining *Relationship*. However, some potential use cases have been advanced for allowing a | Finding with explicit context | to be the value of an *attribute*. If this is permitted then inclusion of | known absent | in such a nested *expression* would create additional complexity when trying to resolve queries.

Applying the current testing rules at appropriate nested levels may have the desired result. However, there would be an increased risk of double-negatives and similar logical problems. Therefore, until there are real cases to test, the possibility of new exceptions arising cannot be ruled out.

7.8.2.4.7.4 Human factors and testing absence

The reasons for recording information about absent findings and the rationale for attempting to retrieve information about absent findings often differ from and interact with the strict logical interpretation of negation. Specific aspect of this general point are illustrated by the following subsections.

7.8.2.4.7.4.1 Subtype classification of absent findings

When considering subsumption testing as part of the process of classifying the *concepts* in *SNOMED CT* the underlying assumptions is that the comparison process is potentially symmetrical. Thus any two *concepts* can be compared to ask the following questions:

- Are A and B identical? ... if not then
- Is A a *subtype* of B? ... if not then
- Is B a *subtype* of A?

If not then we might possibly be interested in the semantic proximity of the *concepts* for example ...

- What supertypes do A and B share?
- Are there any *concepts* that are subsumed by both A and B.

In this relatively abstract environment it is possible to discuss ideas about | known absent | or | not done |. These ideas may seem theoretically sound while being less readily applicable in practical clinical applications. In some cases the practical view may be more complex than the abstract view but in the case of "absence" it seems possible that considering real use cases may in some ways simplify or at least assist in prioritization.

7.8.2.4.7.4.2 Querying records for absence findings

Subsumption testing in a clinical application is typically concerned with testing instances of *expressions* in clinical records ("candidate *expressions* ") against sets of criteria some of which are represented as *expressions* ("predicate *expressions* ").

- A predicate is an *expression* against which other *expressions* are tested. Predicate *expression* may be constructed for specific queries or may be developed as reusable part of clinical protocols, decision support rules or report specifications. In these cases, the author of a predicate is someone trying to find out something by querying a record or set of records.
- The candidate is an *expression* that is tested to see if it is subsumed by the predicate. Candidate *expressions* may be constructed directly by the author of a clinical statement (i.e. an instance of an entry in the record) or by an application designer determining the way in which particular user decisions are recorded. Thus the direct or indirect author of the candidate is typically someone wishing to record (or enable the recording of) a finding or procedure in a record. Although the candidate *expression* is a crucial part of subsumption testing its reason for existing is not determined by the requirements of a specific *query* but rather by what the user wishes to record.

The more abstract subsumption testing for classification described in the previous section is a prerequisite for effective subsumption testing in clinical applications. However, the differences between the motivations of those constructing predicate and candidate *expressions* mean that subsumption testing in clinical applications is rarely a symmetrical comparison. The typical test is "does this candidate satisfy the criteria?" or in some cases "could this candidate possibly satisfy the criteria?"

When considering absence or other kinds of negation the difference between the perception of the author of an instance of clinical information and the view of the person constructing a *query* may be even more significant. Thus technical rules for testing subsumption of | known absent | finding are only one part of the picture.

To avoid misunderstanding and consequent errors it is worth considering two general questions:

- What are the possible motivations for recording a | known absent | finding?
- What are the possible motivations for specifying retrieval queries for absent findings?

The next two sections identify several different answers to these questions.

7.8.2.4.7.4.3 Motivations for recording a | known absent | finding?

There thousands of possible findings that might be made at every encounter (and theoretically every second). The vast majority of absent finding are not recorded but there are clearly some good reasons for explicitly recording the absence of some findings. These might include:

1. To record that the author asked a question and got a negative response.

   Example:

   "Family history - No family history of asthma".

   Implied meaning - "I asked the patient if anyone in their family has or had asthma and they said 'no'".

2. To record that the author examined/investigated and did not find this.

   Example:

   "No heart murmur".

   Implied meaning - "I listened for a heart murmur and did not hear one".

3. To record a possible conclusion that the author considered and rejected.

Example:

"No meningitis" (as part of an assessment of a patient with a fever and headache).

Implied meaning - "I considered the possibility of meningitis and rejected it".

4. To refute a statement made by someone else.

Example:

"Not appendicitis" (in a record that contains an earlier assertion of "diagnosis appendicitis").

Implied meaning - "The admitting doctor's diagnosis of appendicitis seems to be incorrect".

5. To indicate a change in an earlier assessment.

Example:

| Carcinoma of bronchus excluded | (as part of record in which the same author previously thought this a likely diagnosis).

Implied meaning - "I thought they might have Ca bronchus but following investigations I have now rejected this diagnosis".

6. To note the resolution of finding that was previously present.

Example:

"No abdominal pain" (in a record which has a previous finding of "abdominal pain present").

Implied meaning - "The abdominal pain present on admission has now resolved".

7. To indicated that a finding that is commonly present in associated with another finding is not present in this case.

Example:

"No loss of consciousness" (in the record of patient who has had a head injury).

Implied meaning - "They did not lose consciousness following an injury which potentially could have caused this".

In (1), (2) and (3) the dominant motive may be to assert what was done or considered. However, recording absent findings may also be a part of the process the author followed to organize her thoughts.

Both (4) and (5) record difference in view related to some previous assertion. Where this is the intention a strong case can be made for linking the statements in the record structure. However, this is a possible motivation even if such links are supported by the system or have not been added to this instance.

In the case of (6) the use of absent indicates a change in condition of the patient rather than an update of the diagnosis or interpretation by the clinician.

In case (7) an absent finding is recorded to refine the nature of a specific condition.

There is considerable overlap between these reasons motivations for recording absence. However, the overall motivation for recording an absent finding may or may not be aligned with the rationale for requesting retrieval of negative findings. This mismatch is likely to lead to lead to anomalous results if the assumptions based only on a logical interpretation of negation.

7.8.2.4.7.4.4 Motivations for specifying retrieval of "known absent" finding?

When querying for absence of a finding the most likely motivation is to establish the absence of a finding. In the absence of evidence to the contrary the normal assumption is that an abnormal finding is absent. Furthermore, in most cases a point in time assertion of absence does not imply the finding was never true, nor that could not be true at a future point in time. Thus in most cases, a *query* for absence is more concerned with checking that there is no statement indicating the presence of the finding rather that searching for a statement of presence.

There are exceptions to this:

- If the abnormal finding is usually found in associated with a confirmed finding:
  - E.g. "absence of chest pain" in a patient with a confirmed "myocardial infarction").
- If the abnormal finding is obscure and may easily have been overlooked:
  - E.g. "Kopliks spots".
- If an assertion of presence of a finding was made by an informer of unknown reliability:
  - E.g. Bystander asserts that patient had a "heart attack" but clinical assessment excludes this.

If these exceptions apply the presence of a statement of an absent finding may be of interest. However, this depends of specific thinking around the question being posed so that the *query* criteria achieve the desired result. It is not enough to simply search for a specific absence and its *subtypes*. The assumptions about presence or absence of a finding must be considered.

Example: determine the number of road accident victims who have been admitted to hospital but have no fractured bones. In practical *terms* the best approach would just be to exclude those with known presence of fracture. The assumption is that, unless a fracture is mentioned, they are not known to have a fracture.

Another possible motivation for looking for absence findings is to monitor or audit the delivery of care and check that appropriate questions have been asked, tests done, possibilities considered, etc. In these cases, the *query* needs to search for both presence and absence ... or possibly for a procedure code representing the appropriate examination or investigation.

Example: Were all patients admitted for routine surgery asked if they had any allergies.

7.8.2.4.7.5 Conclusions on absent findings

There are rational reasons for wishing to record and retrieve information about absent findings. However, there is not a direct one-to-one *relationship* between the motivations for recording absence and the motivations for retrieval.

The suggested technical advice on subsumption testing of known absent findings addresses the logical question of subsumption, but this is only one part of the picture. The meaning implied by recording a known absent finding needs to be considered in the context of the intention of a *query*. When this is understood the alternative *subsumption test* can be applied appropriately to support complete and accurate retrieval.

7.8.2.4.7.6 Procedures |not done |

The use of the |procedure context | value | not done | (and *subtypes*) has similarities to the | finding context | value | known absent |. The same alternative rules for subsumption computation could be applied to | associated procedure | value. Similar human factor considerations are also likely to apply.

The range of procedure context values is wider and covers decision, request and intent as well as the simple observation that something was not done. Thus variants such as "not to be done" and "not requested" also need to be considered.

7.8.2.5 Terminology query languages

A terminology query language goes beyond *compositional grammar*: it supports additional criteria used to filter content that is not necessarily part of a *concept* definition.

Example use cases:

1. The department responsible for clinical research needs to create queries that will select a portion of the terminology (i.e. "all infectious diseases caused by gram negative bacteria") that are meaningful for some specific purpose.

2. A group planning a translation project needs to define queries that will extract subject-specific *refsets* to be forwarded to specialists groups.

Example query: All the *subtypes* of "Clinical finding" that include a finding site *descendant* of "thorax" and the FSN matches "*pain*"

The *Query Specification Reference Set* can be used to store string forms of terminology queries, associated with a *reference set*; this enables the generation of its members.

An standard specification for a Query Language for *SNOMED CT* is not yet defined. Many terminology management tools have developed their own query languages and representations. The "*Refset* Specifications" in the *IHTSDO workbench* is a representative example.

Expression languages or *concept* definition representation grammars can be used as query languages, using the techniques described in *Concept definition queries* and *Expresion retrieval*.

7.8.3 Creating legacy queries

A *terminology server* may also support the creation of queries that retrieve data encoded in *SNOMED International* (*SNOMED CT* version 3.x), *Clinical Terms Version 3* or earlier versions of the *Read Codes*.

For example, a *terminology server* may generate a SQL predicate list that includes the *Read Codes*, *CTV3IDs* or *legacy SNOMED codes* of all unique *subtype descendants* of a specified *Concept*. Some *constraints* on this functionality may be necessary as top-level or other general *Concepts* may generate extremely long lists of *descendant Identifiers*.

7.9 Creating and maintaining Reference Sets

This section describes the basic steps required to create and maintain *Reference Sets*.

7.9.1 How to create a new Reference Set using an existing pattern

In order to create a new *Reference Set*, you will need access to a *namespace* in order to generate SCTIds. Within your *namespace*, you should add one *moduleId concept* (with an FSN and *Preferred Term*), under the |Module| sub - *hierarchy* within the metadata, for each of your authoring organizations .

Then, follow the steps below to create a new *reference set*:

- *Define the Reference Set in the metadata hierarchy*;
- *Define the Reference Set Attributes within the metadata hierarchy*;
- *Create the Descriptor for the Reference Set*;
- *Add members to the Reference Set*.

Caution: All *components* created during these processes must have unique *Identifiers* and all those *Identifier* must be allocated in the correct partition of your organizations namespace. For details see *SNOMED CT Identifiers*.

7.9.1.1 Define the Reference Set in the metadata hierarchy

First, create a *concept* for the *Reference Set*:

Table 278: Reference Set Management Example - Add Reference Set Concept

Field	Data type	Value
id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *reference set*.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .

Then, add up to three *Descriptions* for the FSN, the *Preferred Term* and optionally the Purpose:

Table 279: Reference Set Management Example - Add Descriptions for Concept

Field	Data type	Value
id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *reference set*.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
conceptId	*SCTID*	The *Identifier* of the *concept* describing the *Reference Set* that you've just added.
languageCode	*String*	The *language* of the *Description*.
typeId	*SCTID*	Create up to three *descriptions*, with each of the following types: |FSN|, | *Synonym* |, |Purpose|. The first two are mandatory, the third is optional.
term	*String*	*Terms* for the FSN, the *Synonym* and the | Purpose |. The *Synonym* will be the *string* used to commonly refer to the *Reference Set*. The conventions for creating *terms* for the FSN and *Synonym terms* are described in *Naming Conventions for Reference Sets*.

Add an | is a | *Relationship* to link the *Reference Set* to the appropriate pattern:

Table 280: Reference Set Management Example - Link to Metadata for Reference Set Pattern

Field	Data type	Value
id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *reference set*.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
sourceId	*SCTID*	The *Identifier* of the *concept* describing the *Reference Set* that you've just added.
destinationId	*SCTID*	The *concept* describing the pattern that this *Reference Set* follows, a *descendant* of | *Reference Set* | in the metadata *hierarchy*.
relationshipGroup	*Integer*	'0'
typeId	*SCTID*	| is a |
characteristicTypeId	*SCTID*	|Stated *relationship* |

7.9.1.2 Define the Reference Set Attributes within the metadata hierarchy

Add new *concepts* for each of the *Reference Set* member attributes, if necessary. If the *Reference Set* attributes describing the pattern are adequate to describe the *Reference Set*'s attributes, then these can be used instead, and you can skip to the next section.

You may wish to create your own *Reference Set* attributes for one of the following reasons:

- You wish to give one or more of the attributes a different name than that of the pattern;
- You wish to make the purpose of a particular attribute more explicit in the metadata;
- You wish to limit the set of allowed values for one or more of the attributes;
- You wish to make the type of one or more of the attributes more specific than that given in the pattern.

You may add new *concepts* for some of the attributes, and reuse existing *concepts* for other attributes, if you wish.

For each attribute that you wish to create, first add a *concept*:

Table 281: Reference Set Management Example - Add Reference Set Concept

Field	Data type	Value
id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *reference set*.

Field	**Data type**	**Value**
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .

Then, link it with an | is a | *Relationship* into the | *Reference set* attribute | metadata *hierarchy*.

Table 282: Reference Set Management Example - Link to Metadata Hierarchy

Field	**Data type**	**Value**
id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *reference set*.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
sourceId	*SCTID*	The *Identifier* of the *concept* describing the *Reference set* attribute that you've just added.
destinationId	*SCTID*	| Reference set attribute |
relationshipGroup	*Integer*	'0'
typeId	*SCTID*	| is a |
characteristicTypeId	*SCTID*	| Stated relationship |

Then, add up to three *Descriptions* (for FSN, *Preferred Term* and optionally Purpose) for each of the new attributes:

Table 283: Reference Set Management Example - Add Descriptions

Field	**Data type**	**Value**
Id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *reference set*.
active	*Boolean*	'1'

Field	**Data type**	**Value**
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
conceptId	*SCTID*	The *Identifier* of the *concept* describing the attribute that you've just added.
languageCode	*String*	The *language* of the *Description*.
typeId	*SCTID*	Create up to three *Descriptions* for each new attribute, with the following types: | Fully specified name |, | Synonym |, | Purpose |. The first two are mandatory, the third is optional.
term	*String*	*Terms* for the | Fully specified name |, a | Synonym | and the | Purpose |. The | Synonym | will be the *string* used to commonly refer to the attribute (and therefore should appear as a column header in tables showing the *Reference Set* member records).

If any of the *Reference Set* member attributes are to be limited to a range of values, then add a *concept* for each allowed value in the range, and link the *concept* using an |Is a| *relationship* to the member attribute. Then add two *Descriptions* for the FSN and *Preferred Term* of each allowed *attribute value*.

In order to limit the range of an attribute, it must have a type of | *Concept* type component| (as held in the *attributeType* field of the Descriptor - see the next section).

For each allowed value that an attribute can take, add a *concept*:

Table 284: Reference Set Management Example - Add Allowed Attribute Value Concept

Field	**Data type**	**Value**
id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *reference set*.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .

Then, link it with an | is a | *Relationship* into the attribute that you've just added in the | Reference set attribute | metadata *hierarchy*.

Table 285: Reference Set Management Example - Link Allowed Attribute Value to Metadata

Field	Data type	Value
id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *reference set*.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
sourceId	*SCTID*	The *Identifier* of the *concept* describing the allowed *attribute value* that you've just added.
destinationId	*SCTID*	The *Identifier* of the *concept* describing the attribute that you've just added.
relationshipGroup	*Integer*	'0'
typeId	*SCTID*	| is a |
characteristicTypeId	*SCTID*	| Stated relationship |

And finally, add two *Descriptions* for the allowed *attribute value concept*:

Table 286: Reference Set Management Example - Add Description for Allowed Attribute Value

Field	Data type	Value
Id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *reference set*.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
conceptId	*SCTID*	The *Identifier* of the *concept* describing the allowed *attribute value* that you've just added.
languageCode	*String*	The *language* of the *Description*.

Field	Data type	Value
typeId	*SCTID*	Create two *descriptions*, with each of the following types: IFSNI, I *Synonym* I
term	*String*	*Terms* for the I Fully specified name I and a I Synonym I. The I Synonym I will be the *string* used to commonly refer to the allowed *attribute value* (and therefore should be the one shown in pick lists used when maintaining *Reference Set* member records).

7.9.1.3 Create the Descriptor for the Reference Set

Add one record to the I *Reference Set* DescriptorI *Reference Set* describing the *referencedComponentId* attribute, and one additional row for each additional optional attribute within the *Reference Set*.

These records together describe the structure of the *Reference Set*, and are called the Descriptor of the *reference set*, for short. If the existing Descriptor Template (that describes the *Reference Set*'s pattern) also adequately describes the *reference set* that you've just created, then a new Descriptor need not be created, and this section may be skipped.

Where a Descriptor is created for a new *Reference Set*, it should have the same structure (i.e. - an identical number of records, each of the same attribute type or *subtype*) as the *Reference Set* Descriptor that described the parent *Reference Set* pattern.

Table 287: Reference Set Management Example - Add Reference Set Descriptor

Field	Data type	Value
id	*UUID*	A unique *UUID* for this record.
effectiveTime	*Time*	The nominal date of release for your *reference set*.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
refsetId	*SCTID*	900000000000456007 I Reference set descriptor reference set (foundation metadata concept) I
referencedComponentId	*SCTID*	Set to the *concept* describing the *Reference Set* that you've just created.
attributeDescription	*SCTID*	Set to the *concept* describing the attribute that you've just created, or alternatively an existing *concept* under the 900000000000456007 I Reference set descriptor reference set (foundation metadata concept) I metadata *hierarchy*.

Field	Data type	Value
attributeType	*SCTID*	Set to a *descendant* of 900000000000456007 | Reference set descriptor reference set (foundation metadata concept) | in the metadata *hierarchy*. This field describes the type of the attribute. If an attribute has been limited to a range of values, then this field must always be set to 900000000000456007 | Reference set descriptor reference set (foundation metadata concept) |. If a *Reference Set* is the *child* of a particular *subtype* of 900000000000456007 | Reference set descriptor reference set (foundation metadata concept) |, this field must be the same as or a *descendant* of the equivalent field for the more general *Reference Set subtype*.

7.9.1.4 Add members to the Reference Set

Follow the steps in the next section to maintain the members of the *Reference set*.

7.9.2 How to add, change or remove members of an existing Reference Set

You should only add members to a *Reference Set* in you organization's namespace or in the namespace of an organization that has authorized you to edit that *Reference Set* and provided you with a *moduleId* in their namespace to use for that purpose.

To add a member to an existing *Reference Set*, create a new record as follows:

Table 288: Reference Set Management Example - Member Added

Field	Data type	Value
id	*UUID*	A unique *UUID* for the new member record.
effectiveTime	*Time*	The nominal date of release that this member is to be first introduced in.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
refsetId	*SCTID*	The id of the *concept* that describes the *Reference Set* that you're adding a member to.
referencedComponentId	*SCTID*	A reference to a *component* , of type (and possibly range) limited by the Descriptor record for this *Reference Set* with *attributeOrder* '0'.
additional field 1		An optional *Attribute*, with a value, of type (and possibly range) limited by the Descriptor record for this *Reference Set* with *attributeOrder* '1'.

Field	Data type	Value
additional field 2		An optional *Attribute* with a value, of type (and possibly range) limited by the Descriptor record for this *Reference Set* with *attributeOrder* '2'.

To delete an existing member from a *Reference Set*, create a new record as follows:

Table 289: Reference Set Management Example - Member Deleted (made inactive)

Field	Data type	Value
id	*UUID*	A unique *UUID* of the existing member record that you wish to delete.
effectiveTime	*Time*	The nominal date of release in which this member is to be deleted.
active	*Boolean*	'0'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
refsetId	*SCTID*	As value in existing record
referencedComponentId	*SCTID*	As value in existing record
additional field 1		As value in existing record
additional field 2		As value in existing record

To modify an existing member in a *Reference Set*, create a new record as follows:

Table 290: Reference Set Management Example - Member Modified

Field	Data type	Value
Id	*UUID*	A unique *UUID* for the existing member record that is to be updated.
effectiveTime	*Time*	The nominal date of release that the update is to become *active* in.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .

Field	Data type	Value
refsetId	*SCTID*	As value in existing record. A member cannot move from one *reference set* to another.
referencedComponentId	*SCTID*	As value in existing record. A member cannot change the *component* that it refers to. Instead, the existing member record should be deleted, and a new one created.
additional field 1		This field may be updated. An optional *Attribute*, with a value, of type (and possibly range) limited by the Descriptor record for this *Reference Set* with *attributeOrder* '1'.
additional field 2		This field may be updated. An optional *Attribute* with a value, of type (and possibly range) limited by the Descriptor record for this *Reference Set* with *attributeOrder* '2'.

7.9.3 How to create a new Reference Set pattern

In order to create a new *reference set* pattern, follow to steps to create a new *reference set*, with the following exceptions:

- The *concept* describing the *Reference Set* pattern should be created as an immediate *child* of the | *Reference set* | *concept*, or as a *child* of another *Reference Set* pattern.
- The *Descriptions* of *typeId* | *Synonym* | and IFSN| should be of the form:
  - *My pattern name*type;
  - *My pattern name* type *reference set* (foundation metadata *concept*).
- A Descriptor Template must be created for a pattern, following the steps as described to create a Descriptor for a *Reference Set*.

7.10 Terminology Server Software

This section outlines the possible characteristics of software that provides *Terminology services* through a programmable interface. Such software represents an approach to development that may enable more rapid implementation of *SNOMED CT*.

This guide does not specify a particular *Application Programming Interface* (*API*) for accessing *SNOMED CT* services. Instead it sets out the general principles and options for delivery and use of a *terminology server*.

7.10.1 Terminology server functionality

A *terminology server* should be able to deliver all the essential *Terminology services* identified in the *Terminology Services Guide (RF2) (7)*. It should also provide the recommended *Terminology services* and should achieve a performance that meets the more general requirements for the functionality of *SNOMED CT enabled applications*.

Terminology server may provide two types of service:

- Reference Services (see *Figure 132*):
  - Services that do not include a *user interface*;
  - The client application may use reference services to undertake many different functions;
  - For some of these functions the client application will populate an appropriate *user interface component*.

Example:

A reference server may return a list of *Descriptions* matching a particular search *string*. The client application may use this data to populate a list from which a user makes a selection.

- *User Interface* (*UI*) Services (see *Figure 133*):
  - Services that include the one or more *user interface components* that can be used in and programmatically accessed by the client application.

Example:

A *UI* server may provide a control that includes a text box and a list. When the user types in the text box, the server populates the list and allows the user to select an item. The selected item is accessible from the client program.

- One possible type of *UI* service is a *SNOMED CT browser* with an *API* for returning selected data to a client application:
  - This may be useful as mechanism for providing some *SNOMED CT* capabilities to an application. However, it is less suitable for frequent entry of *SNOMED CT* encoded information.

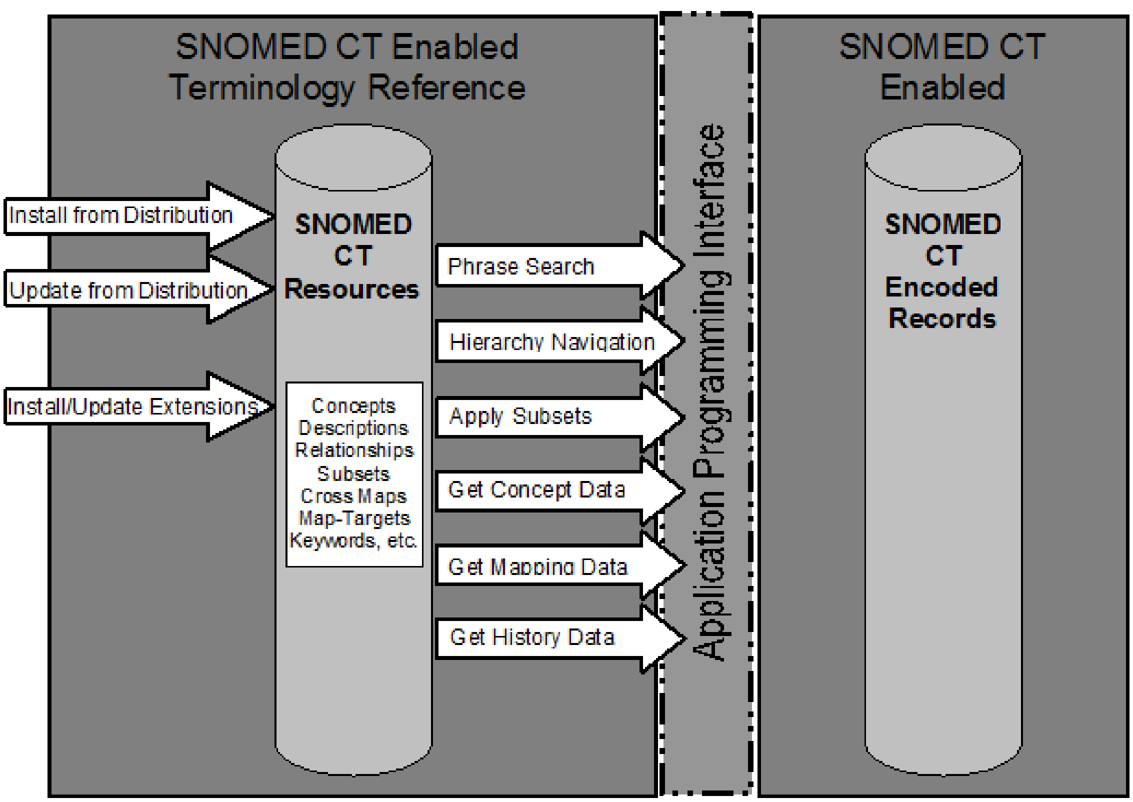


Figure 132: Terminology server providing reference services to a client application

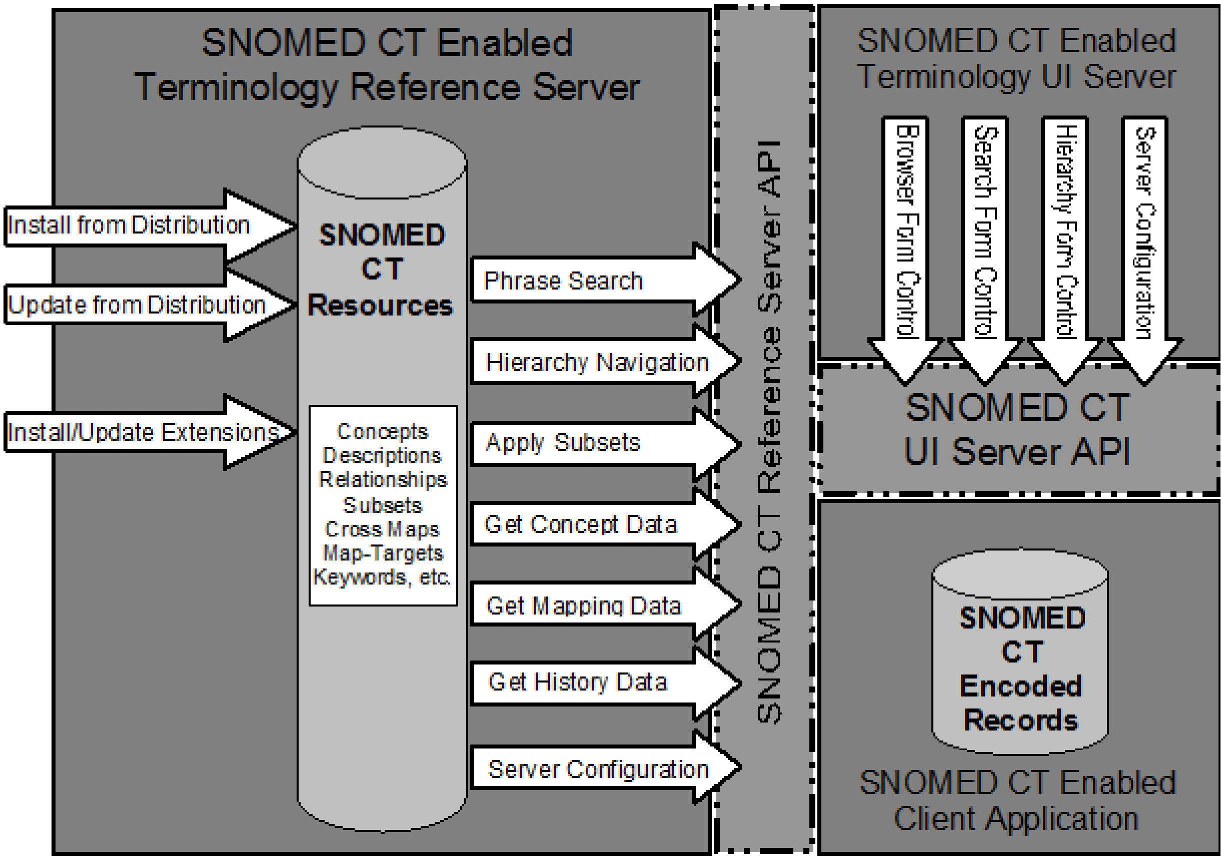


Figure 133: Terminology server providing user-interface and reference services to a client application

7.10.2 Terminology server APIs

This guide does not specify a particular *API*. The services specified in this guide may be delivered using various types of interfaces based on a range of different technologies including:

- Web services such using WSDL (Web Services *Description Language*) or REST (Representational State Transfer) interfaces;
- Java *components* such as JavaBeans™ or Eclipse plug-ins;
- Microsoft .NET® or Active -X® / COM / DCOM in Microsoft Windows® environments;
- CORBA® (Common Object Request Broker Architecture).

Decisions on which technologies to support depend on the intended functionality, performance, accessibility, ease of use and support requirements for maintenance or updates.

Over the past two decades there have been various efforts to specify standards for *terminology servers* and related *APIs*. The most recent development in this area is centered around the Common *Terminology Server Release* 2 (*CTS2*). The requirements initially identified and documented within *HL7* have now led to an OMG (Object Management Group) proposal. At least one of the responses to this proposal focuses directly on *SNOMED CT* related requirements. The OMG process is expected to result in a detailed specification and prototype implementation during 2011.

Chapter 8

8 Record Services Guide

The following sections discuss requirements for *record services* that support entry, storage, retrieval and communication of *SNOMED CT* encoded information. The services are illustrated by *Figure 134*.

The primary use of *SNOMED CT* is to enable information to be entered in a health record and stored in a manner that enables selective retrieval. Effective selective retrieval is required to support aggregation, analysis and decision support. Information in a health record may need to be communicated in the interests of the patient or to enable larger scale aggregation and analysis. Communication of information should convey the information expressed using *SNOMED CT expressions* in ways that preserve the semantics and thus enable recipient systems to process the information effectively.

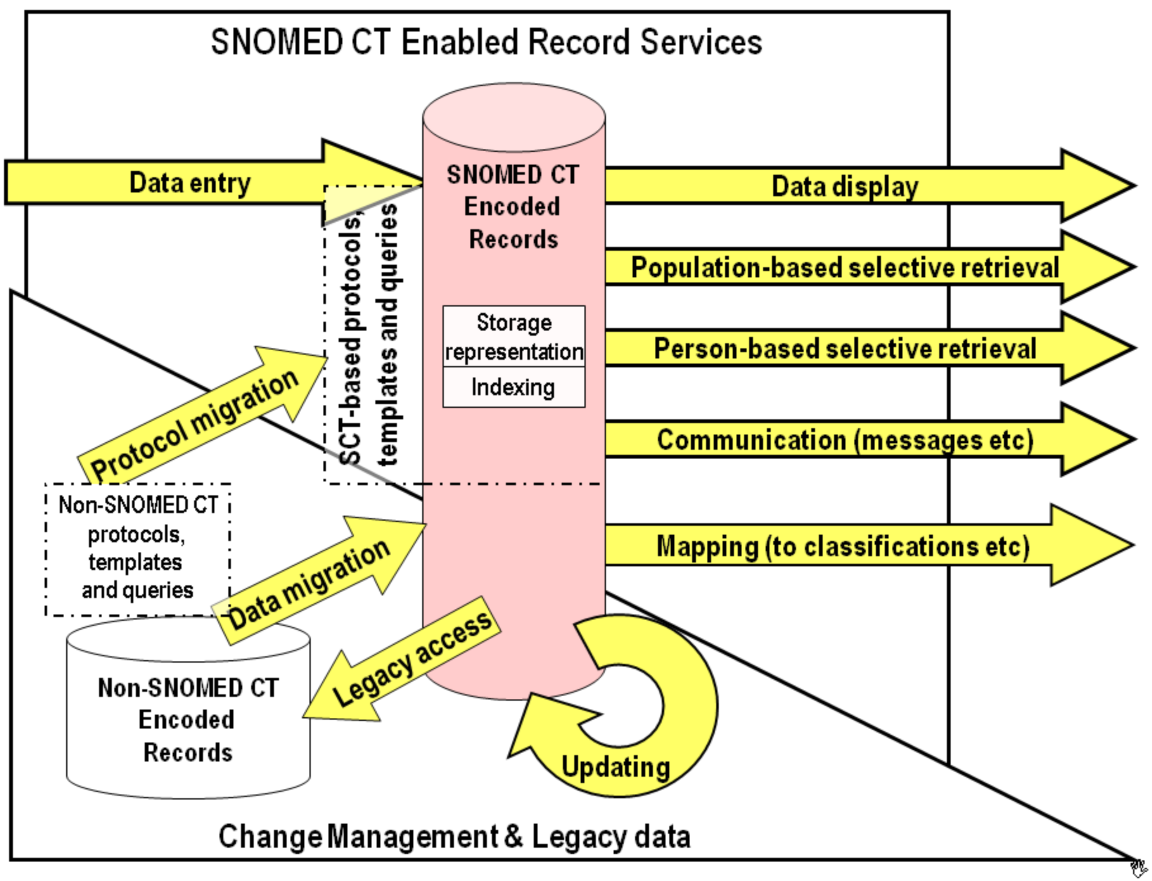


Figure 134: SNOMED CT Enabled Record Services

8.1 Entering Expressions

SNOMED CT enabled applications must facilitate the entry of *SNOMED CT expressions* in ways that allow users to capture relevant information easily and accurately. This section considers various methods that may be used to enter *SNOMED CT* information into a record. These data entry methods require the *Terminology services* specified in the *Terminology Services Guide (7)*.

8.1.1 Using text searches and subtype hierarchy navigation

8.1.1.1 Selection in a browser

The starting point for a consideration of data entry is an efficient method for performing text searches and *subtype hierarchy navigation*. When these functions are integrated in a terminology *browser*, it is possible to select a *Concept* by text search and then to refine or generalize the selection to identify a more appropriate *Concept* for recording.

A terminology *browser* built into an application or offering a programmable interface can be used to allow a user to select a *Concept* (and/or *Description*) and enter this into a record.

This method of data entry allows unconstrained selection of any *Concept* from *SNOMED CT*. This can sometimes be useful but such an unrestricted method should only be used as a fallback, when more selective approaches cannot be used.

8.1.1.2 Limitations of simple browsers

A general-purpose *browser* capable of searching and navigating through the *SNOMED CT hierarchy* is a simple starting point. However, this approach is unlikely to meet the requirements for anything other than occasional entry of *SNOMED CT* encoded information. More selective mechanisms tailored to particular data entry contexts are likely to be more usable and may promote more consistent data recording.

In most situations in which clinical data is entered, access to the full content of *SNOMED CT* through a simple search and *hierarchy browser* is unlikely to be necessary and may be cumbersome and unhelpful. The main reason for this relates to the size and structure of the terminology. As a result:

- Many *terms* may match a single word or short phrase resulting in a long list of options;
- The depth and breadth of the *subtype hierarchy* and *navigation* may require selection of choices from several screens to locate the required *Concept*.

There are many ways to improve and simplify *SNOMED CT* data entry. Some of these can be used in a wide range of situations. Others are specific to constrained contexts that occur in structured data entry driven by a template or protocol.

8.1.2 Optimizing searches for data entry

8.1.2.1 Extending searches and limiting duplication

The *Terminology services* guide addresses ways of:

- Extending text searches to include similar words and phrases by making use of the *Word Equivalents Table*;
- Rationalizing text searches, which, in a simple search, return the same *Concept* more than once due to multiple matching *Terms*.

These techniques may be used to improve access to *Concepts* during data entry.

8.1.2.2 Searches with qualifier resolution

When typing text for a search, the user is unlikely to know if their intended entry can be represented by a single *Concept* or requires a *postcoordinated expression* involving additional *Concepts* or *qualifiers*. Where searches fail to find a *precoordinated* match, expansion of the search to support appropriate or commonly used *qualifiers* is likely to enhance usability.

Some *terminology servers* may provide a general facility of this type. Alternatively, a limited facility for recognizing commonly qualifying words may be used. For example, words such as "left," "right," "routine," I emergency I and I severe I are applicable as *qualifiers* when not included in a *precoordinated Concept*.

8.1.2.3 Real time searching

Conventional text searches require the user to decide how many words to enter and then explicitly request a search. When a search fails to find any matches or returns a very long list of matches, the user is obliged to repeat the process. The need to undertake this type of user interaction for every coded entry is likely to create a significant disincentive to effective data entry.

One possible solution to this is an interface that performs real - time checking of the number of matches as the user types. The interface may indicate this to the user, allowing them to decide when to stop typing and commence the search. A further enhancement is to automatically return the list of matches whenever the user stops typing, or when the number of matches reduces to an acceptable level.

8.1.2.4 Background encoding

Techniques that support real-time searches and *qualifier* resolution may also be extended to enable background encoding of complete sentences as they are entered. This method can be applied to text entered by typing or by voice recognition.

As text is entered, the search mechanism attempts to narrow the selection. If this process eventually finds a single good match, this is used to encode the text. The match should be displayed allowing the user to override it, but the default action is to accept the encoding. If at the end of a sentence there are multiple possible matches, then these are presented for user selection.

There are many possible variants on this technique. For example, as the possible matches are narrowed down, the system could offer an auto-completion option similar to that used in web browsers and word-processors.

Caution: Anyone implementing this approach should take care to undertake appropriate quality assurance of the results. Mention of this approach to data entry does not imply that it is considered safe for a given use-case. Formal professional assessment of the risks and benefits of any type of automated encoding is essential.

8.1.2.5 Automatic and semi-automatic encoding

Techniques similar to those used for background encoding can be applied to previously entered text or to text entered by voice recognition or optical character recognition. Where such methods are used there is likely to be a need for manual intervention to resolve uncertain encoding. The requirement for manual intervention will depend on the sophistication of the matching techniques and the extent to which accuracy is safety-critical. If encoded data is to be used by clinical decision support protocols, which may influence the treatment of a patient, extreme care is needed when using automatic encoding and tools that allow manual review are essential. A less rigorous approach may be acceptable where the purpose of encoding is for aggregation and analysis of large volumes of population data.

Caution: Anyone implementing this approach should take care to undertake appropriate quality assurance of the results. Mention of this approach to data entry does not imply that it is considered safe for a given use-case. Formal professional assessment of the risks and benefits of any type of automated encoding is essential.

8.1.2.6 Mnemonics and personal favorites

Groups of people, such as practitioners of a discipline or specialty, frequently use similar sets of *Descriptions* and *Concepts*. Lists of widely understood (or easily learned) abbreviations or mnemonics that allow rapid entry of these commonly used *concepts* are recommended as a way of accelerating repetitive recording.

A similar facility may also be useful for individual users or organizations that have sets of *Descriptions* and *Concepts* that they use frequently. An easy way to use options to store and recall personal favorites with user-defined abbreviated access *terms* will enhance usability and significantly increase the speed of data entry.

User guidance may be necessary to minimize the risk of shortcuts such as these being overused. Unless the general search facilities are also easy to use, it is likely that users willfavor the shortcuts even when it would be more appropriate to use a more accurate but less accessible *Concept*. An unchecked bias toward easy to record *Concepts* may lead to deterioration in data quality, statistical anomalies, and in the worst case, inappropriate treatment.

8.1.3 Constraining searches for data entry

8.1.3.1 Constraining searches by status

In *Release Format 2* the value of the *active* field determines whether a *concept* or *description* is intended for active use. Searches should usually be filtered to exclude terms associated with *Inactive Descriptions* or any *descriptions* applied to an *Inactive Concepts*.

There are a few cases where a user may legitimately wish to search *Inactive Concepts* and *Descriptions*. Possible reasons for this include creating or editing queries that locate previously entered data recorded using *Concepts* and *Descriptions* that are no longer recommended for *active* use. Therefore, it should be possible to disable the *active* field search filters.

8.1.3.2 Constraining searches by subtype ancestors

Searches may usefully be limited to *Concepts* that have a specified *supertype ancestor*, which is appropriate for the context of a particular field, template or protocol.

Example:

When attempting to record the diagnosis "renal calculus," it is not helpful for a search to include the procedures that may be carried out to treat a renal calculus.

8.1.3.3 Constraining searches by Reference Sets

Searches for *Descriptions* or *Concepts* may need to be constrained by *Reference Sets*. Applications should allow searches to be filtered, ordered or otherwise prioritized in accord with one or more *active Reference Sets*. Specifically, the search mechanism should support the following functions with respect to the following types of *Reference Sets*:

- Filtering of search and *navigation* results to include only those *Descriptions* that are referenced by the Language *Reference Sets* may be applied to limit a search to those *Descriptions* applicable in a particular *language* or *dialect*.
- A Simple *Reference Set* may be used to filter, sort or highlight the results of text search or hierarchical *navigation*. This may simplify or encourage selection of *Concepts* or *Descriptions* used in a particular country, organization or specialty.
- A Simple *Reference Set* or an Ordered *Reference Set* may be used to specify or *order* the valid *Concepts* for entry in a particular field.

8.1.4 Constraining and extending hierarchical navigation for data entry

8.1.4.1 Using the subtype hierarchy for data entry

The most visible hierarchical construct in *SNOMED CT* is the *subtype hierarchy*. This is constructed using a set of logical rules. The purpose of this *hierarchy* is to support data retrieval and aggregation by addressing the question "is *concept* -A a *subtype* of *concept* -B."

The same *hierarchy* can be used for data entry *navigation* but it is not designed for this purpose. Its depth and breadth are determined by logical rules of subsumption rather than by usability. As a result:

- There is no upper limit on the number of *subtypes* a *Concept* may have. This is true because there is no rule that determines the number of *subtypes* that a real world *concept* may have. However, long lists of options are not conducive to effective data entry.
- There is no fixed limit to the number of hierarchical steps between a generalized *Concept* and its most refined *subtype*. This is true since there is no preordained limit on the extent of possible *refinement* of a real world *concept*. However, data entry procedures that involve stepping through several levels of choices before reaching the required selection impair usability.
- The *subtypes* of a *Concept* do not have any particular *order*. The | is a | *Relationship* is primarily a property of the *subtype Concept* and does not express an ordinal position. This is true because logical *subtypes* are inherently an unordered set. However, a user is likely to find it easier to locate their required selection if members of hierarchical lists are displayed in some recognizable *order*.
- The issues of depth, length and *order* noted above are also subject to change between releases. The addition of an intermediate *Concept* or reclassification after the addition of new *defining characteristics* will introduce new layers in the *hierarchy*. Some *Concepts* will then move from the list of immediate *subtypes* of a *Concept* to become *subtypes* of a more refined *Concept*. Hierarchical changes may sometimes simplify *navigation* by reducing the number of choices at a given hierarchical level. However, the general effect of improvements in the *subtype hierarchy* will be to increase its depth and thus to increase the number of steps from a particular general *Concept* to its most refined *subtypes*.
- The nature of a *subtype hierarchy* means that there may be many routes from a given *Concept* to its more general *descendants*. This means that some of the choices presented for user selection are redundant since they simply offer alternative routes to the same *Concept*.

Routine use of *subtype hierarchy navigation* is not recommended for data entry. However, despite the drawbacks listed above, the *subtype hierarchy* may be useful for undertaking an exhaustive search for a particular refined *Concept*.

Example:

The *Concept* "Laparoscopic emergency appendectomy" can be reliably located by *subtype navigation* from any of its supertypes: "appendectomy," "laparoscopic appendectomy" or "emergency appendectomy."

8.1.4.2 Using Ordered Reference Sets to support data entry

Ordered *Reference Sets* provide alternative hierarchical representations of *SNOMED CT*. They are intended to support data entry by addressing the limitations of the *subtype hierarchy* discussed in the previous section.

- Usability *constraints* can be placed on the number of levels in the *hierarchy* and the number of options displayed at each level in the navigational *hierarchy*:
  - If there are relatively few options and many layers, the most common options can be brought to a higher level.
  - If there are long lists of options, these may be subdivided with less frequent options moved to lower levels.

- Options that are rarely or never used by a particular user community can be excluded from a navigational *hierarchy* to limit the range of choices. According to requirements, these options may remain accessible by switching to a *subtype* view.

- Options at each hierarchical level can be ordered to meet the expectations of users and/or to facilitate rapid access to commonly used options.
- The available options at a particular level can be kept stable across releases without affecting the accuracy of the *subtype hierarchy*.

An Ordered *Reference Set* may be based on the foundation of the *SNOMED CT subtype hierarchy*. This can then be modified to add ordering and other features discussed above. An alternative starting point is a *hierarchy* of classification derived from another coding scheme or classification.

Note:

An Ordered *Reference Set* derived from the *Clinical Terms Version 3 hierarchy* is provided with the *SNOMED CT* Developer Toolkit as an example.

Alternative Ordered *Reference Set* can be created from scratch (or as variants of a common source *hierarchy*) to provide views to support users with different requirements. Since *Navigational Hierarchies* do not affect interpretation, retrieval and aggregation, data entered in using different views can be analyzed consistently.

8.1.5 Constraining data entry

Some *Concepts* or *Descriptions* displayed by searches, hierarchical *navigation* or other methods of data entry may not be suitable for recording in a patient record. Various reasons for this are discussed in the following sections. They include:

- Status of the *Concept*
- Status of the *Description*
- Special *Concept*
- *Subtype* relevance;
- *Reference Set* inclusion;
- Mappability;
- Context.

8.1.5.1 Excluding inactive concept and descriptions

In *Release Format 2* the value of the *active* field determines whether a *concept* or *description* is intended for active use. *Inactive Concepts* should not be added to a record. Similarly, terms associated with *Inactive Descriptions* or any *descriptions* applied to an *Inactive Concepts* not be added to a record.

8.1.5.2 Excluding Special Concepts and Model Components

Concepts that are *subtypes* of the top-level *Concept* 370115009 | Special concept | (*RF1*) or 900000000000441003 | SNOMED CT Model Component | (*RF2*) are rarely if ever required in clinical end-user searches. Therefore, they should be excluded from text searches except where explicitly needed to meet a particular requirement (e.g. to display a | Namespace concept |, a | Linkage concept | or a | Navigational concept |).

8.1.5.3 Constraining data entry according to subtype relevance

It may be necessary to prevent entry of a *Concept* in a *subtype hierarchy* that is inappropriate to a particular data entry field or to a particular part of a patient record. For example:

- An application should not allow a disorder *Concept* to be recorded in a field intended for recording a procedure (or vice-versa);

- An application should not allow a *Concept* that is a *subtype descendant* of the top-level *Concept* "attribute" to be recorded, except to associate another *Concept* with an appropriate qualifying value;
- An application should not allow a *Concept* that is a *subtype descendant* of the top-level *Concept* "*qualifier* value" to be recorded, except where it qualifies an appropriate attribute *Concept*.

8.1.5.4 Constraining data entry using Reference Sets

In some cases, identifying selected portions of the *SNOMED CT hierarchy* may be a sufficient *constraint* for entering data into a record. However, that is not always sufficient if *Concepts* from multiple hierarchies are required, or if there is a need to hone down the entry options from the full *hierarchy*. To meet these requirements applications should allow data entry to be constrained by *Reference Sets*.

Applications should be able to:

- Permit or prevent the entry of *Concepts* or *Descriptions* that are members of a specified Simple *Reference Set*.

Example:

A UK GP system might:

- Prevent the entry of *Concepts* in a Simple *Reference Set* that contains all *Concepts* that are non-human;
- Enable the entry of *Concepts* in the "UK Administrative *Reference Set*" only when entering information in an administrative context.

- Encourage or inhibit the entry of *Concepts* or *Descriptions* according to their order in an Ordered *Reference Set*.

Example:

A specialty system might prompt for confirmation when the user records a procedure not in a specified specialty Simple *Reference Set*.

8.1.5.5 Constraining data entry based on mappability

One of the requirements for some applications may be that the data recorded in particular fields has to be mapped to a particular classification or grouping scheme.

One way to simplify this process is for the application to check mappability at the time of data entry. If a selected *Concept* has no unambiguous map, the application may encourage or compel the user to refine their selection until a mappable *Concept* has been selected.

This type of facility should not be applied in situations where it may inappropriately affect the perceived accuracy or detail of a clinical record.

8.1.5.6 Constraining data entry based on context

All fields (data elements) used for data entry must be analyzed to understand what underlying context is implied. The appropriate *concepts* should then be selected for the *value set* of each field. *Concepts* from the Clinical Findings, Procedures, and Observable entities hierarchies can be used directly if the default assumptions are true. Otherwise, *concepts* from the *Concept* -Dependent *hierarchy* should be selected.

Particular care must be taken with systems that enable *postcoordinated* constructs to ensure that the appropriate context attributes are included.

To precoordinate *concepts* that do not already exist in *SNOMED CT*, care must also be taken to determine if any axis modification is shifting the meaning of a *concept* so it should move to the *Situation with explicit context hierarchy*.

8.1.5.7 Absolute and configurable constraints

Some of the *constraints* on data entry discussed in the preceding sections are absolute while others should be configurable.

- An application should not allow *Inactive Concepts* or any Special *Concepts* to be recorded.

Constraints based on *subtype* hierarchies, *Subsets* and other *Reference Sets* including *maps* to other terminologies or code systems should usually be configurable to particular institutions, users and/or data entry fields.

8.1.6 Configuring and applying data entry constraints

The previous sections describe various mechanisms for extending and constraining search and *navigation* during data entry. The scope of applicability of these facilities varies and these variations affect the way in which they may be implemented.

A few *constraints* apply to all data entry events in a particular application. These fixed *constraints* could be hard-coded in the application or explicitly optimized when importing and indexing *SNOMED CT* content.

Example:

One example is to exclude *Inactive Concepts* and *Descriptions* from searches. Before building this type of facility into an application, care should be taken to consider circumstances, such as creation and editing of queries where access to *Inactive Concepts* may be required.

Most search *constraints* are to some extent configurable and these require greater flexibility in the application design. There are several types of configurability that may be required. These range from installation configuration to context-specific dynamic configuration.

8.1.6.1 Installation configuration of data entry

Requirements of an organization that are general to all users may be applied when installing the application or when importing or indexing *SNOMED CT* content. These may include:

- Language *Reference Set* which constrain searches to the local *language* and *dialect*.
- Simple *Reference Sets* which apply national or organization *constraints* applicable to all users of the application.
- Simple *Reference Sets* which apply *constraints* applicable to all the clinical disciplines or specialties that use the installed system. For example, installations that are not intended for use in veterinary medicine will apply *Reference Sets* that exclude specific veterinary *Concepts* and *Descriptions*.
- An Ordered *Reference Set* that provides a data entry *hierarchy* appropriate to the needs of all users within an organization .

8.1.6.2 Log-on configuration of data entry

The application should allow search *constraints* that are specific to a particular user or group of users when loading or logging on to an application. The range of possible search *constraints* may be preset at installation but it should be possible to apply the user profile *constraints* without a significant delay. Uses of this type of configuration include:

- Simple *Reference Sets* which apply *constraints* or optimizations applicable to a particular specialty;
- Ordered *Reference Set* that provide a restricted or extended data entry *hierarchy* appropriate to the needs or preferences of a particular specialty or user;
- Language *Reference Set* that meet the needs of particular users in a multi-lingual environment.

Consideration should be given to requirements for this type of search configuration to be modified by a user or system administrator.

8.1.6.3 Dynamic reconfiguration of data entry

Constraints that assist fast and consistent routine data entry may sometimes need to be relaxed to enable more complex entries to be made.

- If a Ordered *Reference Set* limits the scope of hierarchical *navigation*, the application should enable the user to utilize the *subtype hierarchy* to allow other options or a more complete set of options to be reviewed;
- If a user is unable to locate the *Concept* that they require, it may be useful to enable some or all of the search *constraints* to be temporarily lifted.

8.1.6.4 Context-sensitive of data entry constraints

Some *constraints* may apply to particular data entry contexts. To support this type of functionality, an application should be able to switch between sets of search *constraints* in real - time. The *constraints* need to change instantly as a user moves between different data entry fields. Context-dependent *constraints* may include:

- Limitation of a search to *subtype ancestors* of an appropriate *Concept*.

Example:

A field for entry of a procedure may be associated with a *constraint* that limits searches to *subtypes* of the *Concept* "procedure."

- Limitation of a search to the *Concepts* or *Descriptions* that are members of an appropriate Simple *Reference Set*.

Example:

A field for entry of a laboratory service request may constrain searches to a list of valid investigations supported by a particular laboratory.

- Use of a particular Ordered *Reference Set* or an specified sub-branch of an Ordered *Reference Set*:
    - This is an alternative approach that may be used to allow more sophisticated control of data entry in a particular context.

8.1.7 Entering qualifiers and other postcoordinated representations

SNOMED CT contains many *precoordinated Concepts* that allow fairly complex *Concepts* to be represented by a single *Concept Identifier*. It also permits the qualification or *refinement* of *Concepts* to represent more detailed *Concepts* by *postcoordinated* combinations of several *Concept Identifiers*.

Several types of *postcoordinated* data are outlined in this section from the perspective of data entry. These include *refinement*, qualification and combination. The requirements for and relevance of each of these will depend on decisions about data representation within patient records.

8.1.7.1 Entering refined defining characteristics

The application may allow a user to refine a *Concept* by selecting a *subtype* of one of its *defining characteristics*.

Example:

One of the *defining characteristics* the *Concept* | total replacement of hip | is | using | = | hip prosthesis. | The *Concept* | total replacement of hip | could be refined by allowing the user to specify one of the *subtypes* of "hip prosthesis."

Refinement options may be entered by selecting from hierarchical lists showing *subtype* values for each of the refinable characteristics. Simple lists or option buttons could support selection from limited sets of possible *refinements*. Wider ranges of potential *refinement* could be facilitated by text searches constrained to *subtypes* of one or more of the refinable characteristics.

Caution:

Some concepts should not be refined if the result means the new concept is not a subtype of the parent concept.

This situation occurs when context such as "Family history" or | Planned Procedure | is attached to a Clinical Finding or Procedure. "Family history" of a Clinical Finding needs to be defined in the *situation with explicit context hierarchy*. All *postcoordinated* constructs should consider the impact of context.

8.1.7.2 Entering concept model refinements

The application should allow a user to refine the meaning of a *concept* by selecting an *attribute* permitted by the *concept model* and applying an appropriate value to that *attribute*. The *attributes* permitted for a concept depend on the *domain* in which that *concept* falls (i.e. its position in the *subtype hierarchy*). Similarly, the set of values that can be applied to an *attribute range* specified for that *attribute*.

8.1.7.3 Entry of unsanctioned qualifiers

The application may also permit *refinement* of a *concept* by the addition of *attributes* and values that are not sanctioned by the *concept model*. However, this is not recommended as it results is inconsistency between representation of meaning in different systems. Any facility to allow qualification of *concepts* in this way carries a risk of creating nonsensical or contradictory statements. It may also result in incomplete or inappropriate retrieval where the *qualifier* significantly affects the meaning of the *concept*.

8.1.7.4 Constraints on the entry of refinements

Refinements should only be used where the result of applying them results in a true *subtype* of the original *Concept*. Therefore, refinement should **not** be used for the following purposes:

- Negation.

Example:

| Fracture of humerus | should not be qualified by "excluded."

It would be inappropriate for data retrieval to treat this as a *subtype* of the clinical finding | Fracture of humerus |.

- Certainty.

Example:

| Carcinoma of cervix | should not be qualified by "possible."

It would be inappropriate for data retrieval to treat this as a *subtype* of the diagnosis of | Carcinoma of cervix |.

- Subject of information.

Example:

| Diabetes mellitus | should not be qualified by "family history."

It would be inappropriate for data retrieval to treat this as a *subtype* of the diagnosis of | Diabetes mellitus | in the patient.

- Planning stage.

Example:

"Hip replacement" should not be qualified by "planned" or "requested."

A count of "Hip replacement" operations performed should not include this. Decision support protocols should not assume the patient has had this operation.

These and similar major modifications need to be handled in ways that are explicit and ensure that queries and decision support protocols are able to accurately retrieve and analyze the available information.

8.1.7.5 Entry of concepts combinations

The application may allow other combinations of *Concepts* in a single statement where a *Concept* that represents the full scope of an activity is not available. This approach might for example be applied where a single procedure, which lacks a *precoordinated SNOMED CT* representation, is a combination of two procedures that can be separately represented in *SNOMED CT*.

Facilities for entering combined *Concepts* should be implemented and used with care. It is appropriate to use these facilities when the combined result is conceived as a single statement that could potentially be used in many different patient records.

Example:

A diagnosis of "gallstones with cholecystitis" could be entered by selecting the "gallstone (disorder)"235919008and then selecting | cholecystitis (disorder) |76581006and combining these in a single statement [34].

It is not appropriate to use these constructs to attempt to express an entire encounter, episode or clinical history in a single statement.

Example:

If a patient is treated for "gallstones with cholecystitis" diagnosed by "ultrasonography of biliary tract" with a course of | amoxycillin | followed after the acute phase has resolved by a | cholecystectomy |, this should **not** be entered as a single complex *postcoordinated* statement combining the diagnosis, investigation and treatments.

8.1.8 Template and protocol driven data entry

In many healthcare disciplines similar data sets are collected for each patient. Clinical consultations for many conditions involve repeatable sequences of data entry. These structured and predictable data entry requirements can be met using sets of customized data entry fields or forms (templates) designed to collect particular data items. These data entry templates may be presented in a predefined sequence, as selected by the user. Alternatively the sequence of data entry may follow a branching pathway with previously entered data determining which branches are taken (protocols).

When using a structured data entry mechanism, *SNOMED CT* encoded data can be selected in a variety of ways. Some of these involve direct selection of *Concepts* and *Descriptions* while in others the encoding may result from responses to simple choices or entry of particular data values. The following list outlines some of the possible mechanisms for *SNOMED CT* encoding during structured data entry:

- User selection from a small list of possible *Descriptions* applicable to a particular field in a template or step in a protocol:
  - A Simple *Reference Set* with *Descriptions* as members or a Language *Reference Set* may specify the set of applicable *Descriptions*.
- Text search limited to a set of *Concepts* applicable to a particular field in a template or step in a protocol:

[34] There is also a precoordinated Concept "Calculus of gallbladder with cholecystitis" which is equivalent to this postcoordinated combination.

- A Simple *Reference Set* may specify the set of applicable *Concepts*;
- Alternatively the applicable *Concepts* may be specified as the *subtype descendants* of a single *Concept*.

- Association of a *Concept* with particular options presented by a check box, option button or other data entry control:
  - When selections are made using this control the appropriate *Concept Identifier* is added to the record.
- Association of a *Concept* with a data control used for entering a numeric or other value:
  - When a value is entered in this control it is labeled with the appropriated *Concept Identifier*.
- Association of a *Concept* with a particular combination of values or the result of a computation involving several items of previously entered data:
  - In its simplest form this is an *extension* of one or both of the previous options;
  - In some applications, information derived from the user-entered data, by decision support tools, may be encoded in this way.

Some installations allow free text to be entered at point of care if a needed *concept* is not included in the predefined short list. This text is then reviewed by trained staff who can then search and find the appropriate *Concept*, request the addition of a new *Concept*, and/or request that the *Concept* be added to the template's short list for future use. The success of this option relies upon trained staff who are available to do the review on a timely schedule, and the willingness of the clinician to use this approach sparingly, as it is greatly preferred to choose the appropriate *concept* and not enter free text.

8.2 Storing Expressions

SNOMED CT enabled applications must support the storage of *SNOMED CT expressions* in ways that represent relevant information within the record system. This section is concerned with the different approaches that may be used to store *expressions* in ways that enable them to be subsequently retrieved, displayed, processed and communicated.

The term *SNOMED CT expressions* includes also single *Concept Identifier expressions*, that identify only one specific *concept*. Even when *postcoordination* is not supported in an implementation, each time a single *Concept Identifier* is being assigned to a clinical record it represents a way of using a *SNOMED CT expression*.

8.2.1 precoordinated and postcoordinated representations

8.2.1.1 precoordination

The simplest form in which any *concept* can be stored is as a single *Identifier*. This is referred to as a *precoordinated expression*, because all aspects of a potentially multifaceted *concept* are *precoordinated* into a single discreet form.

SNOMED CT contains more than a quarter of a million *concepts*, and thus allows a wide range of clinical statements to be expressed in *precoordinated* form.

Example: Laparoscopic emergency appendectomy - *precoordinated*

A *precoordinated expression* 174041007 | laparoscopic emergency appendectomy | can be used to record an instance of this procedure.

The procedure "Laparoscopic emergency appendectomy" has at least three distinct facets: "removal of appendix","using a laparoscope" and "as an emergency procedure". *SNOMED CT* includes a *concept* that precoordinates these facets.

The *concept* 174041007 | laparoscopic emergency appendectomy | has the following *defining characteristics*:

260870009 | priority | = 25876001 | emergency |,

116680003 | is a |=80146002 | appendectomy |,[35]

425391005 | using access device | = 86174004 | laparoscope |.

8.2.1.2 postcoordination

A multi-faceted *concept* can be stored using a combination of *Identifiers* for its individual facets. This is referred to as *postcoordination*, because the various aspects of the *concept* are coordinated during data entry rather than in the preparation of the terminology. Three types of *postcoordination* are described in the following sections.

8.2.1.3 postcoordination by refinement

Refinement is a type of *postcoordination* in which a *concept* is made more specific by refining the value of one or more of the defining attributes of the *concept*.

Example: Total replacement of hip using a Sheehan total hip prosthesis - *postcoordinated*

A *postcoordinated expression* based on the *concept* 52734007 | total hip replacement | can be used to record an instance of this procedure. The definition of this *concept* includes 363699004 | direct device |=304120007 | total hip replacement prosthesis | and the value of this attribute can be refined to 314580008 | Sheehan total hip prosthesis | (which is a *subtype* of 304120007 | total hip replacement prosthesis |). Therefore, the following *postcoordinated expression* can be created and used to represent this procedure:

52734007 | total hip replacement | : 363699004 | direct device |=314580008 | Sheehan total hip prosthesis |.

Another common use of refinement is to represent a situation such as a family history, or a planned procedure. In this case, a *concept* representing the general type of situation can be refined by applying a clinical finding or procedure.

Example: Family history of temporal arteritis - *postcoordinated*

A *postcoordinated expression* based on the *concept* 281666001 | family history of disorder | can be used to record a family history of any disorder. The definition of this *concept* includes 246090004 | associated finding |=64572001 | disease | and the value of this attribute can be refined to 400130008 | temporal arteritis | (which is a *subtype* of 64572001 | disease |). Therefore, the following *postcoordinated expression* can be created and used to represent this family history:

281666001 | family history of disorder | : 246090004 | associated finding |=400130008 | temporal arteritis |.

8.2.1.4 Postcoordination by qualification

Qualification is a type of *postcoordination* in which a *concept* is made more specific by applying value to attributes that are permitted by the *Concept Model*. Unlike refinement, the attributes applied need not be present in the definition of the *concept* that is being qualified.

Example: Laparoscopic emergency appendectomy - *postcoordinated*

A *postcoordinated expression* based on the *concept* 80146002 | appendectomy | can be used to record an instance of this procedure by separately specifying the access instrument and priority. The *concept*

[35] In practice the relationship 116680003 | is a |=80146002 | appendectomy | is represented via intermediate supertype and is also represented by the following defining characteristics 260686004 | method | , 405813007 | procedure site - Direct | = 66754008 | appendix structure |.

80146002 | appendectomy | does not have defined values for the attributes 260870009 | priority | and 425391005 | using access device | but the *Concept Model* permits these to be added to *subtypes* of 71388002 | procedure |. Therefore, the following *postcoordinated expression* can be created:

80146002 | appendectomy |:260870009 | priority | = 25876001 | emergency | , 425391005 | using access device |= 86174004 | laparoscope |

This *postcoordinated expression* is equivalent to the definition of the *concept* 174041007 | laparoscopic emergency appendectomy |. However, the *postcoordinated* approach can also be applied to procedures for which there is no *precoordinated concept*.

8.2.1.5 postcoordination by combination

Example:

"Gallstones with cholecystitis" could be represented by combining the *concepts* for the disorders "gallstones" and | cholecystitis | as a single *postcoordinated* statement. Neither of these *concepts* is really a *qualifier* of the other since it could equally well be regarded as | Calculus of gallbladder with cholecystitis |.

SNOMED CT allows *Concepts* to be combined in *postcoordinated* statement.

Combinations like this should only be used to represent *concepts* that can be regarded as discreet reusable clinical statements. They should not be used to construct arbitrarily complex representations of multiple statements to a particular record.

Some *concepts*, such as the first and last examples above, can be represented in either a *postcoordinated* or *precoordinated* form. However, there are other *concepts*, like the second example above, for which no *precoordinated Concept* exists in *SNOMED CT*. Although future releases of *SNOMED CT* will include new *precoordinated Concepts*, there will always be some clinical *Concepts* that require *postcoordination*.

8.2.1.6 Representing postcoordination

This guide does not specify a single right way to represent *postcoordinated expressions*. Alternative representations have different profiles of advantages and disadvantages. The choice of representation depends on functional requirements including performance, information model of the software application and the communication standards to be supported.

Some alternative representations are summarized below. These summaries illustrate some of the main options and do not go into extensive technical detail. Detailed design may lead to further alternatives that are not documented here.

Each of the following summaries assumes that *SNOMED CT expressions* are stored in (or associated with) one or more fields within particular types of record entry. The *expression* is only one part of the data in that record entry.

8.2.1.6.1 Parsable text representation

A way to represent *postcoordinated SNOMED CT* information as a simple parsable text *string* is summarized below:

- Each clinical statement is recorded as a row in a relational database table (or as an element in an XML document);
- The schema for representation of clinical statements contains a field (or element) for representation of the *SNOMED CT expression*;
- The *expression* field (or element) contains a text *string* that is formatted in accordance with the *SNOMED CT compositional grammar*.

8.2.1.6.2 Unrestricted relational representation

An unrestricted relational database representation of a *postcoordinated expression* requires that a data item that may be expressed using *SNOMED CT* is modeled in a way that permits an indeterminate number of *attribute-value pairs* to be appended to a *focus concept*. In addition, the value within each *attribute-value pair* must be able to be refined by addition of nested *attribute-value pairs*.

This offers a flexible and extensible approach but adds significantly to database design complexity. Disadvantages arising from this complexity include storage capacity requirements and the impact on writing queries and retrieval performance.

8.2.1.6.3 Restricted relational representation

An alternative restricted relational representation of *postcoordinated SNOMED CT* information is summarized below:

- Each clinical statement is recorded as a row in a relational table.
- The clinical statements table contains a field for a *Concept Identifier*.
- The clinical statements table also contains fields for a specified number of *qualifiers*. These fields may be provided in different ways:
    - Each *qualifier* is represented by two *Concept Identifier* fields (one for the attribute and one for the value) and an optional field for *Relationship group* field. With this option the only restriction is the total number of *qualifiers* or modifiers that can be stored for each *Concept*.
    - Each *qualifier* is represented as a single *Concept Identifier* and carries the value of a *qualifier* attribute specific to that field. This restricts the usable *qualifiers* to those specified in the database schema.
    - Similar to above, but with different sets of qualifying attributes available according to the semantic type of the primary *Concept* in the statement. There are various ways of implementing this approach to ensure that the appropriate interpretation is applied to each row of the table.
- Combined *Concepts* may be represented by explicitly combining two rows of the clinical statements table.

Unlike the representations discussed in previous subsections, this approach limits the expressivity of *postcoordinated* statements. The advantage of this restricted approach is that it reduces the number of joins involved in retrieval queries. In some software environments this may significantly improve performance.

The balance between demands for flexibility and performance depends on user requirements. Therefore, limitations in expressivity may be acceptable for some users or user communities but not for others. However, it should be noted that these limitations might cause difficulties when communications are received from systems that support richer forms of *expression*.

8.2.1.6.4 XML Representations

A way to represent *postcoordinated SNOMED CT* information as an XML element is summarized below:

- Each clinical statement is recorded as a row in a relational table or as an element in an XML representation.
- The clinical statements table (or element) contains a field (or element) for representation of the *concept*.
- The *concept* field (or element) contains an XML *expression* that encapsulates a *postcoordinated* representation of the *concept* according to a parsable syntax specified for this purpose:
    - Various alternative XML representations could fulfill this role.

8.2.1.6.5 Representation as precoordinated content

In some implementations, *expressions* are stored as *precoordinated* content, with new *concepts*, *Descriptions* and *Relationships* in an extension namespace.

User input includes also a text label for the *expression*, and the new *concept* is created, usually a team of expert *SNOMED CT modelers* review the new *concept* for quality assurance. Other implementations requires that user enter only the text label, and then the modelers team can associate the label to an existing *concept*, or create a new *concept* in a local extension using the label as a *Description* and adding the new *Relationships* for the *concept* definition.

This approach is called Managed Content Additions (MCA). Has some advantages like having all new content available for text searches by users, and allowing the use of a *description logics* classifier for inferring *Relationships* and super-types, avoiding the need of complex real-time *expressions* computations. On the other having a centralized team of experts represents an expensive approach and a possible bottleneck for terminology development, as the experts need to review all content additions in the system.

8.2.1.7 Storing and retaining original expressions

Transforming an *expression* to a *normal form* may be necessary to support effective data retrieval. However, even quite small minor corrections to the definition of a *concept* in future releases may significantly alter the resulting *normal form* of the same *expression*.

Therefore, it is recommended that:

- The primary or original record should be stored using the representation that is as close as possible to the form in which it was recorded.
- If *transformations* to alternative representations are used to enhance the efficiency of retrieval, these should be stored as secondary supporting tables or indices:
  - This has the advantage that these alternative forms can be regenerated based on the most up to date set of definitions when a new release of *SNOMED CT* is installed, without affecting the integrity of the original records.

8.2.2 SNOMED CT storage issues for electronic health records

8.2.2.1 Storing Concepts in electronic health records

Information in an *electronic health record* should accurately reflect the way it was recorded by its author. If the author of a statement in the clinical record chooses a particular form of representation the system should faithfully store the information in that form.

Following this principle, the recommended approach for representation of *SNOMED CT* in a *electronic health records* is as follows:

- If, during data entry, an author selects a single *precoordinated Concept* to represent a clinical statement, the *Identifier* of that *Concept* should be stored in the record:
  - This form of representation should remain as the original record of that statement. It should not be replaced by an apparently equivalent *postcoordinated transformation* of this *Concept*.
- If, during data entry, an author constructs a clinical statement by selecting a *Concept* and one ore more *qualifier* values, *refinements* or additional *Concepts*, the *Identifier* of all the relevant *Concepts* should be stored in the record in a manner that reflects the *relationships* between them:
  - This form of representation should remain as the original record of that statement. It should not be replaced by an apparently equivalent *transformation* of this *Concept* into a *precoordinated* or differently constructed *postcoordinated* form.

An application should prompt for author endorsement of any alternative form of representation that it proposes to store in the original *electronic health record*. In this case, if the author accepts the alternative form presented by the application, this form should be stored as the original record.

The forms in which a technical implementer may wish to store data for efficient retrieval may differ from the forms dictated by the principles appropriate to storage of original entries in a *electronic health record*. However, it is recommended that any retrieval- oriented representation should be derived from rather than replace the original form of the record.

8.2.2.2 Storing terms

A *electronic health record* should also store the *terms* that were actually displayed to and selected by the author of the record. In some *Realms* the *Description Identifier* may be regarded as an adequate proxy for the full representation of the associated *Term*. However, in other jurisdictions there may be a requirement to store the original text as entered or selected by the user.

Storing the *Description Identifier* has the added advantage if a *Description* is found to be wrongly associated with a particular *Concept* or if the associated *Concept* is found to have non-synonymous *Descriptions*. In these cases, the *Description Identifier* can be used to map the information to the appropriate disambiguated *Concept*.

8.2.2.3 Maintaining integrity following SNOMED CT releases

A *SNOMED CT release* may contain changes to that state of one or more *Concepts* or *Descriptions* referenced by a stored *expression*. The original recorded form of each stored *expression* should be retained as record of the information actually entered. However, it may also be useful to include updated representations that take account of changes to the referenced *SNOMED CT* content.

Release Format 2 files contain previous states of each component allowing comparisons to be performed. In addition, members of an appropriate | Historical association reference set | allow data originally recorded with a *Concept* that has been marked as *Inactive* to be mapped to an appropriate *Active Concepts*.

If clinical records are updated using this history information, the changes should be appended to the original representation, rather than replacing it. This ensures that any changes arising from a subsequent release can apply the improved mapping to the original *Concept* this can be utilized to enhance data quality.

Note: In *Release Format 1*, *SNOMED CT Component* History, Reference and *Relationship* tables contain information that allows data originally recorded using *Inactive Concepts* to be appropriately mapped to *Active Concepts*.

8.2.3 SNOMED CT storage options for effective retrieval

The form in which records are represented may have a substantial impact on the efficiency, accuracy and completeness of retrieval. The forms that best suit retrieval may differ from the forms that are required to meet the principles of clinically safe and legally valid *electronic health records*.

8.2.3.1 Storing information as entered

This option leaves information in the form entered in the *electronic health record* with no additions to assist future retrieval. The application must do all the work needed to locate the required records and compute subsumption and *equivalence* when a request is made to retrieve data.

8.2.3.2 Using an Expression Repository

An innovative approach to the issues raised by literal storage of *postcoordinated expressions* is to implement an *expression* repository. Each unique *expression* used in the system is stored in a referenced database table and assigned an internal unique *Identifier* (e.g. a *UUID*). When an *expression* is used in a clinical record entry the unique *expression* id is used to reference the *expression* in the repository.

The key advantages of this approach of this approach are:

- The *expression Identifier* can have a fixed size whereas a *postcoordinated expression* in of variable and indeterminate size. This significantly improves storage and index efficiency.
- The *expression* repository can also be used to store *normal form* representations of each *expression* and to relate these to the original *expression*. This optimizes performance for *expression* normalization during retrieval.
- The *expression* repository could also be processed by a *Description Logic Classifier* and a *transitive closure* table of all the *expression* used in the application could then be generated. *postcoordinated* retrieval would then be highly optimized by using the *transitive closure* to test a single join between each predicate and the candidate *expressions*.

8.2.3.3 Minimizing postcoordination

One possible approach to optimization of retrieval is to *transform* the original stored information into an equivalent representation with the minimum number of *postcoordinated* components.

The objective of this approach is to allow the generation of simple indices for the *precoordinated* representation. It is then possible to undertake most retrievals using the | is a | *subtype hierarchy* to compute whether *Concepts* in the record are *subtypes* of the *Concepts* used to specify retrieval. Where *postcoordination* is required, the minimum number of additional tests are required to confirm that a *Concept* in the record meets the specified retrieval criteria.

One difficulty with this approach is that there may be more than one representation that requires the same degree of *postcoordination*. This is discussed in more detail and illustrated in *Transforming expressions to normal forms*.

If this approach is adopted additional rules need to be applied to determine the choice between alternatives with a similar number of *postcoordinated* components.

Example:

In the hypothetical example illustrated in *Figure 30*, the *Concept* "red steel pedal bicycle", for which no *precoordinated* representation exists, could be represented as:

"red pedal bicycle" + | make of | = | steel |

or

"steel pedal bicycle" + "color " = "red"

Both are equally close to the objective of minimizing *postcoordination*. A rule is needed to determine which of these is preferred. There is no obvious right or wrong solution to this but a simple rule that places the attributes in an *order* will, if applied consistently, allow all *postcoordinated* representations to be reduced to a single minimized form.

8.2.3.4 Maximizing postcoordination

An alternative approach is to expand any *precoordinated concepts* in the record to their fullest possible *postcoordinated* forms. This general type of *transformation* is illustrated in *Transforming expressions to normal forms*.

This approach requires a richer record structure but has the advantage that there are three possible end-points to *postcoordination*, each of which ensures that any computably equivalent representations of *Concepts* will expand to an identical *postcoordinated* form. The three end-points are summarized here:

- Short *canonical form*:
  - This is the most parsimonious of the three options.
  - A *concept* is represented as the combination of:
    - *Subtype relationships* with its most proximate *primitive* supertypes;
    - The recorded *qualifier* values and/or *defining characteristics* that distinguish it from its most proximate *primitive* supertypes.
- Long *canonical form*:
  - This option is more verbose as it includes some redundancy.
  - A *concept* is represented as the combination of:
    - *Subtype relationships* with its most proximate *primitive* supertypes;
    - All of its recorded *qualifier* values and/or *defining characteristics*, irrespective of whether they are shared by its most proximate *primitive* supertypes.
- Exhaustive *postcoordinated* form:
  - This option is extremely verbose.
  - A *Concept* is represented as a combination of:

- *Subtype relationships* with all of its *supertype ancestors*
- All of its recorded *qualifier* values and/or *defining characteristics*, irrespective of whether they are shared by its most proximate *primitive* supertypes.

If the retrieval criteria are expressed in a similar form, a relatively simple *query* can interrogate the record for all entries with a matching set of *primitive Concepts* and specified characteristics.

8.2.4 Record architectures, structures and semantics

8.2.4.1 Record structure standards and proposals

SNOMED CT is a controlled terminology that can be used in many different health record systems. The semantic model of *SNOMED CT* does not replace the need for a logically sound health record structure. Furthermore, the *IHTSDO* does not specify a particular health record structure for use in conjunction with *SNOMED CT*. However *SNOMED CT* representations of clinical *concepts* are intended to meet the needs of standard health record architectures for a consistent controlled coded terminology.

In particular, there is a strong interest in co-evolution of *SNOMED CT* and the following standards to provide a strong standard semantic foundation for future *electronic health record* development.

- The healthcare communication and structured document standards of *HL7* (*www.hl7.org*). In particular:
  - The HL7 *Reference Information Model* and the associated development methodology;
  - *Release* 2 of the Clinical Document Architecture (CDA);
  - The *Guide to the Use of SNOMED CT with HL7 Version 3* developed by the *TermInfo* Project.
- The European (*CEN*) Standard for Electronic Healthcare Record Communication (*EN13606*) (*www.centc251.org*).
- Continuing development and adoption of these Standards at the International level within *ISO TC215*.

8.2.4.2 Using SNOMED CT in standard architectures

The broad principles of the established health record architectures are based on a layered structure of components that contain and provide context to lower level components.

The container structures include some or all of the following:

- A top-level component representing the entire health record of one person.
- Intermediate layers representing information from various sources.
- A fixed transaction/composition layer at which an entry or set of entries are attributed to (and possibly signed by) an author:
  - Examples of this level include consultation notes, letters, reports, and other documents.
- Further levels that represent logical grouping within a record covering:
  - Topics, heading and categories;
  - Cluster or batteries of closely associated information.

Within the containment structures are two lower level components:

- Clinical statements:
  - A clinical statement may vary in structure to accommodate different kinds of information (e.g. patient history, clinical finding, investigation results, plans, procedures, medication and other therapies).
- Link statements:
  - Link statements state associations between clinical statements.

- Links statements can be used to specify:
  - Problem-oriented groups of record components and viewing;
  - Causal and other specified links recorded by the author of a record entry.

Each health record component has the potential to include:

- Dates and times of actual and planned events.
- Associations with people, organizations , devices and other entities that participate or are used in relations to a recorded event or plan.
- Codes or other representations that name or provide the semantic information container, link, or statement:
  - *SNOMED CT* fulfills this role in a structured health record.
- Additional data including text, numeric values, images and other digital data.

When *SNOMED CT* is used in a structured record, the links and temporal associations of components combined add further richness to the potential power of *expression*. This has significant advantages and is essential for many types of aggregation and decision support. However, it also adds a complicating factor that should be taken into account when designing, recording, storage, and retrieval facilities.

Example:

To retrieve and analyze the records of patients with two potentially related conditions such as "AIDS" and "Gastro-enteritis" it is not necessary for this combination to be represented in a single *precoordinated* or *postcoordinated concept*. Instead, it is possible to look for co-existence of the individual *Concept* "Gastro-enteritis" within the records of patients who also have "AIDS."

- The advantage of this is that there is no need for the clinician to have made the association between the two conditions. Therefore a more complete assessment of the incidence of "Gastro-enteritis" in patients with "AIDS" can be made.
- The disadvantage is that if a *precoordinated* or *postcoordinated SNOMED CT* representation of the combined *concept* is used, these records will not necessarily be computably equivalent to those with the two conditions recorded separately.

There is no absolute rule on when to use multiple statements associated using record structure constructs, and when to use intrinsic *precoordinated* or *postcoordinated SNOMED CT* representations. The decision maybe influences the functionality of a particular system and the specific user requirements that the system is serving. However, the following guideline is suggested:

- A combined *precoordinated* or *postcoordinated* representation is appropriate if:
  - The combined *concept* is a discrete recognizable *Concept* that differs in some way from the simple combination of the two *concepts*. For example:
    - | Diabetic cataract | is not the same as | Diabetes mellitus | +| Cataract | because other types of cataract may occur in the same patient;
    - | Fracture of radius and ulna | is a clinically recognizable injury, which is most effectively conveyed as a single *concept*.
- Separate records for each *Concept* are appropriate if any of the following apply:
  - The combined *Concept* represents the coincidence of two potentially associated conditions or procedures;
  - The temporal and other characteristics of the two *Concepts* are different;
  - Where the association between the two *Concepts* is causal.

Example:

I Fracture of femur I caused by "fall down stairs" should be represented as separated statements linked by an appropriate record structure component. The *SNOMED CT Concept* "Due to" could be used to name the link between these statements.

8.2.5 Safely representing the context of recorded codes

A variety of contextual factors may affect the interpretation of statements. Contextual factors typically fall into the gray area between record structure and the semantic model. Some of these may have a profound impact on the meaning or interpretation of a statement.

This section divides this issue into four distinct categories:

- Contextual information that is not represented by *SNOMED CT*
- Structures that may be labeled using *SNOMED CT*
- *Status terms* that have a profound effect on *SNOMED CT* encoded statements;
- Context that can safely be represented using *qualifiers*

8.2.5.1 Contextual information that is not represented by SNOMED CT

Clinical statements that contain *SNOMED CT Concept* representations will be associated with some information which is not intended to be represented using *SNOMED CT*:

- Dates, times of an activity of recording and activity;
- Quantitative information including ranges and durations;
- *Identifiers* or names of authors, providers of information or other parties involved in a recorded activity.

SNOMED CT is not intended to represent this information. Appropriate constructs in a standardized or proprietary record architecture should be used to relate this information to *SNOMED CT* encoded clinical statements.

8.2.5.2 Structures that may be labeled using SNOMED CT

Clinical statements may be contained within structures that represent collections of related information. According to the nature of these structures, *SNOMED CT* may be used to label them. However, care should be taken to ensure that any semantic implication from such a label is clearly specified.

Many labels (such as headings within a document) may be used only to organize information for a human reader. The existence of a label such as "plan" or "family history" (even if encoded using *SNOMED CT*) may not necessarily affect the computer interpretation of the data within it.

Implementers should take extreme care to ensure that any semantic implication that a human reader may assume from such labels is stored on the system in a manner that allows safe interpretation. It is recommended that any apparent inherited semantic context should be represented explicitly at the individual statement level.

Example:

If a data element "Family history" is used and the *concept* I diabetes mellitus I is encoded under that heading, the statement stored in the record should encapsulate the full semantics (i.e. Family history+(Associated finding=Diabetes mellitus) using the *SNOMED CT Concept Model*.

Other areas in which structures might be labeled with *SNOMED CT Concepts* include:

- Links between statements - *SNOMED CT* could be used to identify the nature of the link.

Example:

To indicate a presumed causal *relationship*.

- Indication of types (rather than identities) of people of organizations .

Example:

To indicate that the source of a piece of information was the patient themselves or a specified relative.

8.2.5.3 Context and Axis Modifiers

The following are examples of "axis modifiers" which may fundamentally alter the interpretation of information encoded using *SNOMED CT* :

- Subject of information.

Example:

Stating that a relative of the patient has a particular disease. This may be recorded to state either "family history" or a |social context|. If the disease is represented as a *SNOMED CT Concept* then it must be distinguishable from a statement that patient has that disease.

- Stage.

Example:

Stating that a patient should have, has been referred for, or has declined to undergo a particular procedure. These must be distinguishable from statements that a patient has had the stated procedure.

- Negation and uncertainty.

Example:

Stating that a diagnosis has been excluded or is unlikely.

- Contra-indication.

Example:

Related to a treatment specified using its *Concept Identifier*.

There is a temptation to use these modifications as though they were *qualifiers*. This is not a safe practice because the assumption is that a *qualifier* refines the meaning of the qualified *Concept*. A refined *Concept* should always be a *subtype* of the original *concept*. This is not the case for these major modifications as illustrated by the following:

- Records of a |family history of| +|Diabetes mellitus| would not be expected in a response to a *query* for patients with a record of diabetes mellitus (and its *subtypes*);
- Records of "planned" + "hip replacement" should not be counted when analyzing the records of patients who have had any type of "joint replacement.";
- Records that state "meningitis" + "excluded" should not be counted as cases of meningitis;
- Records that state the patient is |allergic to| + "penicillin" are not records of treatment with an antibiotic.

This issue is discussed in more detail in the section on data entry.

The recommended approaches are:

- To ensure that the record structure captures this information in a consistent and reliable way that can be interpreted accurately when retrieving or communicating information.
- If *postcoordinated SNOMED CT* representations are used, the situation *concept* should be qualified by the finding or procedure:
  - For example, the following condition is valid |family history of disorder|: |associated finding| = |diabetes mellitus|
- The finding or procedure must **not** be qualified by the situation as this would result in an *expression* that computed as a *subtype* of the clinical *concept*:

- For example, the following *expression* is **not** valid | diabetes mellitus |: | qualified by | = | family history |.

8.2.5.4 Context that can safely be represented using qualifiers

Where a contextual modification can be logically regarded as a *refinement* of the original *Concept* it is reasonable to use a *qualifier*. Examples of this include "severity," "episodicity" and "laterality."

8.2.5.5 Concepts with built-in context

Some *Concepts* derived from earlier terminologies (i.e. *SNOMED International* and the *Read Codes*) contain built-in context. An example is the *concept* "FH: Diabetes mellitus" (FH being an abbreviation for family history). These *concepts* are in the *Situation with explicit context hierarchy*.

To the extent possible with released context attributes, these *Concepts* are defined (and will continue to be reviewed) so that they are computably equivalent to appropriate *postcoordinated* representations.

Example:

The *concept* "FH: Diabetes mellitus" is defined to be equivalent to the *postcoordinated* representation [36]

- "family history" + (| associated finding | = | diabetes mellitus |).

8.3 Retrieval and Aggregation

SNOMED CT allows consistent processable representation of clinical information. *SNOMED CT enabled applications* should harness this capability to with practical tools for selective retrieval of information in individual clinical records. They should also support aggregation and analysis of clinical data derived from populations of records.

8.3.1 Requirements for selective retrieval

Selective retrieval is an essential function for a health record system. There are two main types of requirements:

- Retrieval of selected records from the records of members of a population of patients for one or more purposes, including the following:
  - Aggregations and analysis of data to support:
    - Epidemiological monitoring;
    - Clinical research;
    - Audit of care delivery;
    - Service planning.
  - Identification of patients with specific risk factors or other characteristics:
    - To allow specific preventative, investigative or therapeutic measures to be appropriately focused;
    - To allow further selective retrieval and analysis of the records of a subpopulation of patients;
    - To enable selection of patients for entry in a clinical trial.
- Retrieval of selected records from an individual patient record to enable:

[36] This is not the case in the first release as the appropriate defining characteristics are not in the release set.

- Display of summary views and/or pre-completed template screens containing appropriate selected information.

  Example: Active problems/diagnoses, current medication, recent investigation results or blood pressure readings.

- Automating responses to questions posed by a decision support protocol.

  Example: To check the record for specified symptoms, findings, investigations, procedures or diagnoses.

The following subsections address issues and requirements that are common to all types of retrieval.

8.3.1.1 Retrieval performance

The following sections identify factors that may influence performance when undertaking selective data retrieval. There are no fixed rules for optimization of retrieval performance. Application developers should interpret the issues outlined in the guide in the light of their experience with the operating systems and data management tools that they use.

An evaluation of different approaches to retrieval was undertaken for the *NHS*, in connection with work on *Clinical Terms Version 3* implementation. This showed that the "best" approach was not the same for all relational databases. Some software environments favor one approach while a different approach may be more effective in another environment. Therefore, it is likely that some of the factors discussed will have a significant impact on some developers, while being less relevant in others.

8.3.1.2 Retrieval quality

The quality of selective retrieval is measured in terms of two factors:

- **Completeness:** Retrieval should select all records that meet the selection criteria;
- **Specificity:** Retrieval should not select any records that do not meet the selection criteria.

The semantic structures of *SNOMED CT* assist application developers to achieve these goals by allowing different *expressions* that represent the same or similar information to be recognized and compared (see *Supporting Selective Data Retrieval*).

The meaning of a *SNOMED CT expression* may be modified by the context in which it is used. Aspects of this context are represented by:

- The record structure in which the *expression* is stored.
- Data directly associated with the *expression* in the record structure (e.g. dates and times, numeric values and units, the identity and role of the originator of the information or the performer of a procedure).
- Explicit or temporal associations with other information in the same record (e.g. co-existent conditions, likely causes of an abnormal observation, reasons for an investigation or therapeutic intervention, etc).

Storing similar information in differing structures complicates retrieval since each query must take account of alternative ways in which the required information may be stored. As a result the semantic strength of *SNOMED CT* may be obscured by the varied approach to structure. Therefore, realization of the full potential benefit of *SNOMED CT*, requires an information model that accommodates *SNOMED CT expressions* and ensure consistent storage of similar information.

Another limiting factor for retrieval is the consistency and completeness of recording. The extent of use of *SNOMED CT* depends in part of policy and guidance at national or organizational levels, which in turn depends on requirements and priorities for data retrieval and reuse. From a technical implementation perspective a key factor in delivering consistent retrieval is a user-interface that facilitates, simplifies and encourages consistent data entry which uses *SNOMED CT expressions* to the extent need to meet relevant requirements.

8.3.1.3 Retrieval criteria involving record structure

Before addressing the specifics of *SNOMED CT* related retrieval criteria it is important to recognize that these only form one part of the picture. Most selective retrieval criteria will include a mixture of predicates, some of which apply to *SNOMED CT* encoded data and some of which apply to other data in the patient record. This non - *SNOMED CT* encoded data includes:

- Data directly related to coded clinical statements. This includes:
  - Dates and times (e.g. time of an event of finding).
  - Organizations , people or devices involved in a recorded activity or finding.
  - Temporal or causal *relationships* with other clinical activities or findings.
  - Quantitative values associated with *SNOMED CT* encoded statements.
  - Associated *status* and contextual information.
- Data related to the patient:
  - Age and sex;
  - Organizations and people responsible for care;
  - Occupation, pre-existing disorders or other known risk factors.

The interplay of these factors with *SNOMED CT* encoded data may affect the optimum approach for data retrieval. Some non - *SNOMED CT* encoded retrieval criteria may significantly reduce the potential set of patients or in patient record entries that qualify for retrieval. In these cases, it may be useful to apply these criteria before testing *SNOMED CT* specific criteria.

Example:

- A retrieval request for the rubella vaccination *status* of eight-year-old girls in a family practice with average population distribution requires the review of less than 1% of the population of records.
- A retrieval request for patients who have undergone a particular procedure in the last month only needs to review record entries made in the last month.
- A retrieval request for the most recent investigation results and current medication might be more processed by initially identifying a set of recent records. Checking these records for relevant *SNOMED CT* values may be more efficient than applying individual queries to the entire record for each of the required items of recent information.
- A retrieval request for people with a rare clinical condition, who also have a relatively common disorder, may be more efficient if the few people with the rare condition are selected first, limiting the scope of the *query* for the more common condition.

These examples illustrate a general point rather than to offer guidance on the specific searches. It is important to bear in mind that the performance, completeness and specificity of retrieval are dependent on the structure of the record as well as the semantics of *SNOMED CT*.

8.3.2 Retrieving records containing selected concepts and their subtypes

Information in health records may be expressed at various levels of specificity.

Example:

To represent diagnoses of:

- Chest infection;
- Left lower lobe pneumonia caused by pneumococcus.

Criteria for selective retrieval may also need to be stated to different levels of detail.

Example:

To retrieve all records of

- Respiratory tract infections;
- Left lower lobe pneumonia;
- Pneumococcal pneumonia.

Occasionally a *query* may be designed to retrieve only record entries that include a particular general *Concept*. This may be useful for a quality review or to find record entries that are too general to *map* to a required classification.

However, in most cases, a general *query* should include more specific *Concepts* recorded in the record. For example, if the selected *Concept* is | Respiratory tract infection | the user would expect record entries containing *Concepts* such as | Chest infection | or "Left lower lobe pneumonia caused by pneumococcus" to be retrieved. The *subtype hierarchy* of *SNOMED CT* is designed to facilitate this type of retrieval. Four techniques that can be used for this purpose are outlined in the following subsections.

Note:

The *subtype hierarchy* is improved with new releases of *SNOMED CT*. These changes need to be considered if more than one version of the hierarchies is used for data analysis.

8.3.2.1 Queries expanded to identify all subtypes

A *query* that explicitly includes the *Concept* Ids of all *subtype descendants* of the *Concept* to be retrieved can be built using one of the following methods:

- A recursive tree-walk following | is a | *Relationships* - from the selection *Concept* to its *subtypes* and the *subtypes* of its *subtype*. Each branch of the tree walk ends on reaching a *Concept* with no *subtypes* or a *Concept* that is already in the set of selected *Concepts*.
- Using pre-generated branch number ranges associated with the selection *Concept* and looking up all *Concepts* with branch numbers in those ranges. This could be much faster than a tree-walk if *Concepts* are indexed by branch-number.
- Using a stored list of *subtype Concept* Ids for frequently queried *Concepts*. This would initially be generated in one of the other methods and then reused in various queries. Any stored list would need to be rebuilt after installing each release of *SNOMED CT*.

The resulting *query* may contain a large list of potential *Concept* Ids, but the actual *query* structure is simple. Therefore as long as the database engine does not restrict *query* size, this type of *query* can be run in any environment that support SQL or an SQL-like *query language*.

This technique is likely to be most effective when a large number of candidate record entries need to be examined and when *Concept* selection criteria are relatively narrow. Selecting all diagnoses using this approach would generate a predicate with tens of thousands of *Concept* Ids. Extremely large queries may not perform efficiently or may fail to run in some environments.

8.3.2.2 Subtype tests on each recorded concept

The *Concept* recorded in each candidate record entry can be tested to determine whether it is a *subtype* of the *Concept* to be retrieved. The test can be applied in one of the following ways (see also Testing and traversing *subtype relationships*):

- A recursive tree-walk following | is a | *Relationships* from the recorded *Concept* to its supertype and the supertypes of its supertypes. Each branch of the tree walk ends on reaching the *Root Concept* or a *Concept* that has already been visited. The test ends with a positive result if the selection *Concept* is encountered during the tree walk. Otherwise when all supertypes have been visited, the test ends with a negative result.

- Optimized *subtype* testing using techniques such as branch numbering and tree-walk enhanced with semantic-type *Identifiers* or *hierarchy* flags.

This technique is likely to be effective when the number of candidate record entries to be examined is relatively small or if the *Concept* selection criteria are broad. Performance is directly dependent on the time taken for each *subtype* test. Therefore, extensive use of this approach may only be feasible by applying one or more of the optimizations discussed in the guide.

8.3.2.3 Use a database with built in hierarchical functionality

Some databases have features which build in hierarchical functionality. These databases may support *extensions* to SQL that allow a predicate to be specified in a way that implies that the database schema "understands" the *subtype hierarchy*.

Example:

It is possible to envision a statement such as:

WHERE Record.*Expression SUBTYPE*-OF 414024009

If a database supports this type of predicate, it clearly simplifies the writing of *SNOMED CT* queries. It is also reasonable to assume that functionality of this type, built into a database engine rather than added as an afterthought, will deliver enhanced performance. However, this assumption should be tested as it depends on how appropriate the internal implementation is to *subtype hierarchy* of the size and complexity of *SNOMED CT*.

8.3.2.4 Branch-range indexing of individual records

Branch numbering is an approach to *subtype* testing that could be extended to index record entries. The branch numbers could be used to produce an index of all record entries stored in an application. The technique is as follows:

- Every record entry is indexed using the branch number of the *Concept* stored in that entry;
- The set of branch number ranges associated with the selection *Concept* is then used to *query* the branch number index.

This approach is likely to deliver high performance retrieval but it has a significant drawback. Branch numbers have to be regenerated after each *SNOMED CT release* and the numbering changes each time. Therefore, any indices based on branch numbers must also be rebuilt after each release, and until this rebuild is complete, this method cannot be used for retrieval. The previous set of branch numbers could be used for retrieval during the transition period but this requires a parallel set of branch numbers and branch number ranges.

The likelihood of enhanced retrieval performance should therefore be balanced against the addition of complexity to terminology updates and record maintenance.

8.3.2.5 Retrieval Based on other Relationships

While many queries will use *SNOMED CT*'s hierarchical *subtypes* to aggregate data, the attribute *relationships* can also be used. For example, to find all procedure *concepts* that use a laparoscope, search in the *relationship file* for *Concepts* with a *relationship* of Using Access Device: Laparoscope. Note that role hierarchies can be used to construct these queries.

8.3.3 Selective retrieval of postcoordinated expressions

The previous section deals with the retrieval of records that contain *precoordinated* representations of *Concepts*. The mechanisms and methods discussed in that section need to be extended to cover *postcoordinated expressions*.

The selective retrieval mechanisms applicable to *postcoordinated expressions* depend on the way in which this data is stored. If data is transformed to generate tables or indices that facilitate retrieval, the form of this derived data determines the type of mechanisms that can be used.

There are two significant factors in the completeness, specificity and performance:

- The structure used for representing *postcoordinated expressions*;
- Whether the information is only stored in the form entered of is also stored in a manner that seeks to facilitate efficient and consistent retrieval of *postcoordinated expressions*.

8.3.3.1 Retrieval from unrestricted relational representations of postcoordinated data

Unrestricted relational representations provide a flexible medium for storage retrieval of *postcoordinated expressions*. A *query* can be specified at any level of detail to examine the primary *Concept* in a statement and any or all of the associated *postcoordinated* qualifications, modifications, or combinations.

However, the number of joins required to specify an appropriate *query* may affect performance.

- Each clinical statement consists of a row in one table joined to a row in a *qualifier* table for each *postcoordinated* refinement. The clinical statement itself may have other structural relations (based on the record structure) and each patient record may consist of hundreds of thousands of statements.

The effect of this will vary according to the power and configuration of the relational database. However, some application developers may seek alternative, more limited representations to improve performance.

8.3.3.2 Retrieval from restricted postcoordinated information

An application may store data in a restricted relational representation, which limits *postcoordination* to a pre-specified set of *qualifiers*. Criteria for selection based on the values of a limited set of *qualifiers* require a minor *extension* to any of the approaches discussed in the previous section. However, there are two significant points to note:

- When applying criteria to the values of a *qualifier*, any *subtype* of the specified value should be selected. This is similar to the consideration for the primary *Concept*. However, the number of tests to be performed will be more limited because:
    - Typically a *qualifier* value will have relatively few *subtypes*;
    - Only record entries that match on other criteria need to be tested.
- Some of the supported qualifying attributes may also occur in *defining characteristics* of some *Concepts*. A *query* that specifies the presence of a particular *qualifier* must not miss these cases. One way to address this issue is to ensure that when storing or transforming data for retrieval, the value of any *defining characteristics* that are also supported, and qualifying attributes should be copied into the qualifying value field.

8.3.3.3 Retrieval from postcoordinated data stored as parsable text or XML

Parsable text *strings* or XML elements are not well suited to rapid retrieval from large populations of records. However, optimization is possible by augmenting the stored form with indexes to *Concepts* (e.g. indexing *Concept Identifiers* or range number) or by using an XML-aware database. Without such optimization it may be possible to achieve acceptable performance for retrieval from individual records, documents or messages represented in a structured form using XML.

8.3.3.4 Retrieval of postcoordinated data stored as entered

Where *postcoordinated* data is only stored as entered, retrieval mechanism must do all the hard work of calculating the *equivalence* between statements expressed in different ways. This is possible for a small-scale search (e.g. within a single patient record) but across a large population of records it may be difficult to achieve an acceptable performance.

8.3.3.5 Retrieval from minimized postcoordinated forms

If *postcoordination* is minimized before storage, this allows most of the search process to be concerned with querying or testing *subtype descendants*.

If the *query* needs to specify selection criteria that cannot be expressed by a single *Concept*, further testing is required. Even then, if there are rules for consistently minimizing *postcoordination*, most queries remain easy to construct and apply.

Some complex queries may present more difficulties with this approach but it remains a reasonable option for application developers concerned with minimizing the overhead related to storage and retrieval while delivering reasonable performance and flexibility.

8.3.3.6 Retrieval from maximized precoordinated forms

Maximization of *postcoordination* offers them most flexible approach. Of the three forms suggested:

- The exhaustive form:
    - Simplifies queries since everything that is true about a *Concept* is stated and there is no need to check *subtype* descents;
    - Carries a heavy storage penalty for every record stored;
    - Requires computation of the representation of each *Concept* after every release.
- The long *canonical form* :
    - Allows queries that are relatively simple provided that a mechanism exists for checking *subtype* descents.
    - Although more terse than the exhaustive approach, storing this information for every record stored still has a significant storage overhead.
    - Requires rechecking or re-computing after a release, but his can be done directly from the *release files* by combining the I is a I *relationships* in the *Canonical Table* with the other (i.e. not " I is a I ") *defining characteristics* in the *relationship file*.
- The short *canonical form*:
    - Requires slightly more care in construction of queries than the long *canonical form*;
    - Requires slightly less storage than the other maximized forms;
    - Like the long *canonical form*, can be rebuilt directly from the release tables.

8.3.4 Requirements for specific uses of selective retrieval

8.3.4.1 Specifying retrieval requirements

An application should provide a mechanism to allow users to specify retrieval requirements using *SNOMED CT*. This facility should allow queries to be generated that combine *SNOMED CT* specific selection criteria with other health record criteria.

A terminology *browser* that combines text searches and *subtype hierarchy navigation* is likely to be essential for defining *SNOMED CT* specific selection criteria.

Facilities for testing and traversing *subtype relationships* are essential for running most *SNOMED CT* queries that can be run against stored records. Additional functions that take account of the definitions of *concepts* and the refinements in *expressions* are needed to support more sophisticated retrieval.

8.3.4.2 Selective retrieval for reporting and analysis

Population-based retrieval and reporting is usually a task that can be run in the background or scheduled for later execution. Therefore, real - time responses are usually not essential.

The process of analyzing a large number of records may take several minutes or perhaps even hours. If the application spends a little time generating an optimized *query* before starting to access the records, this is acceptable and may shorten overall execution time. Therefore, a technique such as *query* expansion may be appropriate for these tasks.

Users may also have requirements for reports on individual patients or a small group of patients. In some cases there may be an expectation of a real - time response to requests for these reports. If so, the delay while several selection criteria are expanded may be unacceptable. If the same criteria are used many times, storage of the expanded form may be a realistic option. Otherwise, an alternative retrieval technique should be considered.

8.3.4.3 Selective retrieval for decision support

Decision support tools are usually used during a consultation with a patient. Real - time response without significant delay is essential if these tools are to be used regularly and perceived as a help rather than a hindrance. A decision support algorithm may need to selectively retrieve several records to inform a single decision or piece of advice. Many of the retrieval criteria are likely to be quite general. The time taken to expand an apparently simple set of criteria so that they include all appropriate *subtypes* is likely to significantly impair performance. The expanded criteria could be stored in or associated with the protocol. However, the requirement to update these with new *SNOMED CT releases* and whenever the protocol changes add to the maintenance burden.

Since decision support protocols are primarily concerned with the records of an individual patient, it may be feasible to test all candidate records (e.g. all records that fall within a specified date range) to see if any of these are *subtypes* of the selection *Concept* (s).

Other alternatives should also be considered.

8.3.4.4 Decision support tools as authors of data in the record

As well as retrieving *SNOMED CT* encoded information a decision support tool may need to make entries in the record. These entries may arise directly from user interaction with a template or protocol. However, some entries made by decision support tools may record decisions made by or advice given by the tool.

8.3.4.5 Retrieving records encoded with Inactive Concepts

Records that have been encoded using *Concepts* that are no longer *active* can be retrieved by using the Historical Associations *Reference Set* values (i.e. "same as," "may be a," | replaced by | and "was a") in addition to | is a | *subtype relationships*.

An application should allow users to specify which (if any) of these *Relationships* or Associations should be followed when determining whether to retrieve a record entry.

- A sensible default is to treat duplicate *Concepts* related by | same as | associations and erroneous *Concepts* related by | replaced by |*Relationships* as though they were interchangeable with the related *Concepts*.
- In the case of the ambiguous *Concepts* related by | may be a | associations the solution is less clear-cut. A choice must be made between the importance of completeness, which is best served by including these *Concepts*, and selectivity, which is better served by excluding them.

8.3.4.6 Retrieving and analyzing legacy coded data

SNOMED CT can also be used for generated queries that examine legacy data recorded using *SNOMED International*, *Clinical Terms Version 3* or earlier versions of the *Read Codes*. This can be done by using the approach outlined in *strategies for data migration*, to generate a query that includes all the *subtypes* of a selection *Concept*. However, the appropriate legacy codes (i.e. *Read Codes*, *CTV3IDs* or *legacy SNOMED codes*) are added to the *query* instead of the *Concept Identifier*.

8.4 Communicating Expressions

SNOMED CT enabled applications must support inbound and outbound communication of information that includes relevant *SNOMED CT expressions*. This section provides an outline of some general issues related to communication of *SNOMED CT* information using standard communication structures.

8.4.1 Representation of SNOMED CT information in communications

Various media exist for communicating computer processable data between applications. These include:

- Messages;
- Structured documents;
- Portable storage media (smart cards, memory cards or other similar devices);
- Application interfaces (including COM / CORBA and programmable web interfaces).

A common feature of any method of communication is the need for a formal standard (or de-facto agreed) representation of the communicated information.

To enable full communication of *SNOMED CT* information these agreed standards must allow the communication of *precoordinated* or *postcoordinated* representations.

Some current standards do not provide explicit support for *postcoordination*. Examples include:

- *HL7* Version 2.x messages;
- EDIFACT implementations of European (*CEN*) Prestandards for laboratory communication used in by the *UK NHS*.

In these cases, it may still be possible to include *postcoordinated* information by agreeing to a syntactic representation that can be used in a single message element.

Example:

Subject to message field length *constraints*, an *expression* in *compositional grammar* could be included in place of a simple code.

The use of this type of technique is not recommended since it may distort the intended semantics of the message, but also, and more significantly, it requires the recipient to agree to parse the code in a particular way. There is no point sending a parsable text representation of a *postcoordinated Concept* to recipients with no understanding of that form of representation.

More recent standards make specific provision for the support of *postcoordination* in representations of clinical statements. Examples include:

- *HL7* Version 3, which includes the "*concept description*" (CD) data type which provides unlimited scope for *postcoordinated* modifiers;
- European (*CEN*) Prestandard ENV13606 for Electronic Healthcare Record Communication, which include a component name structure element, which permits *postcoordination*.

Communication of *SNOMED CT* data using explicit structures for *postcoordination* is strongly recommended. However, where local agreements permit, other solutions may be used. This is discussed further in the next section.

8.4.2 Overlaps between SNOMED CT and Structural Semantics

Many communication constructs have a built-in, or assumed semantic model.

Example:

Rather than having a single coded *expression* to represent a procedure the *HL7 Version 3* class Procedure contains the following coded Attributes .

- Code (cd);
- Priority (priority_cd);
- Reason (reason_cd);
- Method (method_cd);
- Procedure_site (procedure_site_cd);
- Approach_site (approach_site_cd).

Similar constructs occur in other *HL7 Version 3* classes (e.g. Observation) and message standards from other sources. However, the *HL7 Version 3* Procedure example shown here is probably the best example of a particular dilemma for those communicating with a message that takes some aspects of semantics to the structural level while leaving others to the coding scheme.

Suppose we want to communicate the following procedure:

- "Emergency removal of foreign body from stomach by incision".

The *HL7* Procedure class would allow this to be communicated in several different ways.

- The first option uses the *postcoordinated SNOMED CT expressions* and leaves the structural Attribute blank;
- The second uses the structural Attribute and does not postcoordinate the information in the *expression*;
- The third duplicates the same information both in the structure and in the *postcoordinated expression*.

These options represent the main type of approach to overlaps. However, in practice the structural model may permit similar information to be recorded in more that one way and *SNOMED CT expressions* also offer alternatives depending on the extent of normalization of the representation.

The main point is to stress the potential for confusion even when using the same communication structure and the same terminology. This is not a specific problem for *SNOMED CT* or for a particular message design. Any combination of structural and terminological semantics is susceptible to this issue. Since effective communication of information requires both structure and terminology the challenge is to define an interface between structural and semantic models so that they form part of a common *model of meaning*.

8.4.3 Using Reference Sets to represent allowable value sets

Standard message specifications and communication agreements with particular user communities often specify restricted lists of codes that can be used in particular message elements:

- Examples of this include the *HL7* idea of "vocabulary domains" containing "*value sets*" specified for use either in a general or specific context in a message element;
- The *UK NHS* specification for laboratory report messages, which refers to a "bounded list" of *Read Codes* that are to be used in particular fields of the message.

It is inevitable that a broad terminology such as *SNOMED CT* needs to be restricted in this way. A message element intended for representation of a I requested radiology investigation I must clearly contain a *Concept Identifier* that represents a radiology procedure. The limitations may go further than this. The list of procedures that can be requested may be restricted by local convention or regulation.

A *Concept* Reference Set can be used to represent *value sets* that are permitted in a particular type of message or within a particular user community. This facilitates use of a general-purpose *SNOMED CT* enabled *Terminology services* to populate and validate the coded elements of messages.

8.4.4 SNOMED Clinical Terms and HL7

HL7 develops standards for the exchange of health related data. Most new *HL7* standards developed over the last few years have been based on *HL7 Version 3*. The key features of *HL7 Version 3* include:

- A *Reference Information Model* (RIM):
  - This provides a framework for the structure of communicated information.
- A formal development method:
  - This described the various steps to turn a set of requirements into appropriate models and specifications that support communication of the necessary information.
- Separation between logical models and implementation technologies:
  - Many implementations use XML but other technical approaches can be applied to implement the same models.
- Use of external codes systems and terminologies to represent concepts:
  - Model specifications include coding constraints expressed in abstract terms as Concept Domains which are implemented as *Value Sets* drawn from one or more code systems.

Many *HL7 Version 3* models use *SNOMED CT* and this has led to growing demand for guidance on consistent patterns of use. In 2004, the *HL7* Vocabulary Technical Committee launched the *TermInfo Project* to address this requirement. The project was initially conceived as having two work packages:

1. Specification of a general approach to resolving issues related to the interface between *HL7* information models and terminologies or code systems;
2. A guide to use of *SNOMED Clinical Terms* in *HL7 Version 3* communication standards.

The *SNOMED CT* specific package was actively supported by *SNOMED International* as part of a charter agreement with *HL7*. After several rounds of revision and review, the 'Guide to Use of *SNOMED CT* in *HL7 Version 3*' was accepted by *HL7* as a *Draft Standard for Trial Use* (DSTU).

The guide itself contains both normative and informative sections. The normative sections cover:

- Guidance on dealing with specific overlaps between RIM and *SNOMED CT* semantics and recommendations for use of *SNOMED CT* in relevant *Attributes* of various RIM classes;
- Constraints on *SNOMED CT concepts* applicable to relevant *Attributes* in each of the major classes in the Clinical Statement pattern.

The informative sections cover:

- Examples and patterns for representing common clinical statements;
- A general discussion of the potential overlaps between an information model and a *terminology model* and the pros and cons of various possible approaches to their management;
- References to relevant documents and known open issues.

Following adoption the *TermInfo* group have encouraged in-use testing and have elicited and addressed comments on the 'Guide to Use of *SNOMED CT* in *HL7 Version 3* '. Implementer feedback has further contributed to growing understanding of the issues and resulted in refinement of the guidance. Many of the recommendations included in the 'Guide to Use of *SNOMED CT* in *HL7 Version 3*' have been incorporated into domain specific *HL7* Standards and national implementations.

One of the conclusions from the *TermInfo* group is a recognition that terminology and information models are co-dependent. They need to evolve collaboratively to meet requirements for unambiguous processable representations of information. Work with other Information Models (including *EN13606* and *openEHR*) indicates that the issues raised by *TermInfo* are not specific to *SNOMED CT* and *HL7* (*Representing clinical* *information using SNOMED Clinical Terms with different structural information models*). Harmonization efforts involving the *IHTSDO* and *HL7* and other standards bodies continue to address these issues.

As part of the ongoing harmonization work the copyright of the '*Guide to use of SNOMED CT in HL7 Version 3*' is jointly held by *HL7* and *IHTSDO*. In line with *HL7* policies, the document expired as a formal *HL7* DSTU after two years but it is included here in full as it continues to serve two *Roles*:

- A *Description* of the challenges of integrating an expressive terminology, such as *SNOMED CT*, with a rich information model;
- A pragmatic interim approach to these challenges, which allows for and anticipates the evolution of a more integrated solution.

Chapter 9

9 Change Management Guide

This part of the guide addresses requirements that arise from changes to the content, structure and use of the terminology.

The first significant change management challenge relates to *migration* from other coding schemes or from a less structured electronic record system. Decisions must be made about retaining or converting records, queries and protocols originally created using a terminology other than *SNOMED CT*.

- The initial content of this section focused on *migration* from *SNOMED RT*, *SNOMED International*, *Clinical Terms Version 3* and the *Read Codes*. Some of the general points also apply to *migration* from other code systems.

Each release of *SNOMED CT* introduces some changes to content. Many of these changes are additions to breadth and depth of coverage. There may also be corrections to *concept* definitions and enhance the expressivity of the *Concept Model*.

- These changes are an essential characteristic of an evolving clinical terminology that seeks to support changing requirements. However, significant changes need to be evaluated and managed to assess and adjust for any effects they may have on the data entry and retrieval and validity and comparability of data from different sources.

Occasionally there may also be additional technical artifacts or specifications developed to meet emerging requirements.

- Additional material related to updating to support *Release Format 2* has also been included and provides an example of guidance on technical changes.

As systems evolve and as the content and structure of *SNOMED CT* are enhanced there is a continuing requirement to manage changes smoothly and without loss of information or functionality.

- As experience grows, this guide will be further developed to address a broader range or change management issues.

9.1 Managing Content Changes in SNOMED CT

This section of the guide addresses potential issues that may affect implementations when new releases of *SNOMED CT* contain changes to content.

The likely impact of four general types of change are considered.

- Content additions.
- Content inactivation (e.g. inactivation of a *Concept*).
- Changes to *relationships* between *concepts*
- Changes to the *Concept Model* leading to systematic changes to a range of components.

The impact of these changes is assessed in *terms* of the main *record service* functions

- Data entry (e.g. potential requirements to change to data entry protocols).

- Data storage (e.g. whether pre-existing data needs to be migrated or processed in a particular way to ensure consistency).
- Data retrieval (e.g. whether existing queries are likely to need revising to take account of the changes).
- Communications (e.g. whether communication specification are likely to be affected and in particular potential issues from cross-version communications).

Following discussion of these general considerations, the remainder of this section holds specific advice related to any content changes in a release which are expected to require attention from implementers.

9.1.1 Changes and historical notes

9.1.1.1 EPISODICITY no longer modeled in active content

| EPISODICITY | originated in the *National Health Service Clinical Terms Version 3* where it was used not to specify the first episode of a disease for a patient but rather, the first time a patient presented to their general practitioner (GP) for a particular disorder. A first episode of asthma was not intended to represent the first time a patient had asthma, but rather the first time a patient presented to their GP with asthma. | EPISODICITY | has been removed from existing *concepts* and is no longer used in *precoordinated* definitions. It can still be used in *postcoordination* as a *qualifier*.

9.1.1.2 ONSET and COURSE retired

In earlier releases, there were two attributes named | ONSET | and | COURSE |. These were *retired* because they could not be used reproducibly. While | ONSET | was intended to specify the rapidity of onset or the temporal pattern of presentation for a given condition, it was easily confused with the attribute | COURSE | used to represent the duration of a condition. There was not consistent agreement between observers making this distinction.

9.1.1.3 Dose form values moved

The *concept* 105904009 | Type of drug preparation (product) | and its *subtypes* were moved to the *Qualifier* value *hierarchy* as of the July 2007 release. 105904009 | Type of drug preparation (qualifier value) | better represents these *concepts* because they are not products.

9.1.1.4 Renaming the context/situation hierarchy

The *hierarchy* named 243796009 | situation with explicit context (situation) | was called | context-dependent category | until the July 2006 release. The *hierarchy* was renamed to better describe the meanings in this *hierarchy*.

9.1.1.5 Domain change for measurement/evaluation attributes

In releases prior to July 2009, six *attributes* were approved for use for | measurement procedure | only. For the July 2009 release, the *domain* for these *attributes* was expanded to | evaluation procedure |. See Measurement procedures and laboratory procedures on page 311 for a definition and full discussion of | evaluation procedure | and | measurement procedure |.

9.1.1.6 Move of findings to events

In January 2006, a number of *concepts* from the | Clinical finding | *hierarchy* were moved to the *Event hierarchy*. The attributes used to define those *concepts* when they were *descendants* of | Clinical finding | were retained after the *concepts* were moved to the *Event hierarchy*. Additional editorial policies for the use of attributes in the *Event hierarchy* have yet to be established.

9.2 Managing Technical Changes in SNOMED CT

This section of the guide addresses potential issues that may affect implementations when new or revised *SNOMED CT* technical specifications are planned or released.

Each subsection deals with a specific proposed or actual change.

9.2.1 Release Format 2 Update Guide

9.2.1.1 Introduction

9.2.1.1.1 Purpose

The purpose of *RF2* is to provide a format that is flexible, unambiguous and useful.
Its primary aim is to strengthen *SNOMED CT* by providing a format that is simple and stable, while enabling innovation through adaptations to cater for changing requirements.

This specification was developed by harmonizing proposals reviewed by the *IHTSDO* Enhanced *Release Format* Project Group, including:

- The Enhanced *Release Format* Specification (International Health Terminology Standards Development Organisation. *SNOMED Clinical Terms ® Enhanced Release Format Proposed Specification* , 21 June 2007).
- The *Reference Set* Specification (International Health Terminology Standards Development Organisation. *SNOMED Clinical Terms ® Reference Sets - Proposed Specification* , 31 July 2007).
- The Alternate *Release Format* proposed by NEHTA in coordination with their Australian Affiliates.

9.2.1.1.2 Who should read this guide?

The intended audience for this guide includes technical professionals who are involved in the development and/or implementation of healthcare information systems that use *SNOMED CT*.

For detailed technical guidance on the existing *Release Format*, please consult the *SNOMED CT* Technical *Reference* Guide (TRG) and *SNOMED CT* Technical Implementation Guide (TIG), as well as other applicable technical documentation described in the Associated Documentation table.

For technical guidance on using *Release Format 2*, please consult the "*SNOMED CT Release Format 2* - *Reference Set* Specifications" and the "*SNOMED CT Release Format 2* - Data Structures Specification" documents on the Collaborative site.

9.2.1.1.3 Associated Quality Measures

Although the definition of quality measures to monitor the implementation of this standard do not fall under the scope of this guide, they will be covered by the documentation covering the QA and *Release* process for the *Workbench*.

9.2.1.1.4 Summary of Changes

The *RF2* introduces a number of new *concepts* and capabilities. These are summarized below, and described in more detail later in this guide:

- Addition of an *Identifier file* to allow components to be identified by an arbitrary number of *Identifiers* from an arbitrary number of *Identifier* schemes;
- Addition of a module *Identifier* field to all components, enabling the source module in which each component is maintained to be identified, facilitating configuration management;
- Modified handling of the *language* and *dialect* properties of *descriptions*, for reduced complexity with increased utility;
- Introduction of *concept* enumerations making enumerations within *SNOMED CT* more easily extensible, self contained within the terminology (not dependent upon external documentation) and easily compatible across multiple *languages*;

- Addition of a *description logic* modifier *concept* enumeration to the *Relationship file* to represent different *Description Logic relationship types*, for example - some, all, all-some, not-some etc.

A general extensibility design pattern has also been introduced, which allows specification of a number of *Reference Set* formats, to meet different use cases. In *RF2*, *reference sets*:

- Result from the combination of generic *Reference Set* data structures, a design pattern and the application of domain *constraints* according to documented implementation guidelines;
- Use a machine readable model (called a *Reference Set* descriptor) that defines the extended information pertinent to a specific *Reference Set*;
- Make use of *concept* enumerations for representing optional information to enable machine-readability and increased extensibility;
- Apply the same history tracking and naming conventions as used elsewhere in *RF2*.

The *RF2* enhancements all contribute to greater flexibility and more explicit and comprehensive version control than *RF1*, and additionally introduce new features for configuration management. As a result, *RF2* is expected to accommodate evolving collaborative requirements without a need for further fundamental change in the foreseeable future. Since change to the *Release Format* causes difficulty and incurs cost to content developers, implementers and release centers alike, the *RF2* design is expected to result in long *term* savings as well as improvement in product functionality and quality.

9.2.1.1.5 Timescales for change

It should be noted that there is a difference between the release schedule of *RF1* / *RF2* in official IHSTDO-supported *International Releases*, and the release schedule of *RF1* / *RF2* in Member NRC releases. It is entirely possible that *RF1* will have a longer lifespan in Member NRC releases than in *IHTSDO International Releases*.

Actual timescales for *migration* of the *International release* to *RF2* are provided under separate notices, and have not been included in this guide as they are likely to follow a different review cycle.

9.2.1.2 Principles used in the design of RF2

The following principles were used to guide modifications made to the *Release Format*:

- Consistent history representation across all components and across all artifacts deemed in scope of the *Release Format*.
- Consistent identification of all components throughout their lifecycle and clear identification of all other artifacts in scope of the *Release Format*.
- Consistent representation of allowable values for component characteristics.
- Consistent means of extending component data structures to meet future requirements without modification to the existing table structures.
- Consistent non-centralized means of loosely coupled *Identifier* assignment for components and component characteristics.
- Consistent means of representing localizations and translations for all components.
- The data structures should assist implementers to consistently implement *SNOMED CT*. *Component* data structures ideally should not have to change to accommodate changes in editorial policies.
- Ideally component data structures should be simple, generic and flexible.
- Ideally component data structures should be self-contained, removing dependence on external artifacts .
- Dependencies between components should be explicitly stated and machine-readable. For example, it should be possible to express that a *reference set* released as part of an *extension* is dependent upon version X of the Acme *Extension* and version Y of the *SNOMED CT International Edition*.
- There must be a consistent means of identifying modules and their versions --including the *SNOMED CT International Release* itself.
- The *Release Format* should minimize the total effort of meeting requirements where possible by reuse of existing data structures.
- Metadata should be machine-readable.

- *Component* data structures that enable software reuse are preferred over data structures that require special development of parsers.
- It should be possible to produce from a *Release Format* an instance of that release in the immediately prior *Release Format*.
- Specifications should be based on requirements derived from use cases that describe the scope and environment of their intended use.
- *IHTSDO* specifications should provide a common global foundation that permits the development and maintenance of *SNOMED CT enabled applications* that are interoperable across national and organizational boundaries.
- Changes to *IHTSDO* specifications should only be made if the impact on implementation is considered to be proportionate to benefit. Such changes should be recorded.
- Changes to *IHTSDO* specifications should be evolutionary and should deliver incremental benefits to implementers with a minimum of disruption and re-engineering.
- The *SNOMED CT release format* and associated guidance should facilitate a consistent implementation for known use cases.
- The specifications that support implementation of use cases should be done in a way that doesn't limit the ability to realize other use cases within the scope of *SNOMED CT*.
- The *Release Format* is intended to be a distribution format and is not designed to be an implementation format.
- The *Release Format* should be designed to be consumed efficiently.

9.2.1.3 Rationale for moving from RF1 to RF2

9.2.1.3.1 Overview of Release Format 1

The current *SNOMED CT release format* has been in use since January 2002.
During this period the generic and reusable aspects of the existing release structure have been a considerable strength.

Despite this success, there are a number of commonly accepted inconsistencies and limitations in the current *SNOMED CT distribution format*. This section gives a brief overview of the current *SNOMED CT distribution format*, and describes these limitations. For more details see the *Release Format 1 - Detailed specification*.

The current *RF1* format is summarized in *Figure 135*.

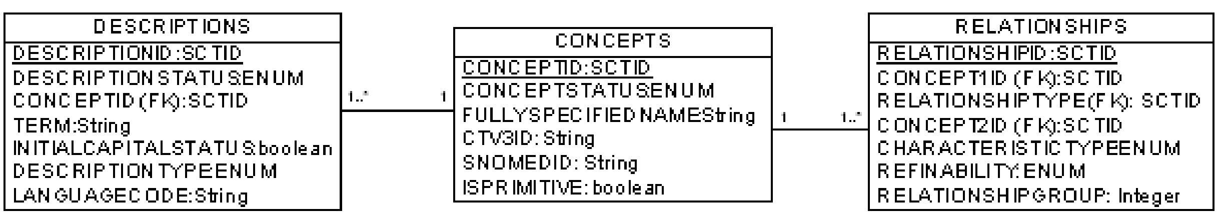


Figure 135: Release Format 1 Core Tables

Each *SNOMED CT concept* is held as a single row in the *CONCEPTS* file. Each *concept* may have one or more *descriptions* associated with it. Each *description* is held in a single row in the *Descriptions* file. Each *relationship*, from a source *concept* to a destination *concept*, is held as a single row in the *Relationships* file. The type of each *relationship* is defined by reference to a linkage *concept*, also held within the *CONCEPTS* file.

Separate file structures are also released to provide support for *maps* to other terminologies and coding systems, history tracking of *components* and *descriptions*, *subsets* of *SNOMED CT concepts* and other uses.

9.2.1.3.2 Drawbacks of Release Format 1

Inconsistencies and limitations in the current *Release Format* have led to a desire for a new *Release Format*. The following list briefly outlines these issues:

- Implicit semantics that must be inferred from external documentation that is not tightly coupled to changes in the terminology itself. Changes in the interpretation of data fields are not represented in the history of the data fields themselves.
- Pervasive use of *integer* enumerations within data fields, rather than using the self-referential means for representing symbolic constants provided for by the *SNOMED CT* terminology itself.
- No consistent and clearly defined mechanism for release centers , developers, implementers, and end users to extend the *RF1* data structures to meet unique and/or common needs not already provided for by the specifications and content of the *SNOMED CT International Release*.
- Inconsistent and unnecessarily complex data structures.
- Field overloading ""where one column represents multiple attributes (i.e. state and reason for inactivation).
- Inadequate Separation of Concerns, where data representation and data usage are often conflated, resulting in a difficulty in supporting software reuse and system evolution over time.
- Inconsistent and incomplete representation of terminology history resulting in a terminology that does not meet basic principles of configuration management and control.
- Inconsistent use of both enumerated values and *concepts* to enumerate values.
- Inconsistent naming and field ordering.
- *Term* length limited to 255 bytes and to plain text format.

Release Format 2 aims to address these issues.

9.2.1.3.3 Overview of Release Format 2

Release Format 2 consists of four primary files or tables. As in the current *SNOMED CT* format, all files:

- are tab delimited text files;
- are *UTF-8* encoded;
- contain a column header row;
- use DOS style line termination (i.e., all lines including the final line are terminated with a carriage return character followed by a line feed character).

The *core table* structure of *RF2* is similar to that of *Release Format 1*, although the fields within each of the *core files* are different. The *core files* within *RF2* consist of a *Concept file*, a *Description file*, and a *Relationship file*.

Each *SNOMED CT concept* is held as a single row in the *Concept file*. Each *concept* may have one or more *descriptions* associated with it. Each *description* is held in a single row in the *Description file*. Each *relationship*, from a source *concept* to a destination *concept*, is held as a single row in the *Relationship file*. The type of each *relationship* is defined by reference to a linkage *concept*, also held within the *Concept file*.

In addition to these files, an *Identifier file* has been added. This file holds one alternate *component Identifier* per row. Each alternate *Identifier* belongs to a particular *Identifier* scheme, and holds that scheme's *Identifier* for the *SNOMED CT Component* that it references. Within a scheme, each *Identifier* uniquely identifies a single *SNOMED CT component*.

The purpose of *RF2* is to provide a format that is flexible, adaptable, consistent, unambiguous and above all useful. Its primary aim is to strengthen *SNOMED CT* by providing a format that is simple yet flexible and powerful, allowing the format to remain constant, while allowing innovation and adaptation to changing requirements.

9.2.1.3.4 Backward compatibility

The proposed *RF2 Release Format* is backward compatible with the previous *SNOMED CT Release Format* (*RF1*), in the sense that all information contained within the current *Release Format* is represented, and legacy file formats can be derived from the new *Release Formats*. However, the *RF2* format contains functionality which is not supportable in the previous *Release Format*.

In order to achieve backward compatibility, the *RF2* may be transformed to create the previous distribution format. Additionally, *International releases* will be made available in both formats for a limited number of consecutive release cycles, for convenience.It is expected that *National Release* Centers will follow the same approach, and also release in dual format for a number of release cycles, unless there are specific reasons not to.

9.2.1.4 Details of Key Changes

The following subsections discuss details of the key changes between *RF2* and the previous *Release Format*.

9.2.1.4.1 Release Format structures and the abstract model

The following tables illustrate the correspondence between the fields of *Release Format 1* (*RF1*) and *Release Format 2* (*RF2*). *Release Format 2* fully represents the Logical Abstract Model for *SNOMED CT concept* and *derivatives*.

Release Format 1 represents a *snapshot view* which is less completely aligned with the logical model. As the tables illustrate all the information represented by *RF1* can be fully captured in the *RF2* representation. However, the reverse is not true. Therefore, some added functionality provided by *RF2* cannot be provided using the *RF1* data.

Table 291: Map between RF1 Concepts Table and RF2 Concept file

RF1 Concepts Table	***RF2 Concept File***
Concepts. ConceptId	*Concept*.id
<not supported> *RF1* releases contain a snap-shot view of the state of each *concepts* at the time of release.	*Concept. effectiveTime*

Concepts. ConceptStatus
• *active* :
  • 0 (*Current*);
  • 11 (*Pending Move*).
• *Inactive*:
  • All other values.
	Concept.active
	• 0 (*Inactive*);
	• 1 (*active*).
	Other aspects of *status* represented by the | Concept inactivation indicator attribute value reference set (foundation metadata concept) |.
	• This set follows the | Attribute value type reference set (foundation metadata concept) |.

<not supported> *RF1* does not support identification of separate modules.	*Concept. moduleId*
Concepts.FullySpecifiedName	*Concept. Description.term* • Where *Description*.typeid = | *Fully specified name*|; • Configured by *Language Reference Set* • The original FullySpecifiedName (which forms the point of reference for the meaning of the *concept* is the FullySpecification with the earliest *effectiveTime*.

RF1 Concepts Table	***RF2 Concept File***
Concepts.Ctv3Id	| CTV3 simple map reference set (foundation metadata concept) | • This set follows the | Simple map type reference set (foundation metadata concept) |
Concepts.SnomedId	| SNOMED RT identifier simple map (foundation metadata concept) | • This set follows the | Simple map type reference set (foundation metadata concept) |
Concepts.IsPrimitive • 0 (*fully defined*); • 1 (*Primitive*).	*Concept.definitionStatusId* • |Defined|; • |*Primitive*|.

Table 292: Map between RF1 Descriptions Table and RF2 Description file

RF1 Descriptions Table	***RF2 Description File***
Descriptions. DescriptionId	*Description*.id
<not supported> *RF1* releases contain a snap-shot view of the state of each *description* at the time of release.	*Description. effectiveTime*

Descriptions. DescriptionStatus
• *active*:
  • 0 (*Current*);
  • 11 (*Pending Move*).
• *Inactive*:
  • All other values.
	Description.active
	• 0 (*Inactive*);
	• 1 (*active*).
	Other aspects of *status* represented by the | Description inactivation indicator reference set |
	• This set follows the | Attribute Value (reference set pattern) |

<not supported> *RF1* does not support identification of separate modules.	*Description.moduleId*
Descriptions.ConceptId	*Description*.conceptId
Descriptions.Term	*Description.term*

<table>
<tr><th>RF1 Descriptions Table</th><th>RF2 Description File</th></tr>
<tr><td>Descriptions. InitialCapitalStatus
• 0 (Initial character case insensitive);
• 1 (Case sensitive);
• <other values not supported>.</td><td>Descriptions. caseSignificanceId
• | Initial character case insensitive |
• | Case sensitive |
• | Case insensitive |</td></tr>
<tr><td>Descriptions. DescriptionType
• Synonym (based on RF2 naming):
  • 0 (Not used in language /dialect):
    • RF2 - Omitted from language /dialect Refset
  • 1 (Preferred term in language /dialect):
    • RF2 - |Preferred| in language /dialect Refset
  • 2 (Synonym in language /dialect) :
    • RF2 - |Acceptable| in language /dialect Refset
• 3 (Fully specified name).</td><td>Description.typeId
• | Fully specified name |
• | Synonym |
• | Definition |
Acceptability in language /dialect represented in | Language type reference set (foundation metadata concept) | for the specified language and dialect.</td></tr>
<tr><td>Descriptions. LanguageCode
• Includes dialect where relevant;
• Language Subsets recommended for representing preferences in dialects.</td><td>Description.languageCode
• Language only not including dialect
• Dialect represented by a | Language type reference set (foundation metadata concept) | for the specified language and dialect.</td></tr>
</table>

Table 293: Map between RF1 relationships tables and RF2 Relationship file

RF1 Relationships Table	*RF2 Relationship File*
Relationships. RelationshipId	*Relationship*.id
<not supported> *RF1* releases contain a snap-shot view of the state of each *active relationship* at the time of release.	*Relationship. effectiveTime*

<table>
<tr><th>RF1 Relationships Table</th><th>RF2 Relationship File</th></tr>
<tr><td><not supported>
Only active Relationship rows are included in the distribution file for each version.</td><td>Relationship.active
• 0 (Inactive);
• 1 (active).
Other aspects of status may in future be represented by the |Relationship inactivation indicator attribute value reference set (foundation metadata concept) |
• This set follows the |Attribute value type reference set (foundation metadata concept) |</td></tr>
<tr><td><not supported>
RF1 does not support identification of separate modules.</td><td>Relationship.moduleId</td></tr>
<tr><td>Relationships.ConceptId1</td><td>Relationship.sourceId</td></tr>
<tr><td>Relationships.RelationshipType</td><td>Relationship.typeId</td></tr>
<tr><td>Relationships.ConceptId2</td><td>Relationship. destinationId</td></tr>
<tr><td>Relationships. CharacteristicType
• 0 (Defining):
  • Inferred - in Relationships Table
  • Stated - in separate Stated Relationships Table
• 1 (Qualifying).
• 2 (Historical).
• 3 (Additional).</td><td>Relationship. characteristicTypeId
• |Inferred relationship|;
• |Stated relationship|;
• |Qualifying relationship|;
• |Additional relationship|.</td></tr>
<tr><td>Relationships. Refinability</td><td>Represented by the |Relationship refinability attribute value reference set (foundation metadata concept) |.</td></tr>
<tr><td><not supported></td><td>modifierId</td></tr>
</table>

9.2.1.4.2 Addition of effectiveTime and active fields

The *effectiveTime* and *active* fields enable the use of a "log style" append-only data model to track all changes to each component for full traceability. Historic data will be supplied in the *RF2 release files*, dating back to the first release in 2002.

Once released, a row in any of the *RF2 release files* will remain unchanged through future releases. In order to change certain properties of a current component (and, therefore, to create a new version of it), a new row must be added to the applicable file, containing the updated data. The *active* field in the newly added row is set to true and the timestamp in the effectiveTimefield indicates the point in time at which the new version comes into effect.

By contrast, where editorial policy does not allow a particular property of a component to be changed whilst keeping the same *Identifier*, the component as a whole is inactivated by adding a new row containing the same data as the final valid version of the component , but with the *active* field set to false and the timestamp in the *effectiveTime* field indicating the nominal release date at which the final version ceased to be valid.

It is thus possible to see both the current values and any historical values of a component at any point in time.

9.2.1.4.3 Active field

As mentioned above, each file contains a *Boolean active* field, used to indicate whether, after the point in time specified in the effectiveTimefield, the version of the component expressed in the row is *active* or *inactive*.

This field replaces the *status* field with a simple binary state. In the previous *Release Format*, this field was overloaded to enumerate both whether the *concept* was *active*, why it was inactivated, and whether it was about to change (or had changed) authority.

The additional information encoded in *RF1*'s *status* enumeration is represented in *RF2* using the following *reference sets*:

- *Concept* inactivation indicator;
- *Description* inactivation indicator;
- *Relationship* inactivation indicator.

These three *reference sets* conform to the *Attribute Value reference set* pattern, and are further described in the "*SNOMED CT Release Format 2 - Reference Set* Specifications" document.

9.2.1.4.4 History tables

History tracking in *RF2*'s main files uses a log-style, append-only data model. Therefore, the separate ComponentHistory file that formed part of the original *Release Format* is no longer required with *RF2*.

The associations between *inactive* and *active Concepts* that are currently supported by *Historical Relationship* types (e.g. I SAME AS I, "REPLACED BY) will continue to be supported. References held in the *References table* from an *inactive component* to other equivalent or related *components* that were current in the *Release Version* in which that *component* was inactivated will also continue to be supported.However, both of these associations have now moved from the *Relationships* file and the References file to one of the following IHistorical associationI *reference sets*.

Table 294: RF1 to RF2 History Field Mappings

***RF1* source**	***RF2* IHistorical associationI *reference set***
MAYBE A (in *Relationships table*)	IPOSSIBLY EQUIVALENT TO association *reference set* I
Refers to ('7' in *References table*)	IREFERS TO *concept* association *reference set* I
Similar to ('3' in *References table*)	ISIMILAR TO association *reference set* I
MOVED FROM (in *Relationships table*) Moved from ('6' in *References table*)	IMOVED FROM association *reference set* I
MOVED TO (in *Relationships table*) Moved to ('5' in *References table*)	IMOVED TO association *reference set* I

<table>
<tr><th>RF1 source</th><th>RF2 |Historical association| reference set</th></tr>
<tr><td>Alternative ('4' in References table)</td><td>|ALTERNATIVE association reference set |</td></tr>
<tr><td>WAS A (in Relationships table)</td><td>|WAS A association reference set |</td></tr>
<tr><td>REPLACED BY (in Relationships table); and
Replaced by ('1' in References table)</td><td>|REPLACED BY association reference set |</td></tr>
<tr><td>SAME AS (in Relationships table)
Duplicated by ('2' in References table)</td><td>|SAME AS association reference set |</td></tr>
</table>

These *reference sets* all conform to the Association *reference set* pattern, and are further described in the "*SNOMED CT Release Format 2 - Reference Set* Specifications" document.

9.2.1.4.5 Field naming

Lower camel case has been used for field names in distribution file headers and in documentation that describes these files. File names will use upper camel case (starting with a capital letter). File names have also been altered to use a singular not plural form.

An example of upper Camel Case is ThisIsUpperCamelCase. An example of Lower Camel Case is thisIsLowerCamelCase.

9.2.1.4.6 Field Ordering

Records in the *Concept*, *Description*, *Relationship* and *Reference Set* member files each start with the following four fields:

- id;
- *effectiveTime*
- *active*
- *moduleId*

The four fields have the following meanings:

- The id field provides a unique *Identifier* for the component described by the record;
- The effectiveTimegives the nominal release date at which this version of the component came into effect;
- The *active* flag states whether the components *active* or *inactive*;
- The *moduleId* identifies the source module in which the component is maintained.

The *Identifier file* does not follow the same format, as it works in a slightly different way to the other files, and is described in more detail in the "*SNOMED CT Release Format 2* - Data Structures Specification" document.

9.2.1.4.7 Concept enumerations

Concept enumerations have been used across *RF2* to replace *integer* enumerations that can only be understood by referencing external documentation. For example, in *RF1*, a *concept status* value of '4' indicates *concepts* that are *inactive* because they are ambiguous. In *RF2*, *concept* enumeration simply uses *concepts* in a metadata *hierarchy* to represent the enumerated *value set* rather than using arbitrary *integer* values directly. Using *concepts* to represent the enumerated values has the following advantages:

- The terminology is self contained, removing the requirement for external documents to explain the meaning of enumerated values;
- Full *language* handling capabilities are available for the enumerated values' representation, useful for standardized multi-lingual representation, and translation support;
- Machine readable model constructs can be used to further describe and enrich the enumerated values.

The following fields have been converted to *concept* enumerations:

Table 295: RF1 to RF2 enumerated field changes

File	**Existing *RF1* field name**	***RF2* field name**
Concept	*IsPrimitive*	*definitionStatusId*
Description	*DescriptionType*	*typeId*
Description	*InitialCapitalStatus*	*caseSignificanceId*
Relationship	*CharacteristicType*	*characteristicTypeId*

Care should be taken not to confuse *Concept* Enumerations with the *term* "enumeration" as used in representational formats. A *Concept* Enumeration is a *concept* whose immediate *children* represent possible values in a range. Each possible value is represented by a single *child concept*, whose *preferred term* may be used, for example, to enable selection from a pick-list of one or more values from the range.

Mappings from *RF1* values to *RF2 concept* enumerations are given below:

Table 296: RF1 to RF2 enumerated value mappings

***RF2* field name**	***RF1* value**	***RF2* valuel**
definitionStatusId	0	| Defined |
	1	| *Primitive* |
typeId	(no specified value)	| Definition |
	3	| *Fully Specified Name* |
	0, 1 or 2	| *Synonym* |
caseSignificanceId	0	| Initial character case insensitive |
	1	| Case sensitive |
	(no specified value)	| Case insensitive |
charactristicTypeId	3	|Additional *relationship* |
	0	|Inferred *relationship* |
	0	|Stated *relationship* |
	1	|Qualifying *relationship* |

***RF2* field name**	***RF1* value**	***RF2* valuel**
	2	(no specified value) - now modeled through the *inactive* association *reference set*.

9.2.1.4.8 Reference Set Data Structures

9.2.1.4.8.1 Overview of Reference Sets

Reference Set data structures provide the foundation pieces for *RF2*'s generic extensibility mechanism. These building blocks provide a common foundation for *extension* builders to extend *SNOMED CT*, and provide *RF2* with the capability to grow with the *IHTSDO*'s requirements over time.

Conventions applied to the *RF2* files such as field naming, field ordering and history tracking have also been applied to the *Reference Set* specification. This has been done to provide consistency across all components in the *Release Format*.

Generic data structures for *Reference Sets* have been used to create a simple core structure that can be extended to meet a variety of requirements, rather than a complex and inextensible structure that can only be used in a finite and constrained number of ways to enforce editorial policy. This stems directly from a desire to decrease impact on the *SNOMED CT* community by being able to meet future requirements without having to alter the underlying data structures.

Using these generic structures, it is possible to extend the data stored within the main files of *SNOMED CT* to satisfy new use cases without altering the primary structure itself. Containing this extended information in externalised structures such as *Reference Sets* also enables terminology consumers to opt in or out of the content without burdening the primary files with the content. This prevents users from having to download all content and filter out what they don't want, and instead allows them to import the *extension* content should it be desired.

Reference Sets allow the *SNOMED CT core* data structures to be extended, allowing existing components to be grouped together into a set, each tagged with a number of additional fields. Each of these additional fields may either be another *SNOMED CT component*, a *string* or an *integer*. *Reference set* descriptors are also introduced, providing a way to identify the format and purpose of each additional field in a machine readable way. Examples of *reference set* data structures are provided in the "*SNOMED CT Release Format 2 - Reference Set* Specifications" document.

9.2.1.4.8.2 RF1 Subsets Representation

The *RF1 Subset* mechanism consists of two tables: a *Subsets table* and a *Subset Members* table. Each row in the *Subsets table* describes a *Subset* and characteristics of that *Subset*, as described in the table below.

Table 297: Subsets Table

Field	***Description***
SubsetId	The unique *SNOMED CT Identifier* for this *Subset*
SubsetOriginalId	The unique *SNOMED CT Identifier* for the original *Subset* of which this *Subset* is a version.
SubsetVersion	An *integer* incremented for each revised release of a *Subset*
SubsetName	A name that describes the purpose or usage of this *Subset*.

SubsetType	Indicates the nature of the *Subset* and the type of *SNOMED CT Component* that may be a member of the *Subset*.
LanguageCode	Identifies the *Language* and optionally the *Dialect* to which the *Subset* applies (only used for *description* -based *subsets*: *Language*, *Realm Description*, and *Realm Concept*).
RealmId	Identifies the *Realm* to which the *Subset* applies.
ContextId	May identify the *Context Domain* to which the *Subset* applies

Each row in the *Subset Members* table sets the *status* of a member of an identified *Subset*.

Table 298: Subset Members Table

Field	***Description***
SubsetId	The unique *SNOMED CT Identifier* for this *Subset*
MemberId	The *SNOMED CT Identifier* of this *Subset Member*. This may be a *Concept Identifier*, *Description Identifier* or *RelationshipId*.
MemberStatus	An *integer* specifying the *status*, type or *order* of this member.
LinkedId	Valid for *Navigation* and *Duplicate Terms Subsets* only. For *Navigation Subsets* it is the *SNOMED CT Identifier* for a *Concept* that is a *Navigation child* of the *Subset Member*. For *Duplicate Terms Subsets* it is the *SNOMED CT Identifier* for the highest priority *Descriptions* haring theDuplicate *Term*.

Some *Subsets* and their members are generated automatically from an XML definition file.

9.2.1.4.8.3 Representing Subsets as Reference Sets

An existing *RF1 Subset* may be represented as an *RF2 Reference Set* in the following way:

A *concept* should be created in the | *Reference Set* | metadata *hierarchy*, using information in the *Subset* table record. A Descriptor for the *Reference Set* should also be set up using information in the *Subset* table record. Then, one *Reference Set* member record should be created for each *Subset Member* table record.

The way in which the *subsets* are represented in *RF2* depends on the *SubsetType* value, as follows:

Table 299: Representing Subsets as Reference Sets

***SubsetType* value**	***Description***	**Mapping to *RF2***
1	*Language*	*Language* type *Reference Set*
2	*Realm Concept*	Ordered type *Reference Set*
3	*Realm Description*	*Language* type *Reference Set*

***SubsetType* value**	***Description***	**Mapping to *RF2***
4	*Realm Relationship*	Ordered type *Reference Set*
5	Context *Concept*	Ordered type *Reference Set*
6	Context *Description*	*Language* type *Reference Set*
7	*Navigation*	Ordered type *Reference Set*.
8	*Duplicate terms*	Ordered type *Reference Set*

9.2.1.4.8.4 Representing Subsets as Ordered type Reference Sets

I Ordered type I *Reference Sets* can be set up as follows:

First, set up a new *concept* for the *Reference Set* in the IOrdered typeI metadata *hierarchy*. The position in the *hierarchy* should be given by the *RealmId* and ContextIdfields in the *Subset* record, as follows:

SNOMED CT Model *component*

Foundation metadata *concept*

Reference set

Ordered type

RealmId

ContextId

If either the *RealmId* field or the *ContextId* fields are "0", "1", blank or null in the *Subset* record, then that level should not be set up in the metadata *hierarchy*. If a *concept* already exists under IOrdered typeI with a matching *RealmId* and *ContextId*, then the new *Reference Set* should be set up in that position (as opposed to creating two IOrdered typeI *children* with duplicate names).

First, the *concept* describing the *Reference Set* should be created with the following values:

Table 300: Reference Set Concept

Field	**Data type**	**Set to**
id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *Reference Set*. If a full state valid representation of a *subset*'s history is required, then each previous release of the *Subset* files must be processed in turn (by identifying *Subset* records with a matching *SubsetOriginalId*, in their SubsetVersion *order*), and each amended version must be applied to the *reference set* by appending rows in the usual fashion. The effectiveTimeof each applied change should be set to the date that each version of the *Subset* was released.
active	*Boolean*	'1'

Field	Data type	Set to
moduleId	*SCTID*	The module *Identifier* for your authoring organization .

Then, add up two *Descriptions* for the FSN and the *Preferred Term* of the *concept*:

Table 301: Reference Set Descriptions

Field	Data type	Set to
id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *Reference Set*.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
conceptId	*SCTID*	The *Identifier* of the *concept* describing the *Reference Set* that you've just added.
languageCode	*String*	The *language* of the *Description*. This field should be set to the *language* that the *Subset* was defined in, for example - 'en' for English.
typeId	*SCTID*	Create two *Description* records, one for each of the following types: | Fully specified name |, | *Synonym* |.
term	*String*	The *term* for the | *Synonym* | should be set to the *SubsetName* field in the *Subset* record. The *term* for the |FSN| should be set to the same, but appended with "*reference set* (foundation metadata *concept*)".

Finally, add one *Reference Set* member record for each record in the *Subset Members* table for the *Subset*:

Table 302: Converting a Priority Subset to an Ordered Reference Set

Field	Data type	How to populate
id	*UUID*	A new unique *Identifier*
effectiveTime	*Time*	The nominal date on which this release was made.
active	*Boolean*	'1'
moduleId	*SCTID*	Set to the *moduleId* of the authoring organization .

Field	Data type	How to populate
refsetId	*SCTID*	A reference to the *concept* describing the *Reference Set* that you've just created.
referencedComponentId	*SCTID*	Set to *MemberId* in the *Subset Members* table record.
order	*Integer*	Set to *MemberStatus* in the *Subset Members* table record.

Note: Although a *Navigation Subset* can be represented in an |Ordered type| *reference set* as described above, the values of the *linkedTo* field would then have a different meaning, referencing a *child concept* instead of grouping *components* together.

A Descriptor can also be set up for the *Reference Set* if required, as follows:

Table 303: -

refsetId	*referencedComponentId*	*attributeDescription*	*attributeType*	*attributeOrder*
| *Reference set* descriptor|	*Concept describing refset*	| Referenced component |	| *component* type|	0
| *Reference set* descriptor|	*Concept describing refset*	| *Order* |	|Unsigned *integer* |	1
| *Reference set* descriptor|	*Concept describing refset*	| Linked to |	| *component* type|	2

Where *Concept describing refset* is the *Concept* that you've just set up to describe the *Reference Set*. The | *Order* | and |Linked to| *concepts* that describe each additional *Attribute* in the *Reference Set* can also be replaced by more descriptive ones if required. To do this, create the new *concepts* describing the additional fields under the | *Reference set Attribute* | metadata *hierarchy*.

9.2.1.4.8.5 Representing Subsets as Language type Reference Sets

Language type *Reference Sets* can be set up in a similar fashion to the above, with the following exceptions:

The *LanguageCode* field in the *Subset* record should be used to link the *Reference Set*'s *concept* into the appropriate place in the | *Language* type| metadata *sub-hierarchy*. For example, a value of "en-US" in the *LanguageCode* field would result in the *Reference Set*'s *concept* being created under |US English|:

SNOMED CT Model *component*

Foundation metadata *concept*

Reference Set

Language type

English

US English

RealmId

ContextId

- Where the *SubsetType* is " *Language*" and the *LanguageCode* is a single level (e.g. "en"), then the *Reference Set* should be created at the first level, under |English| in the example above;
- Where the *SubsetType* is " *Language*" and the *LanguageCode* is a two level (e.g. "en-US"), then the *Reference Set* should be created at the second level, under |US English| in the example above;
- Where the *SubsetType* is "*Realm Description* ", then the *Reference Set* should be created under *RealmId* in the example above (where *RealmId* is the value of the RealmIdfield in the *Subset* record);
- Where the *SubsetType* is "Context *Description* ", then the *Reference Set* should be created under *ContextId* in the example above (where *ContextId* is the value of the *ContextId* field in the *Subset* record and *RealmId* is the value of the RealmIdfield in the *Subset* record).

The *Reference Set* member records should be created as described in the following table:

Table 304: Converting a Language Subset to a Language Reference Set

Field	Data type	How to populate
id	*UUID*	A new unique *Identifier*
effectiveTime	*Time*	The nominal date on which this release was made.
active	*Boolean*	'1'
moduleId	*SCTID*	Set to the *moduleId* of the authoring organization .
refsetId	*SCTID*	A reference to the *concept* describing the *Reference Set* that you've just created.
referencedComponentId	*SCTID*	Set to *MemberId* in the *Subset Members* table record.

A Descriptor can also be set up if required.

9.2.1.4.9 Metadata hierarchy

As the *RF2* data structures and extensibility mechanism contain a number of *concept* enumerations, it is necessary to define *concepts* that represent these values. As well as the enumerated values, there are other machine-readable *concept model* structures not visible in the *Release Format* that require metadata (for example, the structures that define the format of a *description* type).

To meet this need, a new top-level *hierarchy* has been defined as a sibling to the | *SNOMED CT Concept* |, called | *SNOMED CT* Model *component* |. Note that existing metadata *concepts* held within the | *SNOMED CT Concept* | *sub-hierarchy* (|Linkage| and | *Namespace* |) will be moved to the | *SNOMED CT* Model *component* | *sub-hierarchy*.

The top level of the *SNOMED CT* Model component *hierarchy* is structured as follows:

- 138875005 | SNOMED CT Concept (SNOMED RT+CTV3) |
  - 900000000000441003 | SNOMED CT Model Component (metadata) |
    - 900000000000442005 | Core metadata concept (core metadata concept) | ...
      - (*Concept enumerations* required to support *SNOMED CT International Release* data structures)
    - 900000000000454005 | Foundation metadata concept (foundation metadata concept) | ...
      - (metadata required by the *Reference Set* extensibility mechanism)
    - 106237007 | Linkage concept (linkage concept) | ...

    - 246061005 | Attribute (attribute) | ...
    - 416698001 | Link assertion (link assertion) | ...

- 370136006 | Namespace concept (namespace concept) | ...

Figure 136: SNOMED CT Model Component Hierarchy

Note that only | is a | *relationships* will exist between *concepts* in the | SNOMED CT Model Component || *hierarchy* |. Other associations between *concepts* in this *hierarchy* can be modeled using an | Association type reference set (foundation metadata concept) | (see *Association Reference Set*).

9.2.1.4.10 SCTIDs and UUIDs

UUIDs are unique universal *Identifiers*. These 128 bit unsigned *integers* can be used to identify all *SNOMED CT components* internally.

SCTIDs will continue to be used as primary and foreign keys for *concepts* and *descriptions*, both to identify a component and to reference other components. This form is essential for vendors and implementers who will reference *concepts* in *Clinical Information Systems* and messages. *SCTIDs* will also be used to identify *relationships*. However, *UUIDs* will be used to identify *Reference Set* members.

In addition, any *UUIDs* used in development can also be published as additional *Identifiers* via the *Identifier file*.

9.2.1.4.11 Description text

The values permitted within the *description term* field have been extended to support arbitrary length content, and support mark-up content such as XHTML. The 900000000000538005 | Description format reference set | allows a maximum length and format to be associated with each *description* type within the *Description file* (see *Description format reference set specification*) .

This mechanism allows descriptive text of different formats (other than *Fully Specified Names* and *Synonyms*) to be associated with *concepts*, while appropriately constraining existing *description* types. This enables all *descriptions* associated with *concepts* that may require translation to be held in one place in the *Description file*.

9.2.1.4.12 LanguageCode

The languageCode field is retained in the *Description file*, but is restricted to contain only coarse-grained *language* information (e.g. "English" or "French"). *Reference sets* are used to indicate *dialects* and contexts, where required. As an example, the *term* "Bulldozer" would appear once in the *Descriptions* file with the *language* code en ("English"), but be listed separately in each of the Australian, UK and US English *language* national *dialect Reference Sets* as a valid *term* in all three *dialects*.

The languageCode field in *RF2* is a text field and is bound to the *ISO* 639-1 two-character *language* codes.

9.2.1.4.13 Addition of a modifierId field

The underlying semantics on which *SNOMED CT* is based assumes that all *relationships* are existential restrictions. In other words, a *relationship* in *SNOMED CT* implies that there is ***some*** instance of that *relationship* from each instance of the source *concept* to any instance of the target *concept*. Other types of *relationship*, such as universal restrictions do exist and have been studied extensively. For example, the existence of a universal *relationship* in *SNOMED CT* would require that ***all*** instances of that *relationship* from each instance of the source *concept* be to an instance of the target *concept*.

As an example, take the following hypothetical *relationship* |Has *child* | between two *concepts* | Woman | and | Girl |:

| Woman | | Has child | = | Girl |

In *SNOMED CT*, the *relationship* is implicitly an existential *relationship*, that we can make explicit in the above syntax by adding the modifier "some:", as follows:

| Woman | **some:** | Has child | = | Girl |

This means that every instance of | Woman | has at least one | Has child | *relationship* that has as its target an instance of | Girl |. In other words, in our hypothetical world, every woman would have at least one daughter, but may also have any number of sons.

If the existential *relationship* were changed to a universal *relationship*, as follows, then the meaning would be changed:

| Woman | **all:** | Has child | = | Girl |

This means that, for every instance of | Woman |, all its | Has child | *relationships* must have a target of | Girl | . In other words, in our hypothetical world, women could only have daughters or no *children* at all, and could not have sons. This has a very different meaning from the existential *relationship* currently implied within *SNOMED CT*.

A new *modifierId* field has been added to the *Relationship file* to allow future expansion. This *concept* enumeration field will initially be set to | Some | to keep compatibility with the existing semantics of *SNOMED CT*. Widening the range of this field to include other values (such as | All |) would in future increase the expressive power of *SNOMED CT*. However, this is likely to come at the cost of an increase in reasoning complexity, leading to potential issues for classification tooling. Therefore, before extending the range of this field beyond | Some |, a test of the impact on tooling will need to be performed, and the results reviewed and approved.

Notes:

1. The modiferId field has been included at this stage as the *RF2* format is likely to be stable for at least a five year period, without addition or deletion of fields. Within that period it is anticipated that other *modifierId* values will be added. Therefore, although not fully implemented at this stage, this field has been included in the initial RF2 specification as it represents an integral part of the *Description Logic* used by *SNOMED CT*.
2. Any expansion of *SNOMED CT* to include *relationships* with a *modifierId* set to a value other than | Some | will be discussed with *Members* first and approved by the Technical Committee.

9.2.1.4.14 Addition of moduleId field

A *moduleId* field has been added to help identify content and dependencies in a release. This enables release centers to compose a unified release (in a single set of *release files*) from a number of different modules, yet still identify the origin of content down to a row level within each of the releases. For example, this may be used to differentiate *SNOMED CT* International content, Australian Medicines terminology and Pathology content within the Australian *National release*. Currently this is only possible if all modules are assigned unique sub - *namespaces*, and content consumers parse *Identifier namespaces* to differentiate modules.

Components may move from one module to another within a particular *namespace*. Without a *moduleId*, there would be a need to retire a component in one *namespace*, and add another (with a new *SCTID*) to the *namespace* that the component is moved to. Additional *relationships* would also need to be set up, to link the old and new components together. None of this administrative and error-prone work is required if *moduleIds* are used.

Combining the *moduleId* with *Reference Sets* provides a powerful versioning mechanism. The Module Dependency *reference set* (described in more detail in the *SNOMED CT Release Format 2* - Reference SetSpecifications" document) can represent interdependencies between modules and define compatible versions. This functionality can thus be used to represent version information for a terminology's components within the terminology's content itself, in a machine processable way.

The diagram below provides an example of this structure. It shows the components making up an Australian national *SNOMED CT extension* release, containing subcomponents. The links can be described using members of the Module Dependency *Reference Set*. In the example below:

- *SNOMED CT* Australian *Extension* depends upon *SNOMED CT* International 2008-01-31;
- Australian Pathology *Extension* depends upon *SNOMED CT* Australian *Extension* 2008-08-31;
- Australian Discharge Summary *Extension* depends upon *SNOMED CT* Australian *Extension* 2008-08-31.

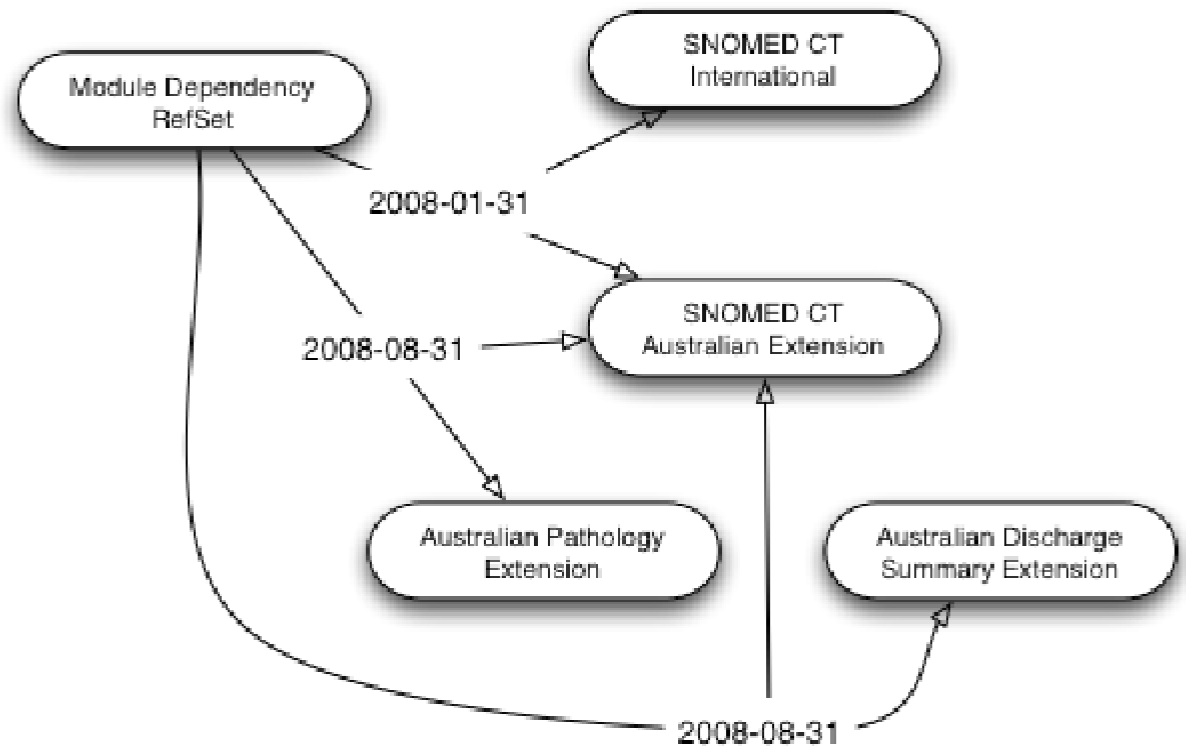


Figure 137: Illustration of module dependencies

9.2.1.4.15 Fully Specified Names and Preferred Terms

RF2, like the original *Release Format*, allows *Fully Specified Names* (FSNs) to be specified in each *language* using the *Description file*. Multiple FSNs and multiple *synonyms* may exist with the same languageCode for a *concept*. However, a particular *language Reference Set* will only contain a single FSN and a single *preferred term* for a *concept*.

As part of the *language* modifications made in the *RF2*, only a broad definition of a *language* can be made for a *Description*. For example, it is possible to declare a *Description* as English, but not US English. Also *RF2* no longer contains a *description* type value for a "*Preferred Term* ", only types of | *Fully Specified Name* | and | *Synonym* |. Each *Synonym* can then be assigned an |Acceptability value| of either |Acceptable| or |Preferred| when included in a *language reference set*.

As a result of these changes, the preference for particular *descriptions* in a *language* or *dialect* is now represented using a *Reference Set*. This matches the specified use of *Language Subsets* in *RF1*, and deliberately removes the deprecated approach applied in some releases where preferences were derived directly from the released *Descriptions* file.

Language reference sets also introduce the notion of overriding or inactivating particular *Descriptions* that may be appropriate in one *dialect*, but not appropriate in another dependent *dialect* or context. This is achieved by allowing *Descriptions* that are inherited from a parent *language reference set* to be overridden in a *child language reference set*.

9.2.1.4.16 Field removals

A number of fields that appeared in the previous *Release Format* do not appear in *RF2*. These fields are listed in the table below, with an explanation of why each field has been removed and to where it has been moved. Note that where a *reference set* replaces a field, this *reference set* will be provided with the *RF2* distribution.

Table 305: RF1 fields that are not use in RF2

File	Field	Rationale for change	Moved to
Concept	CTV3Id	To avoid cluttering the *concept* table.	Moved to the | *CTV3* simple map| *Reference Set*.
Concept	*SNOMEDID*	To avoid cluttering the *concept* table.	Moved to the | *SNOMED RT* IDsimple map| *Reference Set*.
Concept	FullySpecifiedName	This field duplicates one of the *fully specified names* represented in the *Description file*. This duplication has led to misunderstanding of the use of *fully specified names* in multi-lingual distributions of *SNOMED CT*.	The original FSN, which may be required for translation purposes, can be identified as the FSN for the *concept* that has the earliest *effectiveTime*.
Relationship	*Refinability*	As this information is only useful in some environments, it has been moved out of the *Relationship file* to avoid cluttering it.	Moved to the | *Relationship refinability reference set* |.

9.2.1.4.17 Identifier file

The *Identifier file* has been added to provide a standardized means of attaching co-referent *Identifiers* from many different schemes to *SNOMED CT components*. This provides a means to:

- link *UUIDs* and *SCTIDs*, and;
- add external *Identifiers* such as *LOINC* codes, where these are truly co-referent; and;
- track history and organizational responsibility by linking old *SCTIDs* to new ones, where components are transferred from one name space to another, in order to allow uninterrupted use of the old *SCTIDs*.

This provides a mechanism for generically binding *SNOMED CT components* to an arbitrary number of alternative *Identifiers*. It is a more scalable solution than appending columns as needed to the *Concept file*.

Note that the *Identifier file* is not intended as a mapping solution. This structure is only intended to support cases where the external *Identifier* means exactly the same thing as the *SNOMED CT* component to which it is attached. For example, it is not envisioned that ICD-9, *ICD-10* or *CTV3* codes would be entered into this file.

The *Identifier file* is intended to provide a mechanism to represent external codes for *SNOMED CT components* where the meaning is exactly the same. For example, in the Australian Medicines terminology (AMT), *concepts* are "generated" from data sourced from the Therapeutic Goods Administration (TGA) and the TGA has an ARTGID for every therapeutic item. This mechanism allows the ARTGIDs to be attached directly to the corresponding AMT *concept* when generated. In this instance, the *Identifier file* assists meeting the use case without burdening the *descriptions* file or *concepts* file with this content.

9.2.1.4.18 References table

In the previous *Release Format*, the *References Table* contained References from *inactive components* to other equivalent or related *components* that were current in the *Release Version* in which that *component* was inactivated. Each Reference indicated the nature of the *relationship* between the *inactive* and persistent component.

In *RF2*, this information is held in *Historical Association Reference Sets*.

9.2.1.4.19 Textual Descriptions

In the previous *Release Format*, a separate Textual *Descriptions* file held long *descriptions* (of up to 512 bytes, in plain text format). In *RF2*, these textual *descriptions* are transferred to the *Description file*.

9.2.1.4.20 Mapping

9.2.1.4.20.1 Mapping Overview

No bespoke mapping file structures (for example, CrossMapSets tables) have been defined in *RF2*. Instead, the simple map *Reference Set* pattern and alternate map *Reference Set* pattern should be used, in conjunction with other *Reference Set* patterns, to define *Reference Sets* for mapping purposes. See the "*SNOMED CT Release Format 2 - Reference Set* Specifications" document for more details.

9.2.1.4.20.2 Mapping in RF2

RF1 CrossMaps that have a type of either one-to-one or one-to-many can be represented in *RF2* as described below. The type of an *RF1 CrossMap* can be identified from the *MapSetType* field in the CrossMapSets table. The following values in the *MapSetType* field are possible:

Table 306: RF2 Mapping Type Representations

Value	Meaning	Examples	Mapped to *RF2*
1	One-to-one	ICD-O	Can be mapped automatically, as described below
2	One-to-many	ICD-9-CM	Can be mapped automatically, as described below
3	Alternate on-to-one maps	None known of	Can be mapped automatically. Further definition will be given if necessary.
4	Alternate one-to-many	None known of	May need manual intervention to map.

For *CrossMaps* that have a *MapSetType* of either '1' or '2', first, create a *concept* under the |Complex map| *sub-hierarchy* to describe the Complex map *Reference Set*, in the following location:

SNOMED CT Model *component*

Foundation metadata *concept*

Reference Set

Complex map

MapSetRealmId

Where *MapSetRealmId* is set to the contents of the *MapSetRealmId* field in the CrossMapsSets record of the *CrossMap* to be represented in *RF2* format. Where the *MapSetRealmId* field is blank or null, then an intermediate *concept* should not be created, and the *Map Reference Set concept* should be created as a direct *child* of |Complex map|. The *concept* should be created as follows:

Table 307: RF2 Map Versioning

Field	Data type	Set to
id	*SCTID*	A unique id in your *namespace*.

Field	**Data type**	**Set to**
effectiveTime	*Time*	The nominal date of release for your *Map Reference set*. The year of the nominal release should tie up with the year in the MapSetSchemeVersionfield in the *CrossMapSets* record.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .

Once the *concept* is created, add two *Descriptions* for the FSN and a *Synonym*.

Table 308: RF2 Mapping Metadata

Field	**Data type**	**Set to**
id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *reference set*.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
conceptId	*SCTID*	The *Identifier* of the *concept* describing the *Reference Set* that you've just added.
languageCode	*String*	The *language* of the *Description*.
typeId	*SCTID*	Create two *descriptions*, with each of the following types: |FSN|, | *Synonym* |.
term	*String*	*Terms* for the FSN and the *Synonym*. The *Synonym* should be set to the *MapSetName* in the *CrossMapSets* record. The FSN should be set to: MapSetSchemeName + "("+ *MapSetSchemeId* + ")" + " *reference set* (foundation metadata *concept*)".

Finally, add members to the *Reference Set* that you've just created.

To do this, identify each CrossMaps table record with a *MapSetId* that matches the *MapSetId* field in the CrossMapsSets record for the *CrossMap* that you're representing in *RF2*. For each CrossMap table record, identify the related CrossMapTarget record using the *MapTargetId* field in the CrossMaps record. The *TargetCodes* field in the CrossMapTarget record will contain zero or more *target codes*, each separated by a separator character identified by the *MapSetSeparator* field of the CrossMapSets record.

One *Reference Set* member record should be created for each *target code* identified within the *TargetCodes* field, as follows:

Table 309: RF2 Mapping Representation

Field	Data type	Purpose
id	*UUID*	A unique *UUID* for the new member record.
effectiveTime	*Time*	The nominal date of release that this member is to be first introduced in.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
refsetId	*SCTID*	The id of the *concept* that describes the *Reference Set* that you've just created.
referencedComponentId	*SCTID*	Set to the *MapConceptId* in the CrossMaps record.
mapGroup	*Integer*	This field should be set to '1' for the first *target code* within *TargetCodes* field of the CrossMapTargets record. If there is more than one *target code* in the field (separated by a separator character), then this field should be set to '2', '3', etc. For each subsequent code.
mapPriority	*Integer*	'1'
mapRule	*String*	Set to null
mapAdvice	*String*	Set to null
mapTarget	*String*	Set to the *target code* in the *TargetCodes* field of the CrossMapTargets record.

9.2.1.4.21 Release Types

Release Format 1 only supports a single *Release Type* which represented the entire set of currently relevant components. In contrast *Release Format 2* supports three different *Release Types* including a full historical view of all components ever released and a *delta release* that contains only the changes from one release to another.

The *Release Format 2 Specification* describes the *Release types* and the *Terminology Services Guide (7)* provides advice on *importing different Release types*.

9.2.1.4.22 Interchange format

RF2 is conceived as a replacement for the current *Release Format*. It is designed to provide a way to publish releases of *SNOMED CT Release* to implementers and other licensees. There is a close *relationship* between the requirements to support distribution of content and the requirements for exchanging *components* during content development. However, there are also significant differences related to the requirement for additional development information (author, change time, etc) and a need to support work with 'interim' incomplete and unpublished *components* which have not yet been assigned a *SNOMED CT identifier*.

Previous *IHTSDO* work resulted in a draft specification of *SNOMED* Interchange Format (SIF) which addressed some of these issues. Some of the provisions of *RF2* are already supported by SIF but others will require revisions to the SIF specification.

9.2.1.4.23 Post Coordinated expression Syntax

RF2 allows *relationship types* to be extended from "existential qualification" to other types of *relationship* such as "universal qualification". This *extension* will not be used in initial releases until the complexity of the underlying semantics has been fully tested, but once it is introduced, post coordinated *expression* syntax will also need to be extended to cater for this.

9.2.1.5 RF1 Compatibility and Conversion Tools

In January 2012 the *IHTSDO* switched from the original *Release Format* (used for *SNOMED CT* distribution since 2002), to the more flexible and consistent *Release Format 2* (RF2). This means that from that date onward the primary source data for the *SNOMED CT International Release* is maintained and distributed in the RF2 format.

The *IHTSDO* recognizes that, while implementers will wish to benefit from the features of the new format, there is inevitably a transitional period during which both format are in use. Therefore, the *IHTSDO* provides the following resources to support users whose system do not yet support *SNOMED CT Release Format 2*:

- *Release Format 1* files will continue to be included in the *International Release* for a limited period
  - These files are not the authoritative version of *SNOMED CT* but are generated from the authoritative RF2 data using a software utility developed for this purpose.
  - The resulting RF1 data retains the functionality of the original release data but does not support any of the features of RF2. While all the clinically relevant *SNOMED CT* hierarchies are identical in both releases, the additional "Metadata Hierarchy" added as part of the RF2 upgrade is not included in the RF1 converted data. In addition there are some cases where *Identifiers* of RF1 derivatives (Subsets and *Cross Maps*) differ from those used for the equivalent *Reference Sets* in RF2. These differences are an essential consequence of ensuring that the RF1 data produced by conversion from RF2 is fully compatibility with existing RF1 systems.
- The RF2 to RF1 Conversion Tool used for generating the RF1 files is also available to all *IHTSDO Members* and *Affiliate Licensees*
  - The "RF2 Conversion Tool" is an open source, Java-based, software tool to facilitate the conversion of *SNOMED CT* files released in RF2 format into RF1 format. The tool provides both a command line utility and a Graphical *User Interface* (GUI) to facilitate configuration, progress tracking and the maintenance of additional data whenever it is not available as part of an RF2 release.
  - The limitations of RF2 to RF1 conversion (noted above) will also apply to conversion undertaken using this tool. To enable the conversion to be completed successfully in a way that retains and replaces *Identifiers* consistently for the RF1 environment a set of auxiliary files (the "RF1 Compatibility Package") is also required.

The "RF2 to RF1 Conversion Tool" and the "RF1 Compatibility Package" are available for *IHTSDO Members* and Affiliates to download in the same way as the *SNOMED CT International Release*.

Caution:

These resources and tools are intended for use during a transitional period and should not be considered as a long term alternative to migration to support direct use of RF2 data within applications. As *SNOMED CT* continues to evolve more of the specific feature of RF2 will be used to add value to the terminology. Some of the added value delivered by RF2 is soon likely to be regarded as essential for effective solutions to user requirements.

9.2.1.5.1 Relationship Refinability Reference Set

The *Relationship Refinability Reference Set* is included in the RF1 Compatibility Pack. It provides information about whether it is permissible to refine the value of a *Relationship*. This information is equivalent to the *Release Format 1 Relationships. refinability* field. It is not included in the main RF2 release as it value is likely to diminish over time as the *Machine Readable Concept Model* provides a more complete representation of *refinability*.

This *Reference Set* is identified as 900000000000488004 | Relationship refinability attribute value reference set (foundation metadata concept) | and its *Concept enumeration* values are specified in Table 310.

Table 310: Refinability value (foundation metadata concept) (900000000000226000)

Id	Term	Comment
900000000000007000	| Not refinable (foundation metadata concept) |	The value provided by the *destinationId* may be used but none of the *subtypes* of this *concept* are permitted.
900000000000218008	| Mandatory refinability (foundation metadata concept) |	The value may be refined by selecting a *subtype* of the *concept* referred to by the *destinationId*.
900000000000216007	| Optional refinability (foundation metadata concept) |	The value may be refined by selecting a *subtype* of the *concept* referred to by the *destinationId*.

9.2.2 SNOMED CT identifier Update Notes

These notes provides update guidance on a change to the management and usage of *SNOMED CT identifiers* agreed during 2011. The resulting changes to specifications and associated implementation guidance have been incorporated within the relevant sections of the Technical Implementation Guide from 2012-01-31.

The change described by this note is designed to remove the need for changing *SNOMED CT Identifiers* when transferring responsibility for maintenance and distribution of a *SNOMED CT component* from an *Extension* to the *International Release*, while maintaining an effective track of the origin and maintenance responsibilities for each *component*.

In addition, the change also remove the need for changing *SNOMED CT Identifiers* when transferring responsibility for maintenance and distribution of a *SNOMED CT component* from an *Extension* to an *Extension* that is a formally recognized hierarchical-ancestor of the originating *Extension*.

Rationale for the Change

SNOMED CT identifiers are unique integer *Identifiers* which include embedded information about the type and origin of the *components* they identify. One part of this embedded information is the *namespace-identifier* which identifies the *Extension* in which the *component* originated.

Prior to the change described by this note the *namespace-identifier* also determined the organization responsible for maintaining the *component*. As a consequence, the specifications required that whenever responsibility for maintenance of *component* was transferred this required it to be inactivated and replaced by a new *component* with a new *SCTID* with a *partition identifier* and namespace-*Identifier* appropriated to the new maintenance arrangement.

The *Identifier* change resulting from moving a *component* from an *Extension* to the *International Release* causes disruption in the authoring environment. From an implementation perspective several *SNOMED CT identifiers* changed from one release to the next, without any change in intended meaning, as a result of adoption of *concept* from an *Extension* as part of the *International Release*. These changes had a negative impact on system operation and interoperability between systems.

Description of the Change

The *namespace-identifier* continues to identify the *Extension* in which the *component* originated. However, it no longer implies a permanent immutable responsibility for maintenance. Instead, within specified limits and with agreement between the responsible organizations, the maintenance responsibility may be reassigned without issuing a new *Identifier*.

The permitted reassignments of responsibility are limited to ensure that the organization responsible for maintaining a *component* can be determined. Thus the end result of any transfer of responsibility must result in the *component* being maintained by one of the organization responsible for one of the following:

- the *Extension* namespace specified by the *namespace-identifier* of its *Identifier*;
- an *Extension* with a *namespace-identifier* that is a hierarchical ancestor of the *namespace-identifier* of the originating *Extension*;
- the *International Release*.

The values of the of the *partition-identifier* which previously indicated that a *component* was part of an *Extension*, continue to indicate that the *SCTID* contains a *namespace-identifier*. However, some *components* with a *namespace-identifier* may now be maintained as part of the *International Release*. Therefore, for clarity, the definition of *partition-identifier* has been revised to indicate that the values determine whether the *SCTID* conforms to the *long format* (with a *namespace-identifier*) or the *short format* (without a *namespace-identifier*). Only the *IHTSDO* can issue *short format SCTIDs* (without a namespace), whereas an *Extension* owner must issue a *long format SCTIDs* (including a *namespace-identifier* that is registered as belonging to them).

The *moduleId* field, introduced in *Release Format 2* and held against each *component*, records the organization currently responsible for maintaining the *component*. The *moduleId* must refer to a module delivered by the organization maintaining the *component* and the *namespace-identifier* of this *moduleId* must belong to the maintaining organization.

Following the change, migration of *components* between *Extensions* would be possible without a change to their *SCTIDs*, according to the following rules:

- A *component* can be moved from any *Extension* to the *International Release* without a change to its *SCTID*.
- A *component* can be moved from an *Extension* to a parent or ancestor *Extension* without a change to its *SCTID*.
- A *component* can be moved from the *International Release* back to its originating *Extension* without a change to its *SCTID*.

In all other cases, the existing rules for moving *components* between *Extensions* should be used. These require a change of *SCTID* to occur with tracking of inactivation in the appropriate *Component Inactivation Reference Set* and cross-references created in the appropriate *Historical Association Reference Set* (or as *historical relationships* in *Release Format 1*).

Caution: *Components* that originated in the *International Release* must **not** be moved to any *Extension* without a change to the *SCTID*.

In order to make explicit which *Extensions* are parents of which other *Extensions*, *concepts* under the *Namespace Concept* may now be rearranged as a nested hierarchy of namespaces. All namespaces at the top level of this hierarchy are considered to have as their parent the *International release*. The *Namespace Concept* for an *Extension* that is dependent on another *Extension* may be nested as a child (sub-type) of the *Namespace Concept* for the *Extension* on which it depends.

Caution: *Components* that originated in an *Extension* must **not** be moved to any other *Extension* unless the *Namespace Concept* associated with the target *Extension* is an ancestor of *Namespace Concept* associated with source *Extension*. The ancestry of *Namespace Concepts* is determined by the *subtype hierarchy* distributed as part of the *International Release*. Other moves between *Extensions* require a change of *SCTID*.

Guidance has been developed for producers and consumers of *SCTIDs*, to help avoid conflicts of ownership and to facilitate identification of owning organizations (see *Guidance for Producers of SNOMED CT Identifiers* and *Guidance for Consumers of SNOMED CT Identifiers*).

Benefits of the Change

The key benefits of making this change are:

- Large scale retirement and replacement of *SCTIDs* place an increased maintenance burden on implementers with no perceivable benefit. This change significantly reduces that burden.
- The change maintains the distinction of the namespace and module *Identifiers* - the former for the creators of content and the latter for the maintainers.
- The change eases the burdens of content providers in the chain of submissions to National *Extension* and the *IHTSDO* in detecting their content in public releases. It enables them to set policies on how to detect and manage content migration.
- Long term contributions will come from existing *Extensions*. This change will reduce impact on both the *National Release Centre Extension* managers and the source providers.
- The change removes the disincentive to migrate content to the *International release* or to a parent *Extension*.
- It will enable more frequent incremental release of content due to decreased migration burden.

9.3 Migrating Existing Data

The transition to *SNOMED CT* from legacy code systems [37] requires several changes. Many of the most important changes relate to organization and user training, which are outside the scope of this technical guide.

From the technical perspective, there are two principal *migration* issues.

- Maintenance of the integrity and value of pre-existing data recorded using other coding schemes (legacy data).
- Maintenance and development of the functionality delivered by software applications that use queries and protocols that include or refer to codes in other coding schemes (legacy queries and protocols).

9.3.1 Intended audience

The intended audience for this section is individuals or any organizations that wish to develop and deploy systems that use *SNOMED CT* but who currently have existing data represented using other coding schemes. This section is therefore contains information that is relevant to various people including:

- **Clinical software developers**, including those who have worked with a version of the Clinical *Terms* (the *Read Codes*) or with *SNOMED CT* terminologies;
- **Clinicians**, whose patient data has been stored in non - *SNOMED CT* systems and who rely on reports and decision support from these systems;
- **Healthcare planners, managers and information specialists** who rely on the secondary use of coded clinical information.

9.3.2 Migration requirements

Migration is required to enable information originally recorded prior to introduction of *SNOMED CT* to be retrieved and reused within a *SNOMED CT enabled application*.

Types of information that need to be considered as part of the migration process include:

- Coded data stored in existing systems.

[37] A "Legacy code system" is code system used prior to implementation of SNOMED CT.

- Information systems, e.g. software and hardware.
- Decision support protocols.
- Data entry templates.
- Queries and other data retrieval, aggregation and analysis specifications.

9.3.3 Strategies for migration

Moving from a legacy coding scheme to *SNOMED CT* requires attention to be paid to continued accessibility and use of data encoded using the legacy scheme.

The following general approaches may be applied or adapted taking accounts of the capability of the *SNOMED CT enabled application* and the value and relevance of existing data.

- Mapping or converting the data:
  - This requires each instance of a code in the existing data to be mapped to an appropriate *SNOMED CT expression*. This *expression* is then associated with the existing coded record entry as part of a record entry that conforms to the information model used in the *SNOMED CT enabled application*.
- Linking or integrating existing data:
  - The existing data is retained in its native form or in an intermediate form. However, the *SNOMED CT enabled application* is designed (or adapted) to access this existing information as if it had been converted. To deliver this functionality, queries and/or data are in effect mapped at the time of retrieval, rather than at the time of upgrading the system. This can be achieved in different ways some of which involve direct use of mapping tables and others in which, while the existing data is unchanged, an index derived from a map to *SNOMED CT* is generated to optimize subsequent access.
- Archive or retain old data in its original form, and where it is necessary to retrieve historical information, use *components* from the legacy system to do this:
  - This approach completely separates the new *SNOMED CT* from the legacy data and is unlikely to be acceptable in clinical practice . However, it may be appropriate for some data warehousing applications where the wholesale conversion of data is considered too onerous.

9.3.4 General considerations for data migration

A substantial body of clinical information resides in electronic systems, represented using existing coding schemes, terminologies and classifications. This information may be of value to individual patient records or to population aggregations. Similarly, there are many queries and decision support protocols that contain knowledge representation based on existing terminologies.The volume and heterogeneous nature the existing data means different approaches may be required to meet specific sets of requirements.

Factors that need to be considered include:

- The volume and value of existing in the context of the anticipated uses of a future *SNOMED CT application*:
  - The scale of the task and the potential value of migrated data are interrelated. Relatively small amounts of data that are of debatable value to a future system may not justify an elaborate migration process. On the other hand, it is vital to ensure that valuable existing data remains fully accessible within a *SNOMED CT enabled application*.

Example: In the UK alone, there are over 50 million patients primary care electronic records coded using one of the versions of the *Read Codes*. Based on typical patterns of use this means there are several billion coded record entries that may need to be taken into account in the migration process. A substantial proportion of this data has continuing clinical value and thus despite the scale of the task it is important to ensure that data is migrated accurately and efficiently.

- Data quality and consistency:
    - Different users in different settings may select codes and terms in idiosyncratic ways to reflect their needs. This may be acceptable locally but it creates an obstacle to migration if the goal is consistent and comparable information at a regional, national or global level.
- Different source code systems:
    - Several different coding scheme versions are in use and each of these poses specific challenges and offers a different profile of potential benefits.
- Different information systems:
    - There are many system suppliers. As a result of system development and commercial mergers and takeovers, many suppliers support more than one application in the same domain. The challenge is to migrate from this diverse situation to a next generation environment supporting standards such as *SNOMED CT*.
- Different information models:
    - In addition to differences in the use of codes, existing systems inevitably have a variety of approaches to structuring clinical information. As a result, the process of migrating data between systems is not simply a question of converting codes. The underlying architecture of the source data also needs to be taken into account to make optimal use of existing data without losing processable information or introducing errors.

9.3.5 Specific data migration issues

9.3.5.1 Retaining existing coded data

Migration does not mean over-writing legacy coded data with *SNOMED CT expression*. **This is strongly discouraged** and users are advised to ensure that data stored at the time of data entry is preserved. This is essential for two main reasons:

- Medico-legal status of an altered clinical record may become degraded;
- The original record may be an invaluable resource, should *migration* produce unexpected results.

9.3.5.2 Hierarchies and Identifiers

The use and representation of hierarchies in *SNOMED CT* differs from the approach taken in many older code systems and classifications. This has a number of consequences that may affect the migration of queries, decision support protocols and data entry templates.

- Meaningless *Identifiers*:
    - The codes specified in ICD-9, *ICD-10*, the *Read Codes* and *SNOMED International* provide information about where the code is located in a hierarchy. This allowed simple pattern matching queries to be used for some types of retrieval.

      Example: In *SNOMED International*, all 'diagnoses related to the digestive system' can be retrieved by a query for all codes starting with 'D5'

      The *Identifier* of a *SNOMED CT concept* does not provide any information about the way it relates to other *concepts*. Therefore, a simple pattern matching query cannot be used to retrieve related information represented using *SNOMED CT*. Instead, the query that specifies a *subtype* of the required *concept* is evaluated by testing the *transitive closure* of the set of *subtype Relationships*.

      Example: All 'diagnoses related to the digestive system' can be retrieved by a query for *expressions* that are *subtypes* of 119292006 | disorder of gastrointestinal tract |.

- Polyhierarchy:
    - *Statistical classifications* and many other code systems have a *monohierarchy* in which each code falls within only one branch of the hierarchy. In contrast, the *SNOMED CT subtype hierarchy* is a *polyhierachy*, which means that each *concepts* can by a *subtype* of many different *concepts*. This is a powerful feature of *SNOMED CT* but it may significantly alter the results of a migrated from an earlier scheme.

      Example: All 'diagnoses related to the digestive system' is some schemes may exclude codes that are primarily classified as infective disorders even they affect the digestive system. However, a *SNOMED CT* query includes all *subtype concepts* regardless of whether they are also in another hierarchy.

- Different hierarchies:
    - Hierarchies in different code systems may be based on different principles and as a result queries that are migrated to *SNOMED CT* may return unexpected results. This may also relate to different interpretation of apparently identical *concepts*.

      Example: Should a query for the *concept* 'nephrectomy' return only patients who have had a total nephrectomy or should the results include those with a record of a partial nephrectomy? If the partial nephrectomies are to be included then how much kidney tissue must be removed to count as a nephrectomy? If the partial nephrectomies are not included then should a record entry including the *concept*'nephrectomy' (without specifying partial or total) be included? The *SNOMED CT subtype hierarchy* determines the answer to these questions based on the defining *Relationships*. The query may need to be refined to meet more specific expectations.

9.3.5.3 postcoordination

SNOMED CT, enables the use of *postcoordinated expressions* to represent detailed clinical information (such as observations or procedures) by reference to multiple *Concept Identifiers*.

When considering migration of existing data an important question is whether *postcoordination* is required to replace existing coded data. The answer to this question depends on the specificity and expressivity of the existing coding scheme.

There are four situations in which *postcoordination* may be required or useful in the mapping process.

- To capture data *postcoordinated* in the original coding scheme:
    - The extent to which this data can be mapped to *SNOMED CT* depends on the consistency of the original representation and the degree of alignment with the *SNOMED CT Concept Model*. Refinements in the source data are not sanctioned by the *Concept Model* may be mapped to similar approved *Attributes* or downgraded to text. Alternatively, they may be retained as *expressions* that fall outside the scope of *Concept Model* although this may limit effective retrieval.

  Example: Information coded using *NHS Clinical Terms Version 3*, may also include *postcoordination* using *qualifiers*. In most cases, these *qualifiers* can be represented using *postcoordinated expressions*.

- To represent information which the originating coding system precoordinates but which *SNOMED CT* can only represent using a *postcoordinated expression*.

  Example: A specialized radiology coding systems may have separate codes for the same procedure applied to different body sites. If *SNOMED CT* does not include these, they can be represented using *postcoordinated expressions* with 'procedure site' and 'laterality' *Attributes* applied.

- To incorporate additional information which the originating clinical system represents in a consistent proprietary form.

Example: A system may have a separate field for laterality and this can be applied during the mapping process to generate a *postcoordinated expression*.

- To satisfy a preference for a consistent *postcoordinated* representation of a particular type of data.

Example: There may be a preference to always represent allergies by postcoordinating the substance, rather than using one of the *precoordinated* 'allergic to x' *concepts*.

9.3.6 Migration from earlier SNOMED CT code systems

This section contains specific advice related to migration to *SNOMED CT* from previous *SNOMED CT* code systems including *SNOMED RT* and *SNOMED International*.

9.3.6.1 Migration from SNOMED RT

Migration from *SNOMED RT* poses very few significant issues, since many features of the design of *SNOMED CT* including the use of *SCTIDs* were incorporated into *SNOMED RT*.

The transition to *SNOMED CT* for users of *SNOMED RT* is relatively straightforward because the *Concept Identifiers* of *SNOMED RT* are for the most part the same as those used in *SNOMED CT*. In some cases, during the merger of *SNOMED RT* and *Clinical Terms Version 3* some *Concepts* in *SNOMED RT* have been found to be ambiguous or duplicated. These *Concepts* have been inactivated by an appropriate change of status and adding a record in the Component Inactivation *Reference Sets*, but are still present in the *Concepts Table* and are linked to *Active Concepts* by the Historical Associations *Reference Set*:

- Each duplicate *Concept* has a | SAME AS | Association to the *Active Concept* with the same meaning as the duplicate *Concept*;
- Each ambiguous *Concept* has | MAY BE A | Association to one or more *Active Concepts*, which represent possible disambiguated meanings.

These Associations can be used either to allow *Concepts* recorded using these *Concepts* to be recognized by retrieval tools or to enable mapping of the stored information to the appropriate *active Concept Identifier*.

If any stored *Concept Identifier* of an *Inactive Concept* is mapped to an *active Concept Identifier* using these Associations it is strongly recommended that the original *Concept Identifier* is also retained. This enables future improvements or corrections of such mappings if revised Associations are present in a future release of *SNOMED CT*.

In addition, *SNOMED RT* contained both generic and brand name drugs for the US. A decision was made during the merger process to not retire these *concepts* using the *extension* mechanisms, but to place these components directly in the US Drug *Extension*. Therefore to access all *SNOMED RT components* you will need to use the US Drug *Extension* in addition to the *SNOMED CT International Release*.

Like *SNOMED CT*, *SNOMED RT* also contains the appropriate legacy *SNOMEDID* from *SNOMED International*.

9.3.6.2 Migration from SNOMED International

The meaning of coded clinical data encoded using *SNOMED International* is maintained using *SNOMED CT* mechanisms that support *concept* permanence and version control. Even when a *Concept* is *retired* from *active* use its code is never reassigned to another *Concept*.

Concept permanence ensures that codes assigned in *SNOMED International* are retained, accessible, and not reused. The codes used in *SNOMED International* are present in *SNOMED CT* in the 900000000000498005 | SNOMED RT ID simple map reference set | which links the old alphanumeric codes to the *SNOMED CT concept identifiers*. For further information see the specification of *Simple map reference sets*.

The 900000000000498005 | SNOMED RT ID simple map reference set | can be used either to allow recognition of legacy data by *SNOMED CT* retrieval tools or to enable mapping of the codes and storage of the appropriate *Concept Identifier*.

9.3.6.3 Migration from SNOMED 2 and SNOP

For information on migration from SNOMED International (SNOMED 3 to 3.5) see *Migration from SNOMED International*.

To assist in the *migration* of legacy data from *SNOMED* II (1979) and SNOP (1965), a "Bridge File" (mapping table) is available which links each legacy code to its corresponding code in the first release of *SNOMED CT* (note these are same identifier as in *SNOMED RT*).

The bridge file is included in the *SNOMED CT International Release* distribution. The file are locating the following subdirectory:

- SnomedCT_Release_INT_[YYYYMMDD]\RF1Release\OtherResources\BridgeFiles.

The file for mapping from SNOMED 2 is in the zip archive:
zres_BridgeFile_Snomed2ToSnomedRT_INT_20020131.zip

- File name: SNO2-SRT10_Bridge.txt
- This is a tab delimited file with column headings
- Column 1: CODE is the SNOMED code
- Column 4: ConceptId is the SNOMED CT Concept identifier
- (note that the target ConceptId in SNOMED RT are the same as the identifiers in SNOMED CT)

9.3.7 Migration from Read Codes and CTV3

Read Codes and *Clinical Terms Version 3* codes are present in *SNOMED CT* in the 900000000000497000 | CTV3 simple map reference set | which links the old alphanumeric codes to the *SNOMED CT concept identifiers*. For further information see the specification of *Simple map reference sets*.

The 900000000000497000 | CTV3 simple map reference set | can be used either to allow recognition of legacy data by *SNOMED CT* retrieval tools or to enable mapping of the codes and storage of the appropriate *Concept Identifier*.

Attention: Organizations based in the UK that are planning to migrate from NHS *Clinical Terms Version 3* or earlier versions of the *Read Codes* are advised to review the documentation and mapping tables published by the *UK NHS*. These resources support more sophisticated mapping based on use of terms and patterns of use that meet UK requirements. Separate advisory documents and tables for each of the *Read Code* versions are available as part of the NHS Terminology Reference Data Update Distribution Service (TRUD) *https://www.uktcregistration.nss.cfh.nhs.uk/trud/*. These materials are updated with each *SNOMED CT* UK *Extension Release*.

Chapter 10

10 Extension Services Guide

This part of the guide describes additional services which some advanced users or implementers may require to allow them to create or maintain *Extensions* for use in a particular country, organization or specialty.

The most common of these requirements will be to support the creation and maintenance of specialized *Reference sets*. Uses for *Reference Sets* include representation of *value sets*, marking *descriptions* to indicate acceptability of *terms* in a specific *language* or specialty, alternative hierarachies, *maps* to classifications and *annotations*.

10.1 Rationale for Extensions

The *SNOMED CT Extension* mechanism allows authorized organizations to add locally valid *components* and *Reference Sets* without compromising the *SNOMED CT International Edition*. This facility will be valuable to:

- Meet the needs of specialties and *realms*;
- Meet vendor needs;
- Meet local business needs.

10.2 Extension Namespaces and SNOMED CT identifiers

The *components* of an *extension* have *identifiers* (*SCTIDs*) which have the same structure as those used in the *SNOMED CT International Release*. However, these *identifiers* include a *partition-identifier* indicating that the *component* is part of an *extension* and a *namespace identifier* specific to the responsible organization .

Partition-identifiers and *namespace identifiers* serve two roles:

- Prevention of *identifier* collision or reuse:
  - Organizations responsible for an *extension* must only issue *components* within their allocated *namespace* and must not reuse any *identifier* within that *namespace* once it has been issued.
- Indicating the origin of an identified *component*:
  - If an application receives instance data containing an *identifier* that it does recognize , the application can use the *namespace identifier* to determine the responsible issuing organization .
  - The responsibility for an allocated *namespace* remains with the organization to which it was issued unless responsibility is transferred by merger or mutual agreement. Any *namespace* transfer must be notified to and authorized by the *IHTSDO*.

10.3 Moving components between namespaces

The *namespace-identifier* identifies the *Extension* in which the *component* originated. However, within specified limits and with agreement between the responsible organizations, the maintenance responsibility may be reassigned without issuing a new *Identifier*. Therefore, the *namespace-identifier* may not identify the *Extension* in which a *component* is now maintained.

The permitted reassignments of responsibility are limited to ensure that the organization responsible for maintaining a *component* can be determined. Thus the end result of any transfer of responsibility must result in the *component* being maintained by the organization responsible for one of the following:

- the *Extension* namespace specified by the *namespace-identifier* of its *Identifier*;
- an *Extension* with a *namespace-identifier* that is a hierarchical ancestor of the *namespace-identifier* of the originating *Extension*;
- the *International Release*.

The values of the of the *partition-identifier* which previously indicated that a *component* was part of an *Extension*, continue to indicate that the *SCTID* contains a *namespace-identifier*. However, some *components* with a *namespace-identifier* may now be maintained as part of the *International Edition*. Therefore, for clarity, the definition of *partition-identifier* indicates whether the *SCTID* conforms to the *long format* (with a *namespace-identifier*) or the *short format* (without a *namespace-identifier*). Only the *IHTSDO* can issue *short format SCTIDs* (without a namespace), whereas an *Extension* owner must issue a *long format SCTIDs* (including a *namespace-identifier* that is registered as belonging to them).

The *moduleId* field, introduced in *Release Format 2* and held against each *component*, records the organization currently responsible for maintaining the *component*. The *moduleId* must refer to a module delivered by the organization maintaining the *component* and the *namespace-identifier* of this *moduleId* must belong to the maintaining organization.

Following the change, migration of *components* between *Extensions* would be possible without a change to their *SCTIDs*, according to the following rules:

- A *component* can be moved from any *Extension* to the *International Release* without a change to its *SCTID*.
- A *component* can be moved from an *Extension* to a parent or ancestor *Extension* without a change to its *SCTID*.
- A *component* can be moved from the *International Release* back to its originating *Extension* without a change to its *SCTID*.

In all other cases, the existing rules for moving *components* between *Extensions* should be used. These require a change of *SCTID* to occur with tracking of inactivation in the appropriate *Component Inactivation Reference Set* and cross-references created in the appropriate *Historical Association Reference Set* (or as *historical relationships* in *Release Format 1*).

Caution: *Components* that originated in the *International Release* must **not** be moved to any *Extension* without a change to the *SCTID*.

In order to make explicit which *Extensions* are parents of which other *Extensions*, *concepts* under the *Namespace Concept* may now be rearranged as a nested hierarchy of namespaces. All namespaces at the top level of this hierarchy are considered to have as their parent the *International release*. The *Namespace Concept* for an *Extension* that is dependent on another *Extension* may be nested as a child (sub-type) of the *Namespace Concept* for the *Extension* on which it depends.

Caution: *Components* that originated in an *Extension* must **not** be moved to any other *Extension* unless the *Namespace Concept* associated with the target *Extension* is an ancestor of *Namespace Concept* associated with source *Extension*. The ancestry of *Namespace Concepts* is determined by the *subtype hierarchy* distributed as part of the *International Release*. Other moves between *Extensions* require a change of *SCTID*.

Guidance has been developed for producers and consumers of *SCTIDs*, to help avoid conflicts of ownership and to facilitate identification of owning organizations (see *Guidance for Producers of SNOMED CT Identifiers* and *Guidance for Consumers of SNOMED CT Identifiers*).

10.3.1 Guidance where RF1 format is used for an Extension

Where a namespace owner is still releasing an *Extension* using *Release Format 1* (RF1), then content should continue to be moved from and to that *Extension* by creating new *SNOMED CT identifiers* and using the old "move to / move from" mechanism.

Consumers of *RF1* releases or conversions of *Extensions* should be aware that content may have been moved to the *International Release*, or to a parent *Extension*, without a change of *SCTID*. Therefore, these *Extensions* and so may contain *components* from with different.

10.4 Guidance for Consumers of SNOMED CT Extensions

10.4.1 Guidance on validating SCTIDs within an Extension

The following checks may be performed to validate the consistency of *SCTIDs* in one or more *Extensions*:

- An *Extension* should only contain *components* that have a namespace owned by the releasing organization, or a child of a namespace owned by the releasing organization. Note however, that a releasing organization may merge content from its *Extension(s)* with one or more parent *Extensions* and the *International release* into a single *release file*.
- The primary key for *component* versions held as rows in *release files* is the composite of the *SCTID* and the *effectiveTime*. No two *component* versions should have the same primary key, either within or across all *Extensions*. Once loaded, the state valid history of a *component across all loaded Extensions* should be taken in the normal *effectiveTime* order.
- If a child *Extension* releases a new version of a *component* that has not been inactivated within the parent *Extension*, then there is an error. The version of the *component* in the parent *Extension* should be taken as the correct version of the *component* (as they have not formally released control of it), and the error should be reported to the owner of the child *Extension*.
- The *check digit* of each *SCTID* may be validated using the *check-digit* algorithm.

The following provides examples of possible errors that can be picked up as part of a validation process:

Here, the *release file* contains a *concept* that has a namespace that is not a child or parent namespace of namespace 0009999.

Extension for namespace 0009999:

SCTID	***effectiveTime***	**Active**	***moduleId***	***definitionStatusId***
1290989121103	*20071031*	1	|Module 1|	|*Primitive*|

Here, the *concept* has been incorrectly inactivated in an *Extension* at the same *effectiveTime* as the new *concept* version has been included in the *International release*. A clash in primary keys of the two *concept* versions has resulted.

Extension for namespace 0989121:

SCTID	***effectiveTime***	**Active**	***moduleId***	***definitionStatusId***
1290989121103	*20071031*	1	|Module 1|	|*Primitive*|

1290989121103	*20080131*	0	|Module 1|	|*Primitive*|

International release:

SCTID	***effectiveTime***	**Active**	***moduleId***	***definitionStatusId***
1290989121103	*20080131*	1	|Module 2|	|*Primitive*|

Here, the *concept* has not been deactivated in the *International release* before it was reinstated in an *Extension*. This is an error that would result in consumers of the *International release* receiving the 31st January version of the *concept*, with consumers of *Extension* 0989121 (and the *International release*) receiving 31st October version of the *concept*, resulting in risks to semantic interoperability. In this case, the 31st January version included in the *International release* should be taken as the correct version and the error should be reported to the owner of *Extension* 0989121.

Extension for namespace 0989121:

SCTID	***effectiveTime***	**Active**	***moduleId***	***definitionStatusId***
1290989121103	*20071031*	1	|Module 1|	|*Primitive*|
1290989121103	*20081031*	1	|Module 1|	|*Primitive*|

International release:

SCTID	***effectiveTime***	**Active**	***moduleId***	***definitionStatusId***
1290989121103	*20080131*	1	|Module 2|	|*Primitive*|

10.4.2 Guidance on identifying the maintaining authority for a component

Where information about a *component* is available (either from *release files* or from a *terminology server*), then the *moduleId* of the *component* can be used to identify its maintaining authority. As an example, take the following case, where a *concept* was created by the owner of namespace 0989121, but then subsequently transferred to *IHTSDO*:

Extension for namespace 0989121:

SCTID	***effectiveTime***	**Active**	***moduleId***	***definitionStatusId***
1290989121103	*20071031*	1	|Module 1|	|*Primitive*|

International release:

SCTID	***effectiveTime***	**Active**	***moduleId***	***definitionStatusId***
1290989121103	*20080131*	1	|Module 2|	|*Primitive*|

If a consumer wishes to know who is currently responsible for maintaining the *concept* with *SCTID* 1290989121103, then the *release files* may be used to establish ownership. As can be seen, the *moduleId* of the most recent version of the *concept* is |Module 2|, which would be identifiable as belonging to *IHTSDO* as it is a short format *SCTID* (which does not have a namespace). Similar information should also be available via *terminology servers*, where such services are available.

If only the *SCTID* is available, then the namespace embedded in the *SCTID* can be used to check ownership of a *component*. In that case, the maintaining authority is most likely to be the organization that owns the namespace of the *component*. If it is not, then the organization that owns the next namespace up in the namespace hierarchy should be checked, where *IHTSDO* is positioned as the ultimate parent in the namespace hierarchy.

10.4.3 Guidance on parsing and identifying SCTIDs

The constraints on the value range for *SCTIDs* allow a consistent *string* and integer representation of these values. The upper limit of 18 digits ensures that any valid *SCTID* can be stored in either a signed or unsigned 64-bit integer. The lower limit of six digits ensures that a *SCTID* can be distinguished from:

- A *Read Code*, which is 5 characters in length, padded out with dots if necessary.
- A *SNOMED* ID, which always starts with a letter.

10.4.4 Guidance on using state valid data

When receiving data from an *Extension* owner, care should be taken when reviewing historical data only to use snapshots of data relating to one of the release points for that *Extension*. It is only at these release points that the content in the *Extension* is consistent with the content in the *International release* and/or any parent *Extensions*.

For example, take the case where the *International edition* is released in January and an *Extension* is released in April. Generally, the *Extension* will be dependent on the *International release*, and may, for example, hold a *concept* that is a child of a parent *concept* in the *International release*. Now, if the parent *concept* is amended or even retired in January, the child *concept* will need to be reviewed, modified and perhaps moved to another part of the hierarchy to take account of the changes in the *International release*. Generally, the reason that the *Extension* is released a few months after the *International release* and not earlier is that the *Extension* owner needs to review the changes in the *International release*, modifying the content in the *Extension* to keep it consistent. Once the April release is made, the *Extension* and the *International release* will be consistent, and before the April release, consumers of the *Extension* should use the previous (July) version of the *International release* supplied by the *Extension* owner. So once the April release is made, the *Extension* and the *International release* are consistent, but any historic state between January and April will be inconsistent. In practice, this is not a big issue, as no changes will have been made between January and April.

10.5 Guidance for Producers of SNOMED CT Extensions

Prerequisites

Before generating *SCTIDs*, an organization must own a namespace. A namespace can be requested from *IHTSDO* by emailing a request to *info@ihtsdo.org*.

Guidance on Generating SCTIDs

The following guidance is provided for owners of namespaces that generate new content:

- An organization should only generate new *SCTIDs* for *components* within a namespace that they own.
- An organization should have a mechanism in place to ensure that *SCTIDs* are not assigned multiple times. Generally, a single authority that generates *item-identifiers* in a sequential fashion for each type of *component* will achieve this goal.

- Generally, *SCTIDs* should be generated for new *components* as part of the release process for an *Extension*, rather than during the edit process. This is to avoid unnecessary usage of *Identifiers* for *Concepts* that are created during editing but found not to be required prior to release.
- *item-identifiers* should not be generated so as to have meaning. They should be regarded as meaningless numbers.

Guidance on Packaging Content

Organizations may package content into *release files* in a number of ways:

- All content for a particular type of *component* (e.g.: of type *Concept*) that is owned by the organization can be released in a single file. *Components* in this file may have different *moduleIds*, where the content has been authored by more than one group in the organization and each group has its own *moduleId*. Content that is owned by parent organizations may be held in separate files that are also included in the release. Content owned by other organizations should not be included in the release.
- As above, but *components* with different *moduleIds* can be released in separate files.
- As the first bullet above, but content from parent organizations may be included in the same *release files* as content owned by the releasing organization. In this case, the ownership of each *component* can be identified by reference to its *moduleId*. Care should be taken not to modify, add to or remove content that is owned by a parent organization, as this would be considered as editing content that the organization did not own.

Guidance on Promoting Components

Components (whether *Concepts*, *Descriptions* or *Relationships*) may be promoted to a parent *Extension* or to the *International release*. In order to achieve this, the donating organization should contact *IHTSDO* or the owner of the receiving *Extension* with details of the *components* that are to be promoted.

The definition of the *component* in the source *Extension* should not change (for example, a new record should not be added to the source *Extension* to inactivate the *component*). Once the *component* has been promoted to a parent *Extension*, care should be taken not to amend or inactivate it within its original *Extension*.

A *Component* should only be promoted to the *International release* or to an *Extension* that is associated with a namespace that is a parent of the namespace of the *component*.

Guidance on Receiving Promoted Components

Before receiving content into a parent *Extension* or the *International release*, details of the *components* that are to be transferred should be received in writing from an authority within the source organization.

The *component* should then be included in the next release of the parent *Extension* or of the *International release*, with the following fields amended:

- *effectiveTime* – to be set to the *Extension*'s release date, as normal.
- *moduleId* – set to the *moduleId* of the new maintaining organization.

The *SCTID* of the *component* should not change when it is included in the new *Extension* or the *International release*.

Guidance on Defaulting Components to their Original Extension

Where a *component* has been promoted to a parent *Extension* or to the *International release* (perhaps incorrectly), and it is required to default that *component* back to its original *Extension*, then the parent organization should contact the owner of the *components*' original *Extension* with details of the *components* that are to be moved.

The *component* should then be included in the next release of the parent *Extension* or of the *International release*, with the following fields amended:

- *effectiveTime* – to be set to the *Extension*'s release date, as normal.
- active – set to false.

At this point, the *component* will be retired for all consumers of the parent *Extension* or the *International release*.

The *component* should then be included in the next release of the original *Extension* to which the *component* is to be moved, with the following fields amended:

- *effectiveTime* – to be set to the *Extension*'s release date, as normal.
- *moduleId* – set to the *moduleId* of the new maintaining organization.

The *SCTID* of the *component* should not change when it is included in the receiving *Extension*. Care should be taken to ensure that the *effectiveTime* of the inactivation record in the donating *Extension* is prior to the *effectiveTime* of the record in the receiving *Extension*. The donating *Extension* should always inactivate the *component* before it is included in the receiving *Extension*. In particular, the *effectiveTimes* should not be set to the same date in order to avoid a primary key conflict for the *component* across the *Extensions*.

The following example shows how a *Concept* can be created in an *Extension*, promoted to the *International release* and then be defaulted to its original *Extension* without changing its *SCTID*. In this example, |Module 1| is owned by namespace 0989121 and |Module 2| is owned by *IHTSDO*.

A *concept* is first created in *Extension* 0989121:

Extension for namespace 0989121:

SCTID	*effectiveTime*	Active	*moduleId*	*definitionStatusId*
1290989121103	*20071031*	1	|Module 1|	|*Primitive*|

It is then included in the *International release*. At this stage, *IHTSDO* owns the *concept*. Note that there is no need to deactivate the *concept* in the *Extension* as the *Extension* is dependent on the *International release*, and therefore can only be used in conjunction with the *International release*. Because of the state valid representation of RF2, the new *concept* version added to the *International release* automatically supersedes the previous *concept* version in the *Extension*. Also, |Module 2| supersedes |Module 1| as the new module in which this *concept* is now authored:

Extension for namespace 0989121:

SCTID	*effectiveTime*	Active	*moduleId*	*definitionStatusId*
1290989121103	*20071031*	1	|Module 1|	|*Primitive*|

International release:

SCTID	*effectiveTime*	Active	*moduleId*	*definitionStatusId*
1290989121103	*20080131*	1	|Module 2|	|*Primitive*|

Then, *IHTSDO* inactivates the *concept* within the *International release*:

Extension for namespace 0989121:

SCTID	*effectiveTime*	Active	*moduleId*	*definitionStatusId*
1290989121103	*20071031*	1	|Module 1|	|*Primitive*|

International release:

SCTID	***effectiveTime***	**Active**	***moduleId***	***definitionStatusId***
1290989121103	*20080131*	1	|Module 2|	|*Primitive*|
1290989121103	*20080731*	0	|Module 2|	|*Primitive*|

Finally, the original *Extension* owner can include the *concept* within their own *Extension* again. At this stage, the *concept* will be inactive to all consumers of the *International release* that do not also consume *Extension* 0989121:

Extension for namespace 0989121:

SCTID	***effectiveTime***	**Active**	***moduleId***	***definitionStatusId***
1290989121103	*20071031*	1	|Module 1|	|*Primitive*|
1290989121103	*20081031*	1	|Module 1|	|*Primitive*|

International release:

SCTID	***effectiveTime***	**Active**	***moduleId***	***definitionStatusId***
1290989121103	*20080131*	1	|Module 2|	|*Primitive*|
1290989121103	*20080731*	0	|Module 2|	|*Primitive*|

Note that in the above example, the *concept* must be explicitly inactivated in the *International release* and it is not adequate in this case simply to rely on the new *concept* version that was added in the *Extension* to supersede the old *concept* version in the *International release*. This is because the *International release* is not dependent on the *Extension*, and so consumers that are taking the *International release* without taking the *Extension* need to be aware that, for them, the *concept* has been inactivated.

Guidance on other Movement of Components between Extensions

All other movement of *components* between *Extensions*, including moving content that was created in the *International release* to an *Extension*, or from one *Extension* to another unrelated *Extension* should be performed by inactivating the *component* from the source *Extension* and creating a new *component* in the receiving *Extension*, as is described in the *SNOMED CT* Technical Reference Guide.

The reasons for constraining movement of *components* between *Extensions* in this way are:

a) To ensure that two *components* with the same *SNOMED CT identifier* are not released independently (and perhaps inconsistently) in two separate *Extensions*.

b) To allow a consumer to validate that the owner of an *Extension* has the authority to release all the *components* included in their release.

If it were allowed for a *component* to be retired from the *International release* and moved to an *Extension* while keeping the same *SNOMED CT identifier*, then it would be possible for more than one *Extension* owner to include the *component* in their *Extension* (perhaps over a period of time). This would result in issue (a) above. *IHTSDO* would have no way of monitoring this, or providing guidance to consumers of *Extensions*, to allow them to validate ownership of *components* within an *Extension* (issue b above). More seriously, once a *component* is in two separate *Extensions* with the same *SNOMED CT identifier*, then it may get modified in different ways in each *Extension* over time, causing interoperability issues.

Chapter 11

11 SNOMED CT file and field names

This section lists the file and field names used in technical specifications within this guide. The scope of use of these names is limited to the tables in which they are used and the given definitions are not intended for use in any other context.

A

acceptabilityId (field)

A field in a 900000000000506000 | Language type reference set | that indicates the acceptability of a *Description* in the language or *dialect* specified by that *Reference Set*. Values include "preferred" and "acceptable".

Note: Field name in a 900000000000506000 | Language type reference set |

active (field)

A *Boolean* field that specifies whether an identified *component* or is an *active* from the point in time specified by the *effectiveTime*.

Note: Field name in SNOMED CT Release Format 2.

alternateIdentifier (field)

A field in the *Identifier file* containing the representation of an *Identifier* in another code system that is irrevocably linked to a *SNOMED CT identifier*.

annotation (field)

An Annotation *Reference Set* field containing additional information linked to a *SNOMED CT component*.

Note: Field name in SNOMED CT Release Format 2.

attributeDescription (field)

A reference to a *concept* that specifies the name and/or usage of an additional attribute in a *Refset*. If the *attributeType* is component reference, the values applied to this additional attribute are restricted to *subtypes* of this *concept*.

Note: Field name in a SNOMED CT Release Format 2 Reference Set Descriptor.

attributeOrder (field)

An integer representing the position of an additional attribute in a *Refset*. The value 0 (zero) refers to the *referencedComponentId*. All other values refer to the position of an additional attribute relative to the *referencedComponentId*.

Note: Field name in a SNOMED CT Release Format 2 Reference Set Descriptor.

attributeType (field)

A reference to a *concept* that specifies the data type of an additional attribute in a *Refset*.

Note: Field name in a SNOMED CT Release Format 2 Reference Set Descriptor.

B

Boolean (data type)

A datatype that represents either true or false.

Note: In *SNOMED CT release files* the value 0 (zero) represents "false" and the value 1 (one) represents true.

C

caseSignificanceId (field)

A field in the *Description Release File* containing a *SNOMED CT identifier* that indicates whether the text of the term can be modified to by switching characters from upper to lower case (or vice-versa).

Note: Field name in SNOMED CT Release Format 2

characteristicTypeId (field)

A reference to a *concept* that specifies the nature of a *Relationship*. Values include "defining", "qualifying" etc.

Note: Field name in the SNOMED CT Release Format 2 relationships table.

Concept file

The file structure used to distribute *SNOMED CT concepts*.

Note: Component File name in SNOMED CT Release Format 2

conceptId (field)

A field in the *Description file* that associates a *term* with the *concept* to which it applies .

Note: Field name in the *Description file*.

correlationId (field)

A field in the Complex Map *Reference Set* containing a *SNOMED CT identifier* which represents the correlation between the *SNOMED CT concept* and the *target code*.

Note: Field name in SNOMED CT Release Format 2

D

definitionStatusId (field)

A field in the *Concept Release File* containing a *SNOMED CT identifier* which specifies whether the *concept* is *fully defined* or *primitive*.

Note: Field name in the SNOMED CT Release Format 2 concepts table.

Description file

The file structure used to distribute *SNOMED CT descriptions*.

Note: Component File name in SNOMED CT Release Format 2

descriptionFormat (field)

A *Description Format Reference Set* field reference to a *concept* that specifies the maximum length and format of the *term* fields for a particular type of *Description*.

Note: By default the *term* is a *UTF-8* string of up to 255 characters without markup. However, description types can be specified which are longer in length and/or contain format markup (e.g. HTML). For more details of how this is specified see the file structure specification.

descriptionLength (field)

A *Description Format Reference Set* field containing an integer which indicates the maximum length of the term string for a specified type of *Description*.

Note: By default the *term* is a *UTF-8* string of up to 255 characters without markup. However, description types can be specified which are longer in length and/or contain format markup (e.g. HTML). For more details of how this is specified see the file structure specification.

destinationId (field)

A field in the *Relationship Release File* containing a *SNOMED CT identifier* that refers to the *concept* that represents the destination (or *attribute-value*) of the associated *Relationship*.

Note: Field name in *SNOMED CT Release Format 2*. In RF1 this field was called *ConceptId2*

Dualkey (field)

A key used to facilitate textual searches of *SNOMED CT* that consists of the first three letters of a pair of words in a *Description*. All possible pairs of words in each *Description* may be paired irrespective of their relative position in the *Description*. *Dualkeys* are represented as a row in the *Dualkeys* Table.

Note: Field name in SNOMED CT toolkit

Dualkey table

A table in which each row represents a *Dualkey*. See [see *Word Search Tables - Summary* on page 124].

Note: File or Table name in SNOMED CT toolkit

E

effectiveTime (field)

Specifies the inclusive date at which the component version's state became the then current valid state of the component.

Note: Field name in SNOMED CT Release Format 2

Excluded word (field)

A word that in a given *language* is so frequently used, or has so poor a discriminating power, that it is suggested for exclusion from the indices used to support textual searches of *SNOMED CT*. *Excluded Words* are represented as a row in the *Excluded Words Table*

Note: Field name in SNOMED CT toolkit

Excluded words table

A data table in which each row represents an *Excluded Word*. See [see *Word Search Tables - Summary* on page 124].

Note: File or Table name in SNOMED CT toolkit

I

Identifier file

The file structure used to distribute alternative *Identifiers* for *SNOMED CT components*.

Note: The Identifier file is not currently used in the *SNOMED CT International Release* as use of the more flexible *Simple map type references set* structure is preferred for links to alternative codes. The only known current use of this file is for internal identification of components during the content development process.

id (field)

A field that provides the unique identifier of a *component* (*concept*, *description* or *relationship*) or *reference set member*.

Note:

- The data type of the *id* for a *component* is *SCTID* and this identifier is used to refer to the *component*.

- The data type of the *id* for a *reference set member* is *UUID*. This identifier is only used to support versioning of a rows (*member*) in a *Reference set* it does not identify the Reference set itself (see *refsetId*) nor does it identify to a component refered to by the *Reference set* (see *referencedComponentId*).

identifierSchemeId (field)

A field in the RF2 *Identifier file* containing a *SNOMED CT identifier* which identifies the alternate code system.

Note: In practice, the identifier file is not used in the *SNOMED CT International Release* as the use of *Simple map type references sets* is preferred. The only current use of this file is for internal identification during the development process.

Integer (data type)

A datatype that represents a whole number.

Note: In *SNOMED CT release file* specifications integers are represented as a string of decimal digits. The range of values and support for negative values may be constrained for the specification are specified for each usage of this datatype. However, unless otherwise specified, all *release file* fields of data type *integer* are assumed to be 32-bit signed integers.

K

Keyword (field)

A field containing a potential search text in one of the *WordKey Tables* or a word excluded for key generation in the *Excluded Words Table*.

Note: Field name in SNOMED CT toolkit

L

linkedToId (field)

An Ordered *Reference Set* field containing a *SNOMED CT identifier* which refers to either a sub-group of components or a child *concept* in the alternative hierarchy represented by the *Reference set*. The parent of grouping component is represented by the *referencedComponentId*.

Note: Field name in SNOMED CT Release Format 2.

M

mapAdvice (field)

Field in a *Complex or Extended Map Reference Set* containing human-readable advice, that may be employed by the software vendor to give an end-user advice on selection of the appropriate *target code* from the alternatives presented to him within the group.

mapGroup (field)

Field in a *Complex or Extended Map Reference Set* containing an *integer* that groups a set of complex map records from which one may be selected as a *target code*. Where a *SNOMED CT concept* maps onto 'n' *target codes*, there will be 'n' groups, each containing one or more complex map records.

mapCategoryId (field)

Field in a *Complex or Extended Map Reference Set* that identifies the *SNOMED CT concept* in the metadata hierarchy which represents the MapCategory for the associated map member.

Note: The categories vary for different target code systems, each set of categories is represented by a subtype of 609331003|Map category value|. For example in the case of *ICD-10* the individual category values are *subtypes* of:

447634004|ICD-10 Map category value|.

mapPriority (field)

Field in a *Complex or Extended Map Reference Set* that specifies the *order* in which complex map records should be checked. Only the first map record meeting the run - time selection criteria will be taken as the *target code* within each *mapGroup*.

mapRule (field)

Field in a *Complex or Extended Map Reference Set* containing a machine-readable rule, (evaluating to either 'true' or 'false' at run-time) that indicates whether this map record should be selected within its *mapGroup*.

mapTarget (field)

Field in a *Simple Map Reference Set* or a *Complex or Extended Map Reference Set* that contains the *target code(s)* to which the *SNOMED CT concept* represented the *referencedComponentId* is mapped in the *target scheme*.

modifierId (field)

A field in the *relationship file* that indicates the *description logic* modifier that applies to that defining *Relationship* (e.g. "some" or "all").

Usage: Field name in SNOMED CT Release Format 2.

moduleId (field)

A field in each component *release file* which represents the development module within which it was created and is maintained.

Note: Field name in SNOMED CT Release Format 2, which is specified in [see *Identification of Source Module*].

O

order (field)

Order... to be defined.

Note: Field name in SNOMED CT Release Format 2

Q

query (field)

A field in a *Query specification reference set* that contains a text string representing criteria for selection of *SNOMED CT components* to be included in *Simple reference set*

Note: A standard syntax for use in these queries is currently under development and is due for publication in late 2014.

R

referencedComponentId (field)

A field in a *Reference Set* containing an *Identifier* which refers to the *component* to which a row in the *Reference Set* applies.

Note: This field is present in all types of *Reference Set* and, unless otherwise specified, the field data type is *SCTID*.

Reference Set file

The file structure used to distribute *SNOMED CT Reference sets*.

Alternatives

Refset file

refsetId (field)

A field in a *Reference Set* which uniquely *Identifier* which refers to the component to which a row in the *Reference Set* applies.

Note: This field is present in all types of *Reference Sets* and its data type is *SCTID*. It links together all the members of a *Reference Set* and refers to a concept that names the *Reference Set*.

Relationship file

The file structure used to distribute *SNOMED CT relationships*.

relationshipGroup (field)

Field in the *Relationship File* is used to group *Relationships* together for a *concept*. For example, where a particular type of prosthesis is inserted a joint, the *Defining characteristics* describing the prosthesis type would be in one group whereas those describing the location or laterality of the joint would be in another group.

S

SCTID (data type)

A unique integer identifier applied to each *SNOMED CT component* (*Concept*, *Description*, *Relationship*).

Note: The value of an SCTID is structured to include an item identifier, a check-digit and a partition identifier. Depending in the value of the partition identifier it may also include a namespace identifier.

sourceEffectiveTime (field)

A field in the Module Dependency *Reference Set* which specifies the *effectiveTime* of the version of the source module with depends on the specified version of the target module. The *effectiveTime* must match exactly.

Note: Field name in SNOMED CT Release Format 2

sourceId (field)

A field in the *Relationship Release File* containing a *SNOMED CT identifier* that refers to the *concept* that represents the source of the associated *Relationship*. The *sourceId* refers to the *concept* that is defined by the *Relationship*.

Note: Field name in *SNOMED CT Release Format 2*. In RF1 this field was called *ConceptId1*

Stated Relationship File

A distribution file containing the *stated form* of *SNOMED CT relationships*.

Notes:

1. The *stated form* of a *Concept* is the *Description Logic* definition that is directly edited by authors or editors. It consists of the stated | is a | *relationships* plus the defining *relationships* that exist prior to running a *classifier* on the logic definitions. Therefore, the *stated form* of a *Concept* is represented by a collection of *relationships*: one or more | Is a | *relationships* and zero or more defining *relationships*.
2. The *Stated Relationships File* is in the same table format as the *Relationships File*, but the value of the *characteristicTypeId* field is | Stated relationship (core metadata concept) |.

Related Links

Stated Relationships File on page 108

Stated definition view on page 56

Inferred definition views on page 56

String (data type)

A datatype representing a sequence of characters.

Note: In *SNOMED CT release file* specifications strings are represented using *Unicode UTF-8* encoding.

T

targetComponentId (field)

An Association *Reference Set* field containing a *SNOMED CT identifier* which specifies the target of the association from the source component (e.g. a *concept* or *Description*) referred to by the *referencedComponentId*.

Note: Field name in SNOMED CT Release Format 2.

targetEffectiveTime (field)

A field in the Module Dependency *Reference Set* which specifies the *effectiveTime* of the version of the target module on which the specified version of the source module depends. The *effectiveTime* must match exactly.

Note: Field name in SNOMED CT Release Format 2

term (field)

A text *string* that represents the *concept* referenced by the *conceptId* field in the *Description file*.

Note:

By default the *term* is a *UTF-8* string of up to 255 characters. However, description types can be specified which are longer in length and/or contain format markup (e.g. HTML).

Field name in the *Description file*.

Time (data type)

A datatype representing a date or time.

Note: In *SNOMED CT release file* specifications date and times are represented as strings using the ISO 8601 basic format.

- The date format used is YYYYMMDD.
- Where time is included the format is YYYYMMDDThhmmssZ. The time is separated from the date by the letter "T" and followed by the letter "Z" indicating that the timezone is UTC.

Examples:

July 31st 2012: **20120731**.

13:15 UTC on August 2nd 2012: **20120802T131500Z**

Transitive closure file

The file used to distribute the *transitive closure* of the *SNOMED CT subtype hierarchy*.

Note: This file is not currently distributed but can be generated from the Relationships file using a script.

typeId (field)

A field in the *Description* and *Relationship Release Files* which contains a *SNOMED CT identifier* that represents the type of *Description* or *Relationship* represented.

- *Description*. ***typeId*** represents the type of *Description*. *Description* types include *subtypes* of 900000000000446008 | Description type (core metadata concept) |. These include 900000000000013009 | Synonym (core metadata concept) | and 900000000000003001 | Fully specified name (core metadata concept) |. There is no *typeId* value for "*Preferred term*" as the *preferred term* is the *synonym* marked as "Preferred" in the appropriate [see *Language Reference Set*].
- *Relationship*. ***typeId*** represents the type of *Relationship* between the *concept* identified by *sourceId* and the *concept* identified by destinatonId. *Relationship types* are 116680003 | Is a (attribute) | and *subtypes* of 410662002 | Concept model attribute (attribute) |.

Note: Field name in the *Description file* and in the *Relationship file*.

U

Unicode

A standard character set, which represents most of the characters used in the world using a 16-bit encoding.

Note: The Unicode character set can be encoded using either UTF-16 or UTF-8. UTF-16 uses two bytes for every character. UTF-8 is able to store the most commonly used characters in western alphabets using a single byte, but it requires two bytes to encode accented characters and three bytes to encode symbols used in many non-European scripts.

UTF 16

A standard method of directly encoding *Unicode* using two bytes for every character.

Note: SNOMED CT release files do not use UTF-16. However, the UTF-8 representation used in release files can be converted to UTF-16.

UTF-8

A standard method of encoding *Unicode* characters in a way optimized for the ASCII character set. *UTF-8* is described in [see *Unicode UTF-8 encoding* on page 115].

Note: This encoding is used for release file fields of data type "String".

UUID (data type)

A datatype representing a sequence of unique *Identifier* encoded as a 128-bit integer.

Note: In *SNOMED CT release files UUIDs* are represented using as a string following the standard *canonical form*. In this string form a *UUID* is represented by 32 hexadecimal digits, displayed in five groups separated by hyphens, in the form 8-4-4-4-12 for a total of 36 characters (32 digits and four hyphens).

Example: ac527bed-9c70-4aad-8fc9-015828b148d9

Alternatives

Universally Unique Identifier
GUID
Globally Unique Identifier

V

valueId (field)

ValueId... to be defined.

Note: Field name in SNOMED CT Release Format 2

W

Word equivalents table

A data table in which each row represents a *Word Equivalent*. See [see *Word Equivalents* on page 125].

Note: File or Table name in SNOMED CT toolkit

WordBlockNumber (field)

A field in the *Word Equivalents Table*, which links together several rows which have an identical or similar meaning.

Note: Field name in SNOMED CT toolkit

WordKey table

A data table relating each word used in *SNOMED CT* (other than *Excluded Words*) to the *Descriptions*. See [see *Word Search Tables - Summary* on page 124].

Note: File or Table name in SNOMED CT toolkit

WordRole (field)

A field in the *Word Equivalents Table*, which specifies the usual usage of this word, abbreviation or phrase, or the usage in which it has a similar meaning to the text in one or more other rows of the table that share a common *WordBlockNumber*.

Note: Field name in SNOMED CT toolkit

WordText (field)

A field in the *Word Equivalents Table*, which contains a word, phrase, acronym or abbreviation that is considered to be similar in meaning to the text in one or more other rows of the table that share a common *WordBlockNumber*.

Note: Field name in SNOMED CT toolkit

WordType (field)

A field in the *Word Equivalents Table*, which specifies whether this row contains a word, phrase, acronym or abbreviation.

Note: Field name in SNOMED CT toolkit

Chapter 12

References

12.1 SNOMED CT Background

12.1.1 SNOMED CT: A Comprehensive terminology for Health Care

SNOMED Clinical Terms (*SNOMED CT*) was developed between 1999 and 2002 from a convergence of the content of SNOMED Reference Terminology® (*SNOMED RT*) and *NHS Clinical Terms Version 3* (*CTV3*). This convergence was a result of a strategic alliance to between the College of American Pathologists (CAP) and the *UK National Health Service*. In 2007, the *International Health Terminology Standards Development Organisation* acquired *SNOMED CT* and now develops and maintains it on behalf of its *Members* and *Affiliates*.

SNOMED CT combines the robust strength of *SNOMED RT* in the basic sciences, laboratory and specialty medicine with the highly granular clinician focused content of *CTV3* (formerly known as the *Read Codes*). The result is a comprehensive and precise clinical *reference terminology* that provides unsurpassed clinical content and expressivity for clinical documentation and reporting. *SNOMED CT* enables clinicians, researchers and patients to share comparable data worldwide, across medical specialties and sites of care.

SNOMED CT is founded on four basic principles that have guided development of its clinical content and technical design. These principles will continue to guide the evolution of *SNOMED CT* as it adapts and grows in the ever changing global health care environment.

These guiding principles are:

1. Development efforts must encompass broad, inclusive involvement of diverse clinical groups and medical informatics experts;
2. The clinical content must be quality focused and adhere to strict editorial policies;
3. The quality improvement process must be open to public scrutiny and vendor input, to ensure that the terminology is truly useful within healthcare applications;
4. There must be minimal barriers to adoption and use.

The design of *SNOMED CT* has been driven by the expressed needs of software developers for features that improve their ability to develop useful applications. In response to these needs, the design adds unique numeric *Identifiers*, includes links to legacy codes, supports a sustainable migration and maintenance strategy, permits adaptability for national purposes, and fosters alignment with other terminologies and standards such as *HL7*, *LOINC*, and DICOM.

The *IHTSDO* believes that *SNOMED CT* delivers a standardized quality clinical terminology that is required for effective collection of clinical data, its retrieval, aggregation and re-use as well as the sharing, linking and exchanging of medical information.

The file format used for distributing *SNOMED CT* content between 2002 and 2011, which is now known as *Release Format 1* (*RF1*), was balloted and approved as an *ANSI* standard. An enhanced file format, *Release Format 2* (*RF2*), has been approved and adopted by the *IHTSDO*. As *RF2* is phased in during 2011, it will simplify change management and add robust facilities for future extensibility.

12.1.1.1 SNOMED CT Quality Development Process

The *SNOMED Clinical Terms* development process incorporates the efforts of a team of internal and external *modelers*. A documented scientific process is followed which focuses on *Understandability, Reproducibility and Usefulness.* Content is defined and reviewed by multiple clinician editors. Conflicts between editors are resolved through an iterative process, based on achieving agreement and consensus, before being entered into the terminology. As necessary, additional experts are consulted to review the scientific integrity of the content.

The integration of *SNOMED RT* and *Clinical Terms Version 3* to create the first release, was a three year process that involved several stages of review and quality assurance:

- *Description* mapping: *NHS* editors evaluated each *SNOMED CT concept* and *term* and mapped it to the *Clinical Terms Version 3* terminology; *SNOMED CT* editors performed the same task mapping primarily disorders and procedures from *Clinical Terms Version 3* to *SNOMED RT*.
- *Description* mapping conflict resolution: Mapping discrepancies that occurred between *NHS* and *SNOMED CT* editors underwent a conflict resolution process to definitively place each *concept* within the merged *hierarchy*.
- Auto-classification: The merged database following *description* mapping conflict resolution underwent a series of quality control checks including auto-classification to identify and eliminate cycle errors (e.g. *concept* A | is a | B and *concept* B | Is a | A) and equivalency errors (e.g. where two defined *concepts* have the exact same definition).
- *Hierarchy* review: The reviewed database has undergone auto-classification and further review of inferred hierarchies.
- Ongoing *refinement*: The quality control process is continuously supplemented by feedback from users involved in adoption of *SNOMED Clinical Terms*.

12.1.1.1.1 Extent of Review

The quality processes used in the development of *SNOMED CT* were complemented with external review.

- Technical review: The technical specifications for *SNOMED CT* were published for comment on both the *SNOMED CT* and *NHS* websites.
- Alpha test review: Forty-two organizations in six countries tested the *SNOMED CT* alpha test file and completed a structured assessment instrument.
- Alpha test feedback: Debriefing sessions were conducted in the US, in the UK and in Australia, at which time test sites shared their positive experiences and recommendations for improvement.
- Peer review: The methods used in developing *SNOMED CT* were presented in 6 scientific papers at the 2001 American Medical Informatics Association (AMIA) meeting, the largest association of leaders in medical informatics in the world. *SNOMED CT* was also part of an additional three papers and six posters at the 2002 AMIA meeting and additional posters for AMIA 2003 and 2004.

SNOMED CT was also the subject of papers in the American Health Information Management Association (AHIMA) Journal in 2001-2003, posters at 2001 and 2002 annual meetings and presentations at the 2003 and 2004 annual meetings. In addition, AHIMA introduced an education program "Introduction to Clinical terminology" in 2004 which included a *SNOMED CT*.

Early adopters of *SNOMED RT* (a structure that mirrored *SNOMED CT core tables*) were debriefed on their implementation experience in order to identify the key issues to be addressed in the original version of the *SNOMED CT Technical Implementation Guide*.

12.1.1.1.2 Continuous Quality Improvement

Continuous improvement is an aim of the *IHTSDO*: Updating the breadth and scope of the content to reflect changes in clinical care and advances in medical science; refining the content to deliver greater precision for data collection, retrieval and aggregation; and enhancing the functionality to serve our users better.

12.1.2 Acknowledgments of Contributors to SNOMED CT®

SNOMED CT was originally created by the College of American Pathologists.

SNOMED CT has been created by combining *SNOMED RT* and a computer based nomenclature and classification known as *Clinical Terms Version 3*, formerly known as the *Read Codes Version 3*, which was created on behalf of the U.K. Department of Health and is Crown copyright.

The *IHTSDO* also acknowledges the contributions of:

- The American Academy of Ophthalmology, for the ophthalmology-related portions of this work.
- SNODENT®: the Systematized Nomenclature of Dentistry, copyright 1998, American Dental Association. Used with permission.
- SNOVET®: the Systematized Nomenclature of Veterinary Medicine, copyright 1982, 1993, American Veterinary Medical Association. Used with permission.
- *LOINC®*: the Logical Observation Identifier Names and Codes, copyright 1995-2008, Regenstrief Institute *LOINC* Committee. All rights reserved.
- NANDA®: North American Nursing Diagnosis Association Taxonomy II, copyright 2005- 2008, NANDA International. All rights reserved.
- The Perioperative Nursing Data Set® (PNDS), copyright 2002, AORN, Inc. All rights reserved.
- The Omaha System, copyright 1992, Martin and Associates. Used with permission.
- The Clinical Care Classification, copyright 2004, V.K. Saba. Used with permission.
- The Nursing Interventions Classification (NIC), copyright 2004, Mosby, Inc., and the Center for Nursing Classification and Clinical Effectiveness at the University of Iowa College of Nursing. Used with permission.
- The Nursing Outcomes Classification (NOC), copyright 2004, Mosby, Inc., and the Center for Nursing Classification and Clinical Effectiveness at the University of Iowa College of Nursing. Used with permission.
- This work contains material from the AJCC Cancer Staging Manual, Sixth Edition (2002) published by Springer-Verlag New York. Used with permission of the American Joint Committee on Cancer (AJCC), Chicago, Illinois.
- The Anesthesia Patient Safety Foundation's (APSF) Data Dictionary Task Force. Some material contributed. Copyright 2003, APSF, Inc. Used by permission of the APSF.
- This work contains *terms* from the British Association of Dermatology (BAD), and is used by permission of BAD. Crown Copyright 2003 British Association of Dermatologists.
- This work contains *terms* from The Royal College of Anaesthetists (RCoA), and is used by permission of RCoA. Crown Copyright 2003 The Royal College of Anaesthetists.
- This work contains *terms* from the Authorized Osteopathic Thesaurus, and is used by permission of the American Association of Colleges of Osteopathic Medicine and the American Osteopathic Association.

HL7 version 3 - An object-oriented methodology for collaborative standards development

Beeler GW Jr. *HL7* version 3 - an object- oriented methodology for collaborative standards development. Int J Med Inf. 1998 Feb; 48(1-3): 151-61.

Desiderata for controlled medical vocabularies in the twenty-first century

Cimino JJ in Methods Inf Med. 1998 Nov;37(4-5):394-403

This paper identifies some of the key requirements for a clinical terminology. The topics covered include:

- Vocabulary content;
- *Concept* orientation;
- *Concept* permanence;
- Non-semantic *concept identifiers*;
- *Polyhierarchy*;
- Formal definitions;
- Rejection of "not elsewhere classified" terms;
- Multiple granularities;
- Multiple consistent views;
- Context representation;
- Graceful evolution;
- Recognized redundancy.

HL7 Reference Information Model

[RIM] *HL7 Reference Information Model* (www.hl7.org/library/data-model/RIM/modelpage_non.htm)

Quality of clinical information retrieval using a semantic terminological model

Brown PJB, Sonksen P. Evaluation of the quality of information retrieval of clinical findings from a computerized patient database using a semantic terminological model. JAMIA 2000; 7:391-403.

Lexically Assign, Logically Refine strategy for integrating overlapping terminologies

Dolin RH, Huff SM, Rocha RA, Spackman KA, Campbell KE. Evaluation of a "Lexically Assign, Logically Refine" strategy for semi-automated integration of overlapping terminologies. JAMIA 1998; 5(2): 203-13.

Integration of tools for binding archetypes to SNOMED CT

Integration of tools for binding archetypes to *SNOMED CT*, Erik Sundvall, Rahil Qamar, Mikael Nyström, Mattias Forss , Håkan Petersson, Daniel Karlsson, Hans Åhlfeldt and Alan Rector; BMC Medical Informatics and Decision Making 2008, 8(Suppl 1):S7

Normal forms for description logic expressions of clinical concepts in SNOMED RT

Spackman, KA. *Normal forms* for *description logic expressions* of clinical *concepts* in *SNOMED RT*. Proc AMIA Symp. 2001: 627–631;[http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2243264/] webcite.

Representing clinical information using SNOMED CT with different information models

Markwell D, Sato L, Cheetham E: Representing clinical information using *SNOMED Clinical Terms* with different structural information models. [http://www.kr-med.org/] webcite; proceedings of KR-MED May 31/June 2 2008; Phoenix, Arizona, USA 2008.

Toward vocabulary domain specifications for health level 7-coded data elements

Bakken S, Campbell KE, Cimino JJ, Huff SM, Hammond WE. Toward vocabulary domain specifications for health level 7-coded data elements. JAMIA 2000; 7(4): 333-42.

Chapter 13

13 Glossary

This section contains selected terms from the IHTSDO Glossary. The full IHTSDO Glossary is available as follows:

- Online access: *www.ihtsdo.org/gl*;
- PDF file (US English): *www.ihtsdo.org/gl.pdf*;
- PDF file (GB English): *www.ihtsdo.org/gl_gb.pdf*.

A

Active component

A *SNOMED CT component* that is intended for use. *Release files* contain *Active* and *Inactive components* to provide a historical record of the content of the terminology at different points in time.

Note: A component is active when the most recent row with the relevant Component.*id* in the *Full Release* of the relevant *Release File* has the value Component.*active*=1 (one). The most recent row for a component is determined based on the Component.*effectiveTime* value.

Active concept

A *Concept* that is intended for use. *Release files* contain *Active* and *Inactive components* to provide a historical record of the content of the terminology at different points in time.

Note: A component is active when the most recent row with the relevant Component.*id* in the *Full Release* of the relevant *Release File* has the value Component.*active*=1 (one). The most recent row for a component is determined based on the Component.*effectiveTime* value.

Active description

A *Description* that is intended for use. *Release files* contain *Active* and *Inactive components* to provide a historical record of the content of the terminology at different points in time.

Note: A component is active when the most recent row with the relevant Component.*id* in the *Full Release* of the relevant *Release File* has the value Component.*active*=1 (one). The most recent row for a component is determined based on the Component.*effectiveTime* value.

Affiliate

An *IHTSDO Affiliate Licensee* in accordance with the *IHTSDO* Affiliate License Agreement.

Alternatives

IHTSDO Affiliate
Affiliate Licensee

Affiliate Licence Agreement

The agreement between an *IHTSDO affiliate* (the licensee) and the *IHTSDO* (the licensor) under which developers and implementers are permitted to use the *SNOMED CT International Release* and distribute it to their sub-licensees as part of a software system.

Alternatives

Affiliate Licence

ANSI

American National Standards Institute (*ANSI*) is a private non-profit organization that oversees the development of voluntary consensus standards for products, services, processes, systems, and personnel in the United States. The organization also coordinates U.S. standards with international standards.

Alternatives

ANSI
American National Standards Institute

Application Programming Interface

Application Programming Interface

A set of rules and specifications that enable communication between software programs. *Application Programming Interfaces* enables interaction between separate software programs, in much the same way that a *user interface* facilitates interaction between humans and computers.

Alternatives

API

Attribute

An *attribute* represents a characteristic of the meaning of a *concept* or the nature of a refinement.

Note: An *attribute* has a name which is represented by a *concept*. All the *concepts* that can be used to name *attributes* are *subtypes* of the *concept* | *concept model attribute* |. An *attribute* is assigned a value (*attribute value pair*) when used in the definition of a *concept* or in a *postcoordinated expression*. The permitted *attribute values* (*range*.) for an *attribute* depend on the *attribute name* and on the *domain* of the *concept* being refined.

Example: 116676008 | Associated morphology |

Alternatives

Concept Model Attribute
Relationship Type
Role

Attribute group

An association between a set of *attribute value* pairs which causes them to be treated separately from other *attribute value* pairs in the same definition or *postcoordinated expression refinement*.

Example:

The definition of the *concept* |cholecystectomy with exploration of common duct| has two |method| attributes with different values (|excision -action| and |exploration -action|) and two |procedure site direct| attributes with different values (|common bile duct structure| and |gallbladder structure|). The attributes are grouped so that procedure is not incorrectly classified as an |excision of common bile duct|.

Alternatives

AttributeGroup

Attribute name

A *concept* that represents the type of a *relationship* or the type of a *refinement* in a *postcoordinated expression*.

Notes:

1. The type of a *relationship* is indicated by the *typeId* attribute in the *Relationship file*
2. The *concepts* that can be used to name attributes are:
    - 116680003 | Is a (attribute) | and
    - *subtypes* of 410662002 | Concept model attribute |

Alternatives

Relationship Type
AttributeName

Attribute value

A *concept* that represents the target of a *relationship* or the value of an *expression refinement* in a *postcoordinated expression*.

Alternatives

Attribute-value
AttributeValue

Attribute value pair

A combination of an *attribute name* and an *attribute value* used to represent a specific type of information in a generic way without altering the underlying structure of an information model. The *attribute name* identifies the type of information and the *attribute value* provides a value.

Note: *Attribute value pairs* are used by *SNOMED CT* in *relationships* and *postcoordinated expressions*. In both cases, the *attribute name* and *attribute value* are expressed using *SNOMED CT concept identifiers*. In the *Relationship file*, the *attribute name* if represented by the *Relationship.typeId* and the *attribute value* by the *Relationship. destinationId*.

Authoritative concept

A *concept* with a specific meaning defined by an authoritative source such as a national or international professional body or standards organization.

Authorized Triage Organization

An organization approved by the *IHTSDO* to manage and triage change requests to for inclusion of content in the *SNOMED CT International Release* and/or one or more National *Extensions*.

Note: *IHTSDO Members* and their *National Release Centers* are likely to fulfill this role. In addition, *IHTSDO affiliates* and Standards Development Organizations may be eligible for consideration as *Authorized Triage Organizations*.

Alternatives

ATO

Automatic classification

A process that generated a logically consistent *subtype classification* by applying *description logic* rules to the stated definitions of a set of *concepts*.

Alternatives

Auto classify

B

Baseline

A release status applied to a collection of *SNOMED CT release files* that represent the first formally endorsed release of additions of *components* and/or *derivatives* to the *SNOMED CT International Release* or to a *SNOMED CT Extension*. The *Baseline* status indicates the releasing party (*IHTSDO* or the owner of the *Extension*) commits to maintain the release history of this release and all subsequent updates. Once confirmed as a *Baseline*, additional *components* and *derivatives* will be maintained and versioned in accordance with the Release Format 2 specification (i.e. by adding rows to the *Full Release* with the *effectiveTime* appropriate to the update).

Note: The significance the *Baseline* status is that implementers can use the additional *components* and *derivatives* in operational systems, with confidence in the subsequent maintenance of these additions.

Browser

A computer application or software tool used for exploring and searching terminology content. A typical *SNOMED CT browser* can locate *concepts* and *descriptions* by *Identifiers* and by searching the text of *description terms*. Various views of located *concepts* may be displayed including the set of related *descriptions*, the hierarchical *relationships* and other defining *relationships*.

Alternatives

SNOMED CT browser

C

Candidate Baseline

A provisional status applied to a collection of *SNOMED CT release files* that represent a proposed additions of *components* and/or *derivatives* to the *SNOMED CT International Release* or to a *SNOMED CT Extension*. The *Candidate Baseline* status indicates the releasing party (*IHTSDO* or the owner of the *Extension*) expects to subsequently confirm the release as the *Baseline*. However, if a significant issue is reported in its format or content, the releasing party reserves the right to withdraw a *Candidate Baseline* release, or to replace it with another *Candidate Baseline*, to address the issue. The releasing party need not commit to this being an actual *Baseline* release until shortly before the due date for the next release.

Note: The significance the *Candidate Baseline* status is that anyone implementing this data must be prepared for withdrawal or significant changes that may occur to the additional *components* or *derivatives*. Therefore, this data should not be used in an operational environment in ways that create a dependency on continued maintenance of the additional *components* or *derivatives*. However, a *Candidate Baseline* may be confirmed as the *Baseline* and, in that case all subsequent updates to the additional *components* and *derivatives* will be fully version tracked from the release of the *Candidate Baseline*.

Canonical form

An serialized representation of a *SNOMED CT expression* which follows the *normal form* and in which the *refinements*, *attributes* and *attribute groups* are arranged in a standard order.

Cardinality

? A measure of the number of elements in a set. Modeling rules include constraints on the *cardinality* of particular attributes or associations between classes.

CEN

The European Committee for Standardization is a major provider of European Standards and technical specifications. Its mission is to foster the European economy in global trading, the welfare of European citizens and the environment. Through its services it provides a platform for the development of European Standards and other technical specifications.

Alternatives

Comité Européen de Normalisation
European Committee for Standardization
Europäisches Komitee für Normung

CEN TC251

CEN /TC 251 (*CEN* Technical Committee 251) is a committee within the European Committee for Standardization (*CEN*) working on standardization in the field of Health Information and Communications Technology (ICT) in the *European Union*. Its goal is to achieve compatibility and interoperability between independent systems and to enable modularity in *Electronic Health Record* systems.

Check digit

The *check-digit* is the final (rightmost) digit of the *SNOMED CT Identifier* (*SCTID*). It can be used to check the validity of *SCTIDs*. *Clinical Information Systems* can use the *check-digit* to identify *SNOMED CT* codes that have been entered incorrectly (typo errors, etc). It is calculated using the Verhoeff algorithm.

Clinical Information System

A computer-based system that is designed for collecting, storing, manipulating and making available clinical information to support the delivery of healthcare services to individual people and populations.

Alternatives

CIS

Clinical Terms Version 3

One of the source terminologies, along with *SNOMED RT*, that were used to develop *SNOMED CT*. *CTV3* is UK Crown Copyright, distributed by the United Kingdom *National Health Service* (*NHS*), and is integrated into *SNOMED CT*.

Alternatives

CTV3
Version 3 of the Read Codes

C-NPU

Nomenclature, Properties and Units (*C-NPU* in collaboration with International *Union* of Pure and Applied Chemistry (IUPAC) The *IFCC-IUPAC* coding system Provides a terminology for Properties and Units in the Clinical Laboratory Sciences

Alternatives

Nomenclature, Properties and Units
NPU
IFCC IUPAC

Note: The name of the organization responsible for C-NPU sometimes used as a synonym

Collaborative Space

A web resource with software to help people involved in a common task achieve goals by enabling effective communication within an project or organization.

Note: The *IHTSDO Collaborative Space* supports the communication needs of *IHTSDO* governance and advisory bodies. *IHTSDO* Standing Committees, Affiliate Forum, Member Forum and Working Groups all have *Collaborative Space* Projects each of which contain meeting announcements, discussions, shared documents and issue trackers.

Alternatives

Collabnet

Common Terminology Services 2

An *Application Programming Interface* (*API*) specification that is intended to describe the basic functionality that needed by healthcare software implementations to query and access terminological content. *CTS2* defines the functional requirements of a set of service interfaces to allow the representation, access, and maintenance of terminology content either locally, or across a federation of *terminology service* nodes.

Note: *CTS2* is specified as an *API* rather than a set of data structures to enable a wide variety of terminological content to be integrated within a common framework without the need for significant migration or rewrite.

Note: *CTS2* was developed from the original the [see *HL7 CTS specification*] and is now a joint initiative between HL7 and the [see *Object Management Group (OMG)*].

Alternatives

CTS2

HL7 CTS2

Complement

In set theory the *complement* of set A relative to the universal set U is the set of all members of U that are not members of A.

Note: Set theory is applied when describing the intended result of combinations of Reference Sets or Constraints.

SNOMED CT Component

Refers to any item identified by an *SCTID* in the main body of *SNOMED CT*, or in an authorized *Extension*. The *partition-identifier* indicates the type of component referred to by that *SCTID*. Each *component* is a uniquely identifiable instance of one of the following:

- *Concept*
- *Description*
- *Relationship*

Alternatives

Component

Component history

A record of an addition or change in the *status* of a *SNOMED CT Component* in a particular *Release Version*.

Compositional grammar

The set of rules that govern the way in which *SNOMED CT expressions* are represented as a plain text string.

Note: The specification of the [see *SNOMED CT Compositional Grammar*] is available as part of the Technical Implementation Guide.

Alternatives

SNOMED CT compositional grammar

Concept

A clinical idea to which a unique *Concept Identifier* has been assigned.

The *term concept* may also be used informally with the following meanings:

- The *concept Identifier*, which is the key of the *Concept file* (in this case it is less ambiguous to use the *term* "conceptId" or "*concept* code");
- The real-world referent(s) of the *Concept Identifier*, that is, the class of entities in reality that the *Concept Identifier* represents (in this case it is less ambiguous to use the *term* "meaning" or "code meaning").

Alternatives

SNOMED CT concept

Concept enumeration

Use of *SNOMED CT concept Identifiers* to represent of a set of values for a property of a particular type of *SNOMED CT component*.

Note: The *SNOMED CT concepts* used to represent *concept enumerations* are usually *subtype children* (or *descendants*) of a relevant general *concept* in the *SNOMED CT* metadata hierarchy. Each possible value is represented by a single child *concept*, and the set of values can be used to enable selection from a pick-list of one or more *concepts*.

Example:

- 900000000000446008 | Description type (core metadata concept) |
  - 900000000000003001 | Fully specified name (core metadata concept) |
  - 900000000000013009 | Synonym (core metadata concept) |
  - 900000000000550004 | Definition (core metadata concept) |

Figure 138: Concept enumeration for: Description.typeId

Concept equivalence

Equivalence is the state of two *SNOMED CT concept* codes or *postcoordinated expressions* having the same meaning. *Concept equivalence* can occur when a *postcoordinated expression* has the same meaning as a *precoordinated concept* code; or when two different *postcoordinated expressions* have the same meaning.

Concept Identifier

A *SNOMED CT Identifier* that uniquely identifies a *Concept* (meaning).

Example: For the meaning named | Pneumonia (disorder) |, the *Concept Identifier* is 233604007.

Concept model

A set of rules that determines the permitted sets of *Relationships* between particular types of *concept*. The *Concept Model* specifies the attributes that can be applied to particular *concepts* and the ranges of permitted values for each of these attributes. There are also additional rules on the *cardinality* and grouping of particular types of *Relationships*.

Note: The [see *Concept Model Guide (6)*] (which is part of the Technical Implementation Guide) summarizes the current set of rules applied to modeling *SNOMED CT concepts*. More detailed information, aimed at those involved creating and modeling content, is available in the *SNOMED CT* Editorial Guide.

Constraint

A rule that specifies limits on the attributes, values and associations that may be applied to a particular component.

Examples:

1. A modeling constraint may limit the permissible defining *Relationships* applied to a particular type of *concept*.
2. An instance data constraint may limit the permissible refinements that may be applied to particular *concept*

Context domain

A context domain is a set of values that are, or may be, used in an identifiable logical setting in an application, protocol, *query* or communication specification. A context domain may be very broad (e.g. procedures or diagnoses) or very narrow (e.g. procedures performed by a specialty or possible values for a field in specific message).

Context specific characteristic

A *Relationship* to a target *Concept* that provides information about the source *Concept* that is true at a particular time or within a particular country or organization . Contrast with *Defining characteristic* and *Qualifying characteristic*. Referred to in *CTV3* as a 'Fact'.

Context wrapper

The part of a *SNOMED CT expression* that specifies the context that applies to the *focus concept* that it contains.

Example: "Family history of asthma" can be represented by an *expression* in which the *concept* "asthma" is nested within an *context wrapper* that indicates that this is "family history" - rather than a current condition affecting the patient. For further details see [see *Modeling semantic context*].

Core file

A distribution file used to represent the main *SNOMED CT components* (*concepts*, *descriptions* and *relationships*).

Note: In the past the term "core" has also been used to refer to the content of the *SNOMED CT International Release* but this usage is deprecated.

Alternatives

SNOMED CT core
Core table
SNOMED CT core table
SNOMED CT core file
Core table

D

Darwin Information Typing Architecture

The Darwin Information Typing Architecture (*DITA*) is an XML-based architecture for authoring, producing, and delivering information. Although its main applications have so far been in technical publications, *DITA* is also used for other types of documents such as policies and procedures.

Note: *DITA* is used for creation, publication and maintenance of many *IHTSDO* guidance documents.

Alternatives

DITA

Data Analysis System

A computer system that is used to analyze records or other data that is encoded using *SNOMED CT*, but not if that system is also a *Data Creation System*;

Note: *IHTSDO* charges fees for use of *Data Analysis Systems* and *Data Creation Systems* in Non-Member Territories.

Data Creation System

A computer system that is used to create records or other data that is encoded using *SNOMED CT*.

Note: *IHTSDO* charges fees for use of *Data Analysis Systems* and *Data Creation Systems* in Non-Member Territories.

Data migration

Steps taken to enable legacy data to be accessible as part of a system that uses *SNOMED CT*.

Note: The objective of *data migration* is to enable data recorded prior to introduction of *SNOMED CT* can be retrieved and reused within a *SNOMED CT enabled application*. Options for *data migration* include actual conversion of the data or provision of methods for accessing the data in its original form.

Defining relationship

A *relationship* to a target *concept* that is always necessarily true from any instance of the source *concept*.

Example: The *defining relationships* of the concept | gastrectomy | include |method|=|excision - action| and |procedure site - Direct|=|stomach structure|.

Alternatives

Defining characteristic

Delta release

A *Release Type* in which the *release files* contain only component versions created since the previous release. Each component version in a *delta release* represents either a new component or a change to an existing component.

SNOMED CT Derivative

A document, subset, set of maps, or other resource that consists of, includes, references or is derived from one or more *SNOMED CT components*. The standard computer processable representation for most types of *SNOMED CT derivatives* is a *Reference set*.

Alternatives

Derivative

Description

An association between a human-readable phrase (*term*) and a particular *SNOMED CT concept* code. Each *description* is represented by a separate row in the *Description file*.

Note: Each *description* has a unique *identifier* and connects *concept* with a *term* of a specified *description type*.

Alternatives

SNOMED CT description

Description Identifier

A *SNOMED CT Identifier* that uniquely identifies a *Description*.

Description logic

A representation of semantic knowledge that allows formal reasoning to be applied based on axioms that state *relationships* between *concepts*.

Note: *Description logic* definitions of *SNOMED CT concepts* are represented by *defining relationships*. The formal rules of *description logic* can be applied to *defining relationships* by software tools (*description logic classifiers*) to interpret the meaning of *concepts*. This enables confirmation of the logical integrity of the terminology, and can also be used to support meaning-based retrieval from *SNOMED CT enabled* record systems.

Alternatives

DL

Related Links

Wikipedia entry on Description logic

Description logic classifier

A software tool that applies the rules of a *description logic* to a set of data to make inferences about the *relationships* between sets of *concepts*.

Note: *SNOMED CT concepts* and *relationships* are processed by a *description logic classifier* to generate the *subtype hierarchy*. *SNOMED CT expressions* can also be processed by a classifier to make inferences that support selective retrieval.

Alternatives

Classifier

Description Type

An indication of the intended usage of the *term* of a *SNOMED CT description* when applied to the associated *concept*.

Notes:

1. The *description type* is represented by the value of the *description.typeId* attribute.
2. Permitted values include the following (other types may be defined in future):

Table 311: Description types

typeId (with term)	Further information
900000000000003001 | Fully specified name |	A *term* unique among *active descriptions* in *SNOMED CT* that names the meaning of a *concept* code in a manner that is intended to be unambiguous and stable across multiple contexts (see *fully specified name (FSN)*).
900000000000013009 | Synonym |	A *term* that is an acceptable way to express a the meaning of a *SNOMED CT concept* (see *synonym*).
900000000000550004 | Definition |	An additional textual *description* applied to some *SNOMED CT concepts* that provides additional information about the intended meaning or usage of the *concept* (see *textual definition*).

3. The The *preferred term* is the *synonym* marked as preferred for use in the *Language reference set*for a given *language* or *dialect* (it is not a distinct *description type*).

Dialect

A *language* modified by the vocabulary and grammatical conventions applied to the *language* of a particular geographical or cultural environment.

Directed Acyclic Graph

A set of nodes connected to one another by lines (edges) in which each connection has a specified direction such that no route that follows the direction of the connections enters a loop (cycle).

Example: The *SNOMED CT subtype hierarchy* is an example of a *Directed Acyclic Graph*. *SNOMED CT concepts* are nodes and *"is a" Relationships* are the directed lines that connect them. All *"is a" Relationships* lead from a more specific *concept* to a more general *concept*, so a cycle would be a logical error (e.g. if "rubella virus" is a type of "virus" and "virus" is a type of "microorganism", then "microorganism" cannot be a type of "rubella virus").

Alternatives

DAG

Domain

A set of *concepts* which the *Concept Model* permits to be defined or refined using a particular set of *attributes* and *ranges*.

Note: A *domain* to which an *attribute* can be applied is typically defined to include concepts in one or more branches of the subtype hierarchy.

Example: The *domain* of the *attribute* 116676008 | Associated morphology | is defined as subtype of 404684003 | Clinical finding | hierarchy. Similarly, the *range* for values of 116676008 | Associated morphology | is subtypes of 49755003 | Morphologically abnormal structure |.

Alternatives

Concept model domain

Draft Standard for Trial Use

A *Draft Standard for Trial Use* is a specification and process to allow implementers to test a standard. At the end of the trial period the standard may be balloted, revised or withdrawn.

Example: The joint project between HL7 International and the *IHTSDO*, *TermInfo*, is an example of an HL7 DSTU.

Alternatives

DSTU

Duplicate term

A *Term* that occurs in several *Active Descriptions*. *Duplicate Terms* are valid in *SNOMED CT* since the intention is to provide natural *terms* used by clinicians rather than to apply formalized phraseology. The formalized form is provided by the *Fully Specified Name* and these are not permitted to be duplicated.

Dynamic snapshot view

A "*snapshot view*" for a specified date that is generated by filtering a "*full view*".

E

Electronic health record

A systematic collection of health information about individual patients or populations that is stored in a digital form. An *Electronic health record* may contain a complete and detailed record of a patient's health or may consist of a summary of information of particular relevance to continuing delivery of care.

Alternatives

EHR

EN13606

Electronic Health Record Communication (EN 13606) European Standard developed by *CEN TC251* to define a rigorous and stable information architecture for communicating part or all of the *Electronic Health Record* (*EHR*) of a single subject of care (patient). This is to support the interoperability of systems and components that need to communicate (access, transfer, add or modify) *EHR* data via electronic messages or as distributed objects:

- preserving the original clinical meaning intended by the author;
- reflecting the confidentiality of that data as intended by the author and patient. .

Enabled application

A software application designed to support the use of *SNOMED CT*.

Alternatives

SNOMED CT enabled application
SNOMED enabled application
SNOMED CT application
SNOMED application

Enabled implementation

Implementation of information systems that are able to make effective use of *SNOMED CT* in an organization or region.

Note: *SNOMED CT enabled implementation* has a broader meaning than *SNOMED CT enabled application*. An implementation involves practical deployment of one or more applications but extends beyond the software itself to address personnel and organizational issues that allow the potential benefits to be realized.

Alternatives

SNOMED CT enabled implementation
SNOMED enabled implementation
SNOMED CT implementation
SNOMED implementation

Equivalence

See *Word Equivalents*, *Phrase equivalence* and *Concept equivalence*.

Expression

A structured combination of one or more *concept identifiers* used to express a clinical idea.

Note:

An *expression* containing a single *concept identifier* is referred to as a *precoordinated expression*. An *expression* that contains two or more *concept identifiers* is a *postcoordinated expression*.

The *concept identifiers* in a *postcoordinated expression* are related to one another in accordance with rules expressed in the *SNOMED CT Concept Model*.

These rules allow an *expression* to *refine* the meaning of a *concept* by applying more specific values to particular attributes of a more general *concept*.

Example:

284196006 | burn of skin | : 363698007 | finding site | = 33712006 | skin of hand |

Alternatives

SNOMED CT expression

Expression refinement

The part of a *SNOMED CT expression* that applies qualifying details to a *focus concept*.

Example: A "spiral fracture of the left humerus" can be represented by an *expression* in which the *concept* "fracture of humerus" if made more specific by the addition of two refinements "laterality: left" and "associated morphology: spiral fracture".

Alternatives

Refinement

Extension namespace identifier

See *namespace identifier*.

F

Focus concept

The part of a *SNOMED CT expression* that represents a clinical finding, observation, event or procedure. This *focus concept* may be given context by a surrounding content wrapped and may be made more specific by a refinement.

Example: A past history of replacement of the left hip may be represented by a *SNOMED CT expression* in which the *focus concept* "hip replacement" is refined by "laterality: left" and enclosed in a *context wrapper* representing "past history".

Full release

A *Release Type* in which the *release files* contain every version of every component ever released.

Full view

A view of *SNOMED CT* that includes all the components in a *Full release*. This includes the full history or all components ever released. A *Full view* can be filtered to provide a *Dynamic snapshot view* of the components as they were at any point in the past.

Fully Specified Name

A *term* unique among *active descriptions* in *SNOMED CT* that names the meaning of a *concept* code in a manner that is intended to be unambiguous and stable across multiple contexts.

Notes:

1. *Fully specified names* are indicated with the typeId 900000000000003001 | Fully specified name |.
2. There may be more than one *active description* with the typeId 900000000000003001 | Fully specified name |. However, only one *fully specified name*should be marked as preferred for use in a given *language* or *dialect* in the relevant *Language reference set*.
3. The US English *fully specified name* is the point of reference for the meaning of all *concepts* in the *SNOMED CT International Edition*. However, where a *concept* is part of an *extension* the *fully specified name* specified in the original language of that *extension* applies.

Alternatives

FSN

H

Health Level 7

A not-for-profit, *ANSI*-accredited standards developing organization dedicated to providing a comprehensive framework and related standards for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice and the management, delivery and evaluation of health services.

Alternatives

HL7

Health Level 7 Version 3

A standard for communication of health care information developed by *HL7*. Version 3 is based on a formal development framework and its communication structures a derived as refinements from a *Reference Information Model* (*HL7 V3 RIM*).

Alternatives

HL7 V3

Health Level 7 Version 3 Reference Information Model

The *reference information model* on which *HL7 Version 3* is based.

Alternatives

HL7 V3 RIM

Hierarchy

An ordered organization of *concept* codes linked together through | is a | *relationships*. *Concept* codes linked to their more general parent *concept* codes directly above them in a *hierarchy Concept* codes with more general meanings are usually presented as being at the top of the *hierarchy* and then at each level down the *hierarchy* code meanings become increasingly more specific or specialized . Formally, a *hierarchy* is represented as a *Directed Acyclic Graph*.

HL7 TermInfo

An *HL7* project that developed the ' *HL7* Version 3 Implementation Guide: Using *SNOMED CT*' as a *Draft Standard for Trial Use* (DSTU). The purpose of this guide is to ensure that *HL7* Version 3 standards achieve their stated goal of semantic interoperability when used to communicate clinical information that is represented using *concepts* from *SNOMED CT*

Alternatives

Term Info

I

ICD-10

The International Statistical Classification of Diseases and Related Health Problems 10th Revision (ICD-10) is a coding of diseases and signs, symptoms, abnormal findings, complaints, social circumstances and external causes of injury or diseases, as classified by the *World Health Organization*. (*WHO*).

ICD-9

The International Statistical Classification of Diseases and Related Health Problems 9th Revision (ICD-9) is a coding of diseases and signs, symptoms, abnormal findings, complaints, social circumstances and external causes of injury or diseases, as classified by the *World Health Organization*. (*WHO*).

Note: Replaced by *ICD-10*.

ICD-9-CM

The International Classification of Diseases, 9th Revision, Clinical Modification" (ICD-9-CM), Sixth Edition, issued for use beginning October 1, 2008 for federal fiscal year 2009 (FY09). The ICD-9-CM is maintained jointly by the National Center for Health Statistics (NCHS) and the Centers for Medicare & Medicaid Services (CMS).

IFCC IUPAC

Nomenclature, Properties and Units (*C-NPU*) in collaboration with International *Union* of Pure and Applied Chemistry (IUPAC) The *IFCC-IUPAC* coding system Provides a terminology for Properties and Units in the Clinical Laboratory Sciences

Inactive component

A *SNOMED CT component* that is not intended for use. *Active* and *Inactive components* are included in *release files* to provide a historical record of the content of the terminology different points in time.

Note: A component is inactive when the most recent row with the relevant Component.*id* in the *Full Release* of the relevant *Release File* has the value Component.*active*=0 (zero). The most recent row for a component is determined based on the Component. *effectiveTime* value.

Alternatives

Inactive

Inactive concept

A *Concept* that is not intended for use. *Release files* contain *Active* and *Inactive components* to provide a historical record of the content of the terminology at different points in time.

Note: A component is inactive when the most recent row with the relevant Component.*id* in the *Full Release* of the relevant *Release File* has the value Component.*active*=0 (one). The most recent row for a component is determined based on the Component.*effectiveTime* value.

Inactive description

A *Description* that is not intended for use. *Release files* contain *Active* and *Inactive components* to provide a historical record of the content of the terminology at different points in time.

Note: A component is inactive when the most recent row with the relevant Component.*id* in the *Full Release* of the relevant *Release File* has the value Component.*active*=0 (one). The most recent row for a component is determined based on the Component.*effectiveTime* value.

Intellectual property rights

As defined in the IHTSDO affiliate License Agreement: patents, trade marks, service marks, copyright (including rights in computer software), moral rights, database rights, rights in designs, trade secrets, know-how and other *intellectual property rights*, in each case whether registered or unregistered and including applications for registration, and all rights or forms of protection having equivalent or similar effect in any jurisdiction.

Note: The *IHTSDO* owns the *intellectual property rights* of *SNOMED CT*. The *IHTSDO* is responsible for ongoing maintenance, development, quality assurance, and distribution of *SNOMED CT*.

Alternatives

IPR
Intellectual Property
IP

International Health Terminology Standards Development Organisation

The *International Health Terminology Standards Development Organisation* (*IHTSDO*) is a not-for-profit association that develops and promotes use of *SNOMED CT* to support safe and effective health information exchange.

Alternatives

IHTSDO

Intersection

In set theory the *intersection* of the sets A and B, is the set of all objects that are members of both A and B.

Note: Set theory is applied when describing the intended result of combinations of Reference Sets or Constraints.

IS A

The RelationshipType that defines a supertype - *subtype*. *Relationship* between two *Concepts*. Usually expressed as *subtype* | is a | supertype. For Example, Blister with infection | is a | Infection of skin.

ISO

ISO (International Organization for Standardization) is the world's largest developer and publisher of International Standards. ISO is a network of the national standards institutes from over 160 countries, one member per country, with a Central Secretariat in Geneva, Switzerland, that coordinates the system.

ISO TC215

ISO TC215 is the ISO Technical Committee for Standardization in the field of information for health, and Health Information and Communications Technology (ICT). Its objectives are to enable compatibility and interoperability between independent systems, to ensure compatibility of data for comparative statistical purposes (e.g. classifications), and to reduce duplication of effort and redundancies.

K

Kind of value

The nature of a value that may be associated with a *Concept*. For example, the *concept* I systolic blood pressure I can label a numeric value. The Kind-of-Value that it labels is a pressure.

L

Language

For purposes of *SNOMED CT* translations, a *language* is a vocabulary and grammatical form that has been allocated an ISO639-1 *language* code. See also *dialect*.

LOINC

Logical Observation Identifiers Names and Codes, a dataset of universal identifiers for identifying medical laboratory observations and other clinical observations to facilitate exchange and storage of clinical results or vital signs.

Alternatives

Logical Observation Identifiers Names and Codes

M

Machine readable concept model

A representation of the rules that comprise the *SNOMED CT Concept Model* in a form that can be processed by computer software and applied to validate content.

Note: The *Machine readable concept model* can be applied to support consistent authoring of *SNOMED CT* content and can also support the creation of valid *postcoordinated expressions* in instance data.

Alternatives

MRCM

Managed content addition

An implementation strategy that involves creating additional *concepts*, *Descriptions* and *Relationships* in an extension so that data can be recorded to the required level of detail using only *precoordinated expressions*.

Note: A *description logic classifier* can be used to obtain an updated inferred view of the whole terminology in order to support data retrieval.

Alternatives

MCA

Mapping

The process of converting data from a representation in one code system, classification or terminology so that it is represented in another code system, classification or terminology.

Note:

The process as a whole includes the preparation and maintenance of resources used to enable this conversion and the application of such resources to convert instance data.

In *SNOMED CT Mapping* resources are distributed as [see *Simple*] and [see *Complex and Extended Map Reference Sets*]

.

Alternatives

Cross Mapping

Member

A Member of the *International Health Terminology Standards Development Organisation* (*IHTSDO*) in accordance with the *IHTSDO* Articles of Association.

Alternatives

IHTSDO member

Member territory

A territory that is represented by an *IHTSDO Member* (as published by the Licensor from time to time)

Metadata

SNOMED CT content (including *concepts*, *Descriptions* and *Relationships*) that is used to describe or provide additional information about *SNOMED* content and derivatives (including *reference sets*).

Note:

All *SNOMED CT* metadata *concepts* are *subtypes* of 900000000000441003 | SNOMED CT Model Component (metadata) |. The top level of the metadata hierarchy represent broad groups of metadata as shown below.

- 900000000000441003 | SNOMED CT Model Component (metadata) |
  - 106237007 | Linkage concept (linkage concept) | ...
  - 370136006 | Namespace concept (namespace concept) | ...
  - 900000000000442005 | Core metadata concept (core metadata concept) | ...
  - 900000000000454005 | Foundation metadata concept (foundation metadata concept) | ...

Figure 139: Top level of the SNOMED CT metadata hierarchy

Alternatives

SNOMED CT Metadata

Migration

See *Operational migration*, *Data migration* and *Predicate migration*.

Model of meaning

An information model that is structured in a way that is designed to provide a common representation of particular types of information which is reusable between different use cases. A model of a meaning combines structural and terminological component in ways that avoid ambiguity and minimize alternative representations of similar meanings.

Example: A model that specifies a how *SNOMED CT expressions* are used to represent in a particular *reference information model* to represent clinical findings and procedures in an *electronic health record*.

Note: In contrast, a *model of use* represents the underlying meaning in a way that is determined by a limited set use cases.

Model of use

An information model that is structured in a way suggested by a particular intended use of the information that will be represented by that model.

Example: A database that is structured with tables and fields that match specific *user interface* forms and the data entry box on those forms.

Note: In contrast, a *model of meaning* represents the underlying meaning in a way that is common to and reusable between different use cases.

Modeler

A person who directly edits the logic definitions and other structures of the terminology. Also sometimes called Clinical Editor or Terminology Manager.

Alternatives

SNOMED CT modeler
Modeller
SNOMED CT author

Modeling

The process of editing logic definitions to reflect the meaning intended by the *Fully Specified Name*.

Alternatives

SNOMED CT modeling
Modelling
SNOMED CT authoring

Monohierarchy

A *Monohierarchy* is a hierarchy in which each node is linked to one and only one parent node.

This type of hierarchy can be represented as a tree with a single root to which each node is attached.

Alternatives

Monohierarchical classification

Moved elsewhere

A *Status* value applicable to a *component* that has been moved to another *Namespace*. *Concepts* or *Descriptions* may be moved from an *Extension* to the *International Release*, from the *International Release* to an *Extension* or between one *Extension* and another. Moves occur if responsibility for supporting the *Concepts* changes to another organization .

Note: Component status value

N

Namespace concept

A *Concept* that exists to represent a *SNOMED CT Namespace-Identifier*. All *Namespace Concepts* are direct *subtypes* of the *Concept* "*Namespace Concept* which is a *subtype* of the Top-Level *Concept* "Special *Concept* ".

Namespace identifier

A seven digit number allocated by the *IHTSDO* to an organization that is permitted to maintain a *SNOMED CT Extension*. The *namespace identifier* forms part of the *SCTID* allocated every *component* that originated as part of an *Extension*. Therefore, it prevents collision between *SCTIDs* issued by different organizations . The *namespace-identifier* indicates the provenance of each *SNOMED CT component*.

Note: Short format *SCTIDs*,which are used for *components* that originate in the *International Release*, do not include a *namespace-identifier*. In this case the *partition identifier* provides sufficient information about the origin of the component.

Alternatives

Extension namespace identifiers
NamespaceId

National Health Service

Located in the United Kingdom, the *National Health Service* (*NHS* worked with the College of American Pathologists in the development of *SNOMED CT*. The *NHS* is was one of the founder Members of the *IHTSDO* that is now responsible for *SNOMED CT*.

Alternatives

UK National Health Service
UK NHS
NHS

National Library of Medicine

The *National Library of Medicine* (*NLM*, in Bethesda, Maryland, is a part of the National Institutes of Health, US Department of Health and Human Services (HHS). *NLM* is the world's largest medical library. The *NLM* represents the US, as a founder Member of the *IHTSDO*.

Alternatives

NLM

National Release Center

The organization within an *IHTSDO Member* country that is responsible for maintaining and releasing *SNOMED CT* content including any National *Extensions* of *SNOMED CT*.

Natural language processing

Natural *Language* processing (*NLP* is concerned with the interactions between computers and human-readable *languages*. *NLP* includes understanding and generation of human-readable representations. *NLP* understanding systems convert human-readable text into formal representations, which may for example include *SNOMED CT expressions*, to enable more effective processing by other software. *NLP* generation systems convert information from formal representations into human-readable text.

Alternatives

NLP

Navigation

The process of locating a *Concept* by traversing *Relationships* or *Navigation links*. For example, moving from a supertype *Concept* to more refined *Concepts*, from a specific *Concept* to a more general *Concept* or from a *Concept* to its *Defining characteristics*. *Navigation Links* allow *navigation* to follow intuitive routes through *SNOMED CT* even where there are no direct supertype or *subtype Relationships*.

Navigation concept

A *Concept* that exists only to support *Navigation*. A *Navigation Concept* is not suitable for recording or aggregating information. All *Navigation Concepts*:

- Are direct *subtypes* of the *concept* "Navigational *Concept*";
- Have not other supertype or *subtype Relationships*
- Are linked to other *Concepts* only by Navigational Links.

Navigation Hierarchy

A hierarchical view of a set of *SNOMED CT concepts* that is intended to assist navigation at the *user interface*.

Note: There are several differences between *navigation hierarchies* and the formal *subtype hierarchy*:

1. Links between *concepts* in a *navigation hierarchy* are represented by an [see *Ordered Reference Set*]
2. *Navigation links* do not contribute to the semantic definitions of *concepts*. Therefore, the criteria for creating a *navigation hierarchy* can be based on arbitrary criteria relating to usability;
3. A *navigation hierarchy* may specify the order in which a set of *concepts* are to be displayed when nested under another specified *concept*.

Non-member territory

A territory that is not an *IHTSDO Member* Territory

Note: In accordance with *IHTSDO affiliate* License, fees are payable to the *IHTSDO* for use of *SNOMED CT* in non-Member Territories.

Normal form

A representation of a *SNOMED CT expression* in which none of the referenced *concepts* are *fully defined* and where there is no redundancy or duplication of meaning.

Notes:

1. *Normal forms* can be used to determine *equivalence* and subsumption between *expressions* and thus assist with selective retrieval.
2. Any *SNOMED CT expression* can be transformed to its *normal form* by replacing each reference to a *fully defined concept* with a nested *expression* representing the definition of that *concept*. Transformation rules then resolve redundancies, which may arise from expanding *fully defined concepts*, by removing less specific *attribute values*.

Normal form transformation

The process of converting a *SNOMED CT expression* into its *normal form*.

Notes:

1. The *normal form* provides a way compare different *expressions* which have a similar meaning.
2. The transformation rules are described in [see *Transforming expressions to normal forms*].

Alternatives

Transform

Transformation

O

openEHR

openEHR is an international not-for-profit Foundation working toward making the interoperable, life-long *electronic health record* a reality and improving health care in the information society. It is develops specifications that are primarily based on and extend key aspects of the *CEN* Standard for *Electronic Health Record* Communication (EN 13606).

Operational migration

Steps taken to enable an organization that either used a previous coding scheme (or no clinical coding scheme) to make use of *SNOMED CT*.

P

Partition-identifier

The second and third digits from the right of the string rendering of the *SCTID*. The value of the *partition-identifier* indicates the type of component that the *SCTID* identifies (e.g. *Concept*, *Description*, *Relationship*, etc) and also indicates whether the *SCTID* contains a *namespace identifier*.

Alternatives

PartitionId

Pending move

A *Status* value applicable to a *component* that is thought to belong in a different *Namespace* but which is maintained with its current *SCTID* while awaiting addition to the new *Namespace*. A new *Concept* and associated *Descriptions* may be added with this *Status* where a missing *SNOMED CT Concept* is urgently required to support the needs of a particular *Extension*. Existing *Concepts* are also given this *status* when it is recognized that they should be moved to a different *Extension* or to the *International Release*. See also *Moved elsewhere*.

Note: Component status value.

Phrase equivalence

Two words or phrases with a similar meaning. For example, "renal calculus" and "kidney stone". See *Word Equivalents*.

Polyhierarchy

A *Polyhierarchy* is a hierarchy in which each node has one or more parents.

This type of hierarchy can be represented as a graph in which each node has a one or more directed links to or from other nodes. Since a node in a hierarchy cannot be a *descendant* of itself the resulting graph must not contain cyclic *Relationships*. This type of graphs is referred to as a "*Directed Acyclic Graph*".

Alternatives

Polyhierarchical classification

Postcoordinated expression

Representation of a clinical meaning using a combination of two or more *concept identifiers* is referred to as *postcoordination*.

Note: Some clinical meanings may be represented in several different ways. *SNOMED CT* technical specifications include guidance for transforming logical *expressions* to a common *canonical form*.

Example: *SNOMED CT* includes the following *concepts*:

125605004 | fracture of bone |
363698007 | finding site |
71341001 | bone structure of femur |

SNOMED CT also includes a *precoordinated concept* for 71620000 | fracture of femur |. Therefore It is possible to represent the clinical meaning "fracture of femur" in different ways:

- as a *precoordinated expression*:
  - 71620000 | fracture of femur |
- or as a *postcoordinated expression*:
  - 125605004 | fracture of bone | : 363698007 | finding site | = 71341001 | bone structure of femur |

Alternatives

Postcoordinated
Postcoordination

Precoordinated expression

Representation of a clinical meaning using a single *concept identifier* is referred to as a *precoordinated expression*.

Note: In contrast, *expressions* that contain two or more *concepts Identifier* are referred to as *postcoordinated expressions*. For more information and examples see the glossary entry for *postcoordinated expression*.

Alternatives

precoordinated expression
Precoordinated
Precoordination

Predicate migration

Steps taken to enable pre-existing data retrieval predicates (including queries, standard reports and decision support protocols) to be converted or utilized in a system using *SNOMED CT*.

Preferred term

The *term* that is deemed to be the most clinically appropriate way of expressing a *Concept* in a clinical record. The *Preferred Term* varies according to language and *dialect*.

Note: In *Release Format 2* the *Preferred Term* is indicated by the *acceptabilityId* field of a *Language Refset*.

Note: In *Release Format 1* the *Preferred Term* is indicated by a *Language Subset* and/or the *DescriptionType* field of the *Description file*.

Primitive concept

A *concept* with a formal logic definition that is not sufficient to distinguish its meaning from other similar *concepts*.

Note:

The meaning of *SNOMED CT concept* is expressed in a human-readable form by its *Fully Specified Name*. Each *concept* also has a formal logic definition represented by a set of defining *relationships* to other *concepts*. This logic definition is computer processable. A *primitive concept* does not have sufficient defining *relationships* to computably distinguish them from more general *concepts* (supertypes).

See also *sufficiently defined concept*.

Example: The *concept* 5596004|atypical appendicitis (disorder)| is *primitive* because the following definition is not sufficient to distinguish "atypical appendicitis" from any other type of "appendicitis".

- 116680003 | is a | = 74400008 | appendicitis |
- 116676008 | associated morphology | = 23583003 | inflammation |
- 363698007 | finding site | = 66754008 | appendix structure |

Figure 140: Definition of: |atypical appendicitis (disorder)| (primitive)

Q

Qualifying characteristic

An *attribute-value relationship* associated with a *concept* code to indicate to users that it may be applied to refine the meaning of the code. The set of qualifying *relationships* provide syntactically correct values that can be presented to a user for *postcoordination*. Example: 'Revision *status* ' = 'First revision' is a possible *qualifying characteristic* of 'Hip replacement'. A *qualifying characteristic* is contrasted with a *defining characteristic*. It is referred to in *CTV3* as a '*Qualifier*.

Alternatives

Qualifier

Quality characteristic

A type of attribute of a component by which its quality is assessed or measured.

Note: The set of *IHTSDO quality characteristics* are a typology of attributes of an *IHTSDO* Component by which its quality is assessed or measured. A typology is the study or systematic classification of types that have attributes or traits in common.

Quality metric

An agreed method and means for measuring levels of achievement, performance or conformance of a component or its *Quality characteristic*(s).

Quality target

An agreed level of achievement, performance or conformance of a component for any given *Quality characteristic*.

Query predicate

A statement of a condition that determines whether candidate instance data should be included in or excluded from a selection.

Note: *Query predicates* applied to a set of *SNOMED CT expressions* may test for subsumption of the overall meaning and/or may test the values applied to particular *attributes* in the *expression*.

R

Range

A constrained set of values that the *Concept Model* permits to be applied to a specific *attribute* when that *attribute* is applied to a *concept* in a particular *domain*.

Note: The *range* of permitted values that can be applied to an *attribute* is typically defined to include concepts in one or more branches of the subtype hierarchy.

Example: The *range* for values of 116676008 | Associated morphology | is subtypes of 49755003 | Morphologically abnormal structure |.

Alternatives

Concept model range

Read Code

A five-character code allocated to a *concept* or *term* in *CTV3*. Note that codes allocated in *Read Codes Version 2* and the *Read Codes 4-Byte Set* are also included in *CTV3*. The original 4-byte codes are distinguished from 5-byte codes in the general representation by prefixing them with a full stop.

Alternatives

Read Codes 4-Byte Set
Read Codes Version 2

Realm

A sphere of authority, expertise, or preference that influences the range of *components* required, or the frequency with which they are used. A *Realm* may be a nation, an organization , a professional discipline, a specialty, or an individual user.

Record services

Functions performed by software that interacts with a record system used to capture information which may include references to information in a terminology.

Note: *Record services* are intimately related to ways in which information is entered, stored and retrieved by a particular application. These services interact with *Terminology services* but, unlike *Terminology services* they are usually specific to a particular application.

Reference information model

A high-level generalized model that allows information to be represented and related consistently within a particular field of human endeavor.

Note: The *Health Level 7 Version 3 Reference Information Model* is the most widely used *reference information model* in health care.

Reference set

A work consisting of a set of references to *SNOMED CT components* that may associate additional properties with *components* that are members of the set and/or which may indicate associations between members of the set or between members of the set and content of another nomenclature, classification or knowledge structure. The uses of *Reference sets* include identification of subsets of *SNOMED CT* content, representation of alternative hierarchical structures and *maps* to classifications.

Alternatives

SNOMED CT reference set
Refset

Reference set member

A uniquely identified row within the *snapshot view* of a *reference set*.

Note:

1. Different versions of a *reference set member* may share the same identifier (*id*) but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.
2. Each *reference set* has an identifier (*refsetId*) and contains one or more *reference set members*. Each *reference set member* has its own unique identifier (*id*) which allows it to be versioned using the *effectiveTime* and *active* fields. All *reference set members* also contain a *referencedComponentId* (which refers to a component that is part of the set) and other fields that depend on the type of *reference set*.

Reference terminology

A terminology in which each *term* has a formal computer processable definition that supports meaning based retrieval and aggregation. *SNOMED CT* is a *reference terminology*

Relationship

An association between a source *concept* and a destination *concept*. The nature of the association is indicated by a reference to another *concept* referred to as the *relationship type*.

Notes:

1. Each *relationship* provides information about the source *concept*. In the example below
2. *Relationships* are represented by rows in the *Relationship File*

Example:

Table 312: Illustrative example of a *Relationship*

source	type	destination
74400008 | appendicitis |	363698007 | finding site |	66754008 | appendix structure |

Alternatives

SNOMED CT relationship

SNOMED CT Release

The content of a version of a *SNOMED CT Edition* that has been made available to licensees at a particular point in time.

Release file

A computer file used to distribute *SNOMED CT* content from the *IHTSDO* (or from the originator of an *Extension*) in a form that can be readily imported by a software application.

SNOMED CT release files follow one of the *release format* specifications *RF1* or *RF2*.

Alternatives

SNOMED CT release file
SNOMED CT distribution file

Release format

A file structure specified by the *IHTSDO* for files used to distribute *SNOMED CT* content.

Note: There are currently two *release formats*: *Release Format 1* and *Release Format 2*.

Alternatives

SNOMED CT release format
SNOMED CT distribution format

Release Format 1

The file structure specified by the *IHTSDO* for the files used to distribute *SNOMED CT* content in 2002.

Note: This format was replace by *Release Format 2* in January 2012, which is now the primary format for the *SNOMED CT International Release*. However, for backward compatibility *Release Format 1* files can be generated using a conversion utility and continue to be distributed available during an interim transitional period.

Alternatives

SNOMED CT Release Format 1
RF1

Release Format 2

The file structure specified by the *IHTSDO* for files used to distribute *SNOMED CT* content from 2011.

Note: See also: *Release Format 1*.

Alternatives

SNOMED CT Release Format 2
RF2

Release Type

The temporal scope and completeness of a *Release Format 2* file or set of files.

Table 313: SNOMED CT Release Types

Release Type	***Description***
Full	The files representing each type of component contain **every** version of every component ever released.
Snapshot	The files representing each type of component contain **one** version of every component released up to the time of the snapshot. The version of each component contained in a snapshot is the most recent version of that component at the time of the snapshot.

Release Type	*Description*
Delta	The files representing each type of component contain only component versions created since the previous release. Each component version in a *delta release* represents either a new component or a change to an existing component.

Root concept

The single *concept* that is at the top of the | SNOMED CT Concept | hierarchy.

Root metadata concept

The single *concept* that is at the top of the | SNOMED CT Model Component (metadata) | hierarchy.

Note: Most of the data in the metadata hierarchy is only relevant to *Release Format 2*. Therefore, this *concept* may not be present in some *Release Format 1* files.

Alternatives

Root metadata code

S

Situation with explicit context

A *concept* that specifically includes a definition the context of use of a clinical finding or procedure.

Example: "Family history of diabetes mellitus" is a situation with explicit *concept* because it defines the context as "family history". In contrast, "diabetes mellitus" is not a *situation with explicit context* because it can be used in many different situations including "family history", "past medical history", "current diagnosis", etc.

Note: A *situation with explicit context* is defined as a *subtype* of the situation to which it applies with an attribute associating it with the relevant clinical finding or procedure.

Alternatives

Explicit context
Clinical situation

Snapshot release

A *Release Type* in which the *release files* contain one version of every component released up to the time of the snapshot. The version of each component contained in a snapshot is the most recent version of that component at the time of the snapshot.

Snapshot view

A view of *SNOMED CT* that includes all the components in the state there were in at a specified point in time. A *Snapshot view* be provided by a fixed representation that matches the content of a *Snapshot release* or may be generated as a *Dynamic snapshot view* by filtering a *Full view*.

SNOMED

An acronym for the **S**ystematized**No**menclature of **Med**icine originally developed by the College of American Pathologists and now owned and maintained by the *IHTSDO*. *SNOMED Clinical Terms* is the most recent version of this terminology. It was preceded by *SNOMED RT* and *SNOMED International*.

SNOMED Clinical Terms

SNOMED CT is a clinical terminology maintained and distributed by the *IHTSDO*. It is considered to be the most comprehensive, multilingual healthcare terminology in the world. It was created as a result of the merger of *SNOMED RT* and *NHS Clinical Terms Version 3*.

Alternatives

SNOMED CT

SNOMED CT Edition

The combination of a *SNOMED CT Extension* with the *SNOMED CT International Edition* and, where relevant, any module from other *Extensions* on which the *SNOMED CT Extension* depends.

Note: A *SNOMED CT Edition* may be released by the provider of the *SNOMED CT Extension*. However, in general a *SNOMED CT Edition* is derived by combining the *SNOMED CT Extension* release files with relevant release data from the *SNOMED CT International Edition* and any other *Extensions* on which it depends.

Alternatives

Edition

SNOMED CT Extension

A set of terminology *components* and *derivatives* that add to and are dependent on the *SNOMED CT International Edition*, and are created, structured, maintained and distributed in accordance with *SNOMED CT* specifications and guidelines.

Notes:

1. *Components* that are created in an *extension* are identified using *extension SCTIDs*. These identifiers include an *extension namespace* which ensures that they do not collide with other *SCTIDs*, and can be traced to an authorized originator.
2. *Namespace identifiers* are allocated in response to requests from *IHTSDO Members* and *Affiliates*. For further information about this process and for access to the current SNOMED CT Namespace Register please refer to the *IHTSDO web page on Namespaces*.
3. *IHTSDO Members* may create, maintain and distribute *extensions* to address specific national, regional and language requirements. *IHTSDO Affiliates* may also create, maintain and distribute *extensions* to meet the needs of particular software solutions and customers.
4. See also *Edition* which refers to the combination of an *extension* with the *International Release* and, where relevant, any modules from other *extensions*on which it depends.

Alternatives

Extension

SNOMED CT Identifier

A unique *integer* identifier applied to each *SNOMED CT component* (*Concept*, *Description*, *Relationship*, *Subset*, etc.). Each *SCTID* includes an item identifier, a *check-digit*, a *partition identifier* and, depending on the *partition identifier*, may also include a *namespace identifier*.

Alternatives

Identifier
SCTID

SNOMED CT International Edition

The part of *SNOMED CT* that is maintained and distributed by the *IHTSDO* and available to all *IHTSDO Members* and *Affiliates* as the shared foundation of the terminology.

Notes:

1. The *International edition*, provided by the *IHTSDO*, may be supplemented by *Extensions* maintained by *IHTSDO Members* and *Affiliates* to meet additional national, local and organizational requirements.
2. The combination of the *International edition* with a *National Extension* is referred to as a *National Edition*.
3. The *International release* refers to a release of content from the *International edition* at a particular release date.

Alternatives

International edition

SNOMED CT International Release

The set of *release files* provided on a specified release date, to represent the part of the content of *SNOMED CT* that forms the common foundation to the terminology available to all *IHTSDO Members* and *Affiliates*.

Notes:

1. The *International release*, provided by the *IHTSDO*, may be supplemented by *Extension* releases provided by *IHTSDO Members* and *Affiliates* to meet additional national, local and organizational requirements.
2. See also *International Edition* which refers to the same general content, without specifying a particular release date.

Alternatives

International Release

SNOMED CT Module

A group of *SNOMED CT components* and/or *reference set members* that are at a given point in time managed, maintained and distributed as a unit.

Notes:

1. *Components* and *reference set members* that are part of the same *module* are share the same *moduleId* value.
2. Each *component* and *reference set member* is a part of one and only one *module* as at a given point in time.
3. The organization responsible for a *module* can move a *component* and *reference set member* from that *module* to another *module* that the same organization is responsible for, by creating a revised version of the *component* or *reference set member* with a different *moduleId* that applies from the *effectiveTime* of the revised version.
4. Subject to rules related to movement of components between two extensions or between an extension and the International Edition, it is possible for a *component* and *reference set member* to be moved between *modules* maintained by different organizations.

Alternatives

Module

SNOMED CT National Edition

The combination of a *National Extension* with the *SNOMED CT International Edition* and, where relevant, any module from other *Extensions* on which the *National Extension* depends.

Note: The *National Edition* may be made available to licensees at a particular release date as part of a *National Release*. However a *National Edition* can also be derived by combining the *National Extension* release files with relevant release data from the *SNOMED CT International Edition* and any other *Extensions* on which it depends.

Alternatives

National Edition

SNOMED CT National Extension

A *SNOMED CT Extension* that is maintained by an *IHTSDO Member* for use in a particular country.

Note: See also *National Edition* which refers to the combination of a *National Extension* with the *International Release* and, where relevant, any modules from other *Extensions*on which it depends.

Alternatives

National Edition

SNOMED CT National Release

A National Extension and/or *National Edition* as made available to licensees by an *IHTSDO Member* at a particular release date.

Notes:

1. The *National Release* is made available as a set of *release files* which contain components and derivatives from a National Extension maintained and distributed by an *IHTSDO Member*.
2. A *National release* may also include the *SNOMED CT International Release* on which it depends, in which case it is a release of the *National Edition*.
3. Alternatively, a *National Release* may consist only of the *National Extension release files* for the specified release date. In this case, the *National Edition* is generated by combining these files with the *International Release* on which it depends.

Alternatives

National Release

SNOMED International

SNOMED International is the version of *SNOMED* ® that was first released in 1993 and which, as version 3.5 released in 1998, It was the immediate predecessor of *SNOMED RT*.

SNOMED Reference terminology

The version of *SNOMED* ® prior to the collaborative effort to develop *SNOMED Clinical Terms*. It was one of the source terminologies, along with *CTV3*, from which *SNOMED CT* was developed.

Alternatives

SNOMED RT

Sponsored Territory

A Non-Member Territory that has been recognized and designated by the Licensor (*IHTSDO*) as a sponsored territory

Note: *SNOMED CT* may be used free of charge by *IHTSDO affiliates* and their sub-licensees in Sponsored Territories. Information about Sponsored Territories is published on the *IHTSDO* web site.

Stated view

The *stated view* of a *Concept* definition consists of the *Relationships* directly edited by terminology authors. It consists of the stated *subtype Relationships* plus the defining *Relationships* that exist prior to running a *Description Logic classifier*.

Note: The *Relationships* distributed in the main *Relationships* files are inferred from the stated *Relationships* using a *Description Logic classifier* to ensure consistency and completeness. The *stated view* is distributed in the *Stated Relationships File*.

Alternatives

Stated form

Statistical classification

A hierarchical organization of *terms* or ideas that allows aggregation into categories that can be counted and compared without double counting. A *statistical classification* is monohierarchical which means that each node in the *hierarchy* is part of one node is the level above. This avoids double counting but means that arbitrary decisions must be made where a node is naturally related to more than one parent. For example, in a *statistical classification* such as *ICD-10*, 'bacterial pneumonia' is be related to 'lung disorder' or 'infection disorder' but not to both.

Structure-Entire-Part

A modeling approach used in SNOMED CT to represent anatomical entities such as body organs, body systems, body regions,etc.

- **Stucture** is the most general way to refer to an organ, body system or region.
- **Entire** refers to a complete organ, body system or region.
- **Part** refers to a part of an organ, body system or region. It explicitly does not refer to the entire organ, body system or region.

Example: The figure below illustrates the relationships between the structure, entire and part concepts applied to a the heart.

- 80891009| heart structure |
- 302509004| entire heart |
- 119202000| heart part |

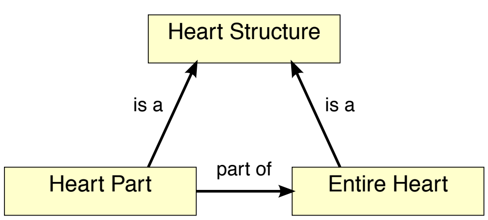


Figure 141: Structure-Entire-Part applied the heart

Alternatives

SEP

Subset

A set of *components* which part of and fully included with a larger set (e.g. a specified set of *Concepts* or *Descriptions*)

Notes:

1. In Release Format 2 the standard way to represent a *subset* of *components* is by using a *Simple Reference Set*
2. In Release Format 1 the term *subset* has the following special meaning:

   - A group of *components* (e.g. *Concepts*, *Descriptions* or *Relationships*) that share a specified common characteristic or common type of characteristic. *Subsets* represent information that affects the way the *components* are displayed or otherwise accessible within a particular *realm*, specialty, application or context.

   This special meaning arose from the "Subset Mechanism" which has now been replaced by *Reference Sets*. Therefore, except when referring to RF1 files the term *subset* should should now be used for its more correct general meeting.

Subsumption test

A test to determine whether a specified candidate *concept* or *expression* is a *subtype descendant* of another specified *concept* or *expression*.

Alternatives

Subtype test

Subtype

A specialization of a *concept*, sharing all the definitional attributes of the parent *concept*, with additional *defining characteristics*. For example, bacterial infectious disease is a *subtype* of infectious disease. Bacterial septicemia, bacteremia, bacterial peritonitis, etc. are *subtypes* of bacterial infectious disease (and infectious disease as well). *Subtype* is sometimes used to refer to the *concepts* in a *hierarchy* that are directly related to a parent *concept* via the | is a | *relationship*. In this usage, it is distinguished from *descendants* which explicitly includes *subtypes* of *subtypes*

Subtype child

A *concept* that has a direct | is a | *subtype Relationship* to a specified *concept*. See also *subtype* and *subtype descendant*.

Example:

The figure below shows an example hierarchy in which *concept* "E" has three *subtype children* (G, H and J).

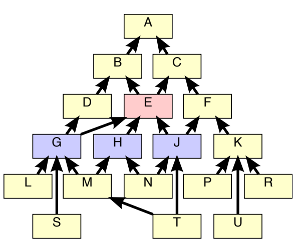


Figure 142: Hierarchy Illustration - Subtype children

Alternatives

Subtype children
Child
Children

Subtype classification

A classification hierarchy in which each node is connected to its supertypes. This allows aggregation of information based on a hierarchy of types.

Alternatives

Subtype hierarchy

Subtype descendant

All *subtypes* of a *concept*, including *subtypes* of *subtypes*. For example, if a *concept* has four *children*, then *descendants* are those *children* plus all the *concepts* that are descended from those four *children*. See also *subtype* and *subtype child*.

Example:

The figure below shows an example hierarchy in which *concept* "E" has eight *subtype descendants* (G, H, J, L, M, N, S and T).

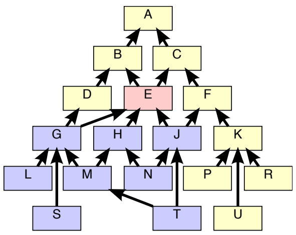


Figure 143: Hierarchy Illustration - Subtype descendants

Alternatives

Descendant

Sufficiently defined concept

A *concept* with a formal logic definition that is sufficient to distinguish its meaning from other similar *concepts*.

Note:

The meaning of *SNOMED CT concept* is expressed in a human-readable form by its *Fully Specified Name* (FSN) and has a formal logic definition represented by a set of defining *relationships* to other *concepts*. A *Sufficiently defined concept* has sufficient defining *relationships* to computably distinguish it from other *concepts*.

See also *primitive concept*.

Example: The *concept* 74400008|appendicitis (disorder)| is *sufficiently defined* by the following definition because any *concept* for which this definition was true would be the disorder "appendicitis".

- 116680003 | is a | = 18526009 | disorder of appendix |
- 116680003 | is a | = 302168000 | inflammation of large intestine |
- 116676008 | associated morphology | = 23583003 | inflammation |
- 363698007 | finding site | = 66754008 | appendix structure |

Figure 144: Definition of: |appendicitis (disorder)| (sufficiently defined)

Alternatives

Fully defined concept

Supertype ancestor

Any *concepts* of which the specified *concept* is a *subtype*. Includes the *supertype parents* and the *supertype parents* of each *supertype parent* and so on recursively until the *root concept* is reached.

Example:

The figure below shows an example hierarchy in which *concept* "T" has ten *supertype ancestors* A, B, C, D, E, F, G, H, J, and M).

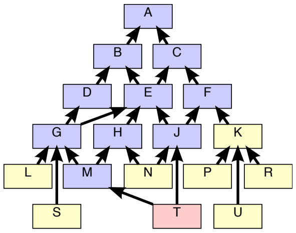


Figure 145: Hierarchy Illustration - Subtype ancestors

Alternatives

Ancestor

Supertype parent

A *concept* that is the target of a direct | is a | *subtype Relationship* from a specified *concept* (see also *supertype ancestor*).

Example:

The figure below shows an example hierarchy in which *concept* "T" has two *supertype parents* (J and M).

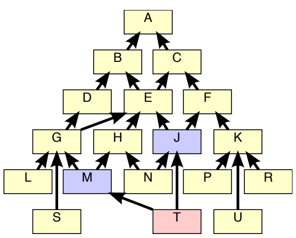


Figure 146: Hierarchy Illustration - Supertype parents

Synonym

A *term* that is an acceptable way to express a the meaning of a *SNOMED CT concept* in a particular *language*.

Note:

1. *Synonyms* are represented as *SNOMED CT descriptions* with the *typeId* value 900000000000013009 | Synonym |.
2. *Synonyms* allow representations of the various ways a *concept* may be described.
3. *Synonyms* (unlike *fully specified names*) are not necessarily unique because the same *term* can be used to describe more than one *concept*.
4. The *preferred term* is the *synonym* marked as preferred for use in the *Language reference set*for a given *language* or *dialect*.

T

Target code

A code or other *Identifier* within a *Target Scheme*.

Target scheme

A terminology, coding scheme or classification to which some or all *SNOMED CT Concepts* are mapped.

Technology Preview

An experimental status applied to a collection of *SNOMED CT release files* that represent a proposed additions of *components* and/or *derivatives* to the *SNOMED CT International Release* or to a *SNOMED CT Extension*. The *Technology Preview* status indicates the releasing party (*IHTSDO* or the owner of the *Extension*) is only releasing these additional *components* or *derivatives* for review and testing by implementers and other stakeholder. The objective of a *Technology Preview* is to test the chosen approach and elicit comments before committing to the content and/or release format for the additional material. It is likely that, prior to release of a *Candidate Baseline*, significant changes may be made to address comments made and issues identified by testing.

Note: The significance a *Technology Preview* release is that this data should not be used in an operational environment that may incorporate the data into a record or create a dependency on continued maintenance of the additional *components* or *derivatives*.

Term

A human-readable phrase that names or describes a *concept*. A *term* is one of the properties of a *description*. Other properties of a *description* link the *term* to an identified *concept* and indicate the type of *description* (e.g. *Fully Specified Name*, *Synonym*, etc.).

Terminology binding

A link between a terminology component and an information model artifact, such as class or attribute in a *electronic health record* or message.

Notes:

1. Terminology components include *SNOMED CT expressions*, *reference sets* and constraints.
2. Information model artifacts include classes and attributes in reference models for *electronic health records* and communication specifications.
3. *Terminology binding* can also be used to refer to the process of creating and persisting links between terminology components and information model artifacts.

Examples:

1. A set of coded values that may be applied to a particular attribute in an information model. The set may be expressed either explicitly (extensionally) or as a definitional constraint (intensionally).
2. The association between a named attribute value in the information model and a specific coded value or *expression*.
3. A rule that determines the way that a coded *expression* is constructed based on multiple attribute values in the information model.

Terminology server

Software that provides access to *SNOMED CT* (and/or to other terminologies). A *terminology server* typically supports searches and *Navigation* through *Concepts*. A server may provide a *user interface* (e.g. a *browser* or set of screen controls) or may provide low-level software services to support access to the terminology by other applications. See the *SNOMED CT Technical Implementation Guide*.

Alternatives

SNOMED CT terminology server

Terminology services

Functions performed by software that interacts with one or more representations of the terminology and provide access to information derived from the terminology.

Note: *Terminology services* can be generalized , so that they are independent of the way the terminology is used in a particular application. *Terminology services* may be used by *record services* that enter, store and retrieve information that includes *SNOMED CT expressions*. In contrast to *terminology services*, *record services* are usually specific to the design of a particular application.

Textual definition

An additional textual *description* applied to some *SNOMED CT concepts* that provides additional information about the intended meaning or usage of the *concept*.

Note:

Textual definitions are distributed in a file that follows the same structure as the *Description file* (RF2) but the terms permitted by the "textual definition" are much longer the 255 character limited applied to *synonyms* and *fully specified names*. Textual definitions are not essential for *SNOMED CT implementations* but they are useful as they provide narrative *Descriptions* of *concepts* that may be easier to understand than the shorter terms.

These *Descriptions* go beyond the detail of the *Fully Specified Name* as shown in the example below.

Example:

Table 314: Textual Definition

conceptId	*Fully Specified Name*	Textual Definition
11530004	Brittle diabetes mellitus (finding)	Diabetes mellitus in which there are frequent, clinically significant fluctuations in blood glucose levels both above and below levels expected to be achieved by available therapies.

Top level concept code

A *Concept* Code that is directly related to the *Root Concept* Code by a single *Relationship* of the *Relationship Type* I is a I. All *Concept* Codes (except for metadata *concepts*) are descended from at least one Top-Level *Concept* Code via at least one series of *Relationships* of the *Relationship Type*"I Is a I ".

Top level metadata code

A *Concept* Code that is directly related to the *Root Metadata Code* by a single *Relationship* of the *Relationship Type* I is a I. All Metadata *Concept* Codes are descended from at least one Top-Level Metadata *Concept* Code via at least one series of *Relationships* of the *Relationship Type*"I Is a I ".

Note: Most of the data in the metadata hierarchy is only relevant to Release Format 2. Therefore, this concept may not be present in Release Format 1 files.

Transitive closure

A comprehensive view of all the *supertype ancestors* of a *concept* derived by traversing all the I is a I *relationships* between that *concept* and the *root concept*.

Note: A *transitive closure* table represents the *transitive closure* of all *active concepts*.

Translation

The process of rendering text originally written in one language (source language) into another language (target language).

Translation source language

The language in which the original text is written.

Example: English is the source language for the *International edition* of *SNOMED CT*.

Alternatives

Source language

Translation target language

A language into which the original text is being translated or rendered.

Example: For the Spanish language edition, Spanish is the target language.

Alternatives

Target language

Translation Service Provider

Person or organization supplying a translation service.

Alternatives

TSP

U

Understandability, Reproducibility and Usefulness

Criteria applied to test the validity of new *concepts* and design features of *SNOMED CT*.

- Understandable: The meaning of a *concept* can be understood by an average health care provider, without reference to private or inaccessible information.
- Reproducible: Multiple users apply the *concept* to the same situations.
- Useful: The *concept* has a practical value to users that is self-evident or can be readily explained.

Alternatives

URU

Union

In set theory *union* of the sets A and B, is the set of all objects that are a member of A, or B, or both.

Note: Set theory is applied when describing the intended result of combinations of Reference Sets or Constraints.

User interface

The way a software application presents itself to a user including, its on screen appearance, the commands it puts at a users disposal, and the manner in which the user can access and update information by using the application.

Alternatives

UI

V

Value Set

A uniquely identifiable set of valid concept representations, where any concept representation can be tested to determine whether or not it is a member of the *value set*.

Notes:

1. This definition is used in *HL7* Vocabulary Working Group documents. In *SNOMED CT* a concept representation may be a *concept* identifier or a *SNOMED CT Expression*.
2. A *Reference set* can be used to represent a value set of *SNOMED CT concepts* each of which is represented by a *concept* identifier in the *referencedComponentId* field.

W

Word equivalent

A word or abbreviation that is stated to be equivalent to one or more other words, phrases or abbreviations for the purposes of textual searches of *SNOMED CT*. *Word Equivalents* and Phrase equivalents are represented as rows in the *Word Equivalents Table*.

Workbench

A set of *IHTSDO* sponsored software tools designed to support the development, maintenance, and use of *SNOMED CT* in health systems around the world.

Alternatives

IHTSDO Workbench

World Health Organization

the directing and coordinating authority for health within the United Nations system. The *World Health Organization* (*WHO* maintains the International *Statistical Classification* of Diseases and Related Health Problems (ICD).

Alternatives

WHO