Leading healthcare terminology, worldwide

Practical Guide to Postcoordination

v1.0

Publication date: 2023-05-24

Web version link: snomed.org/postcoordination

SNOMED CT document library: http://snomed.org/doc

This PDF document was generated from the web version on the publication date shown above. Any changes made to the web pages since that date will not appear in the PDF. See the web version of this document for recent updates.

SNOMED
International

Leading healthcare
terminology, worldwide

SNOMED International

Leading healthcare
terminology, worldwide

Table of Contents

SNOMED International

Leading healthcare terminology, worldwide

This document provides guidance on how to implement SNOMED CT postcoordinated expressions within a healthcare environment. A range of factors need to be considered, depending on what tasks are being performed. For example, implementations may require expressions to be authored, parsed, validated, stored, displayed, exchanged, classified or queried.

Web browsable version: snomed.org/postcoordination

SNOMED CT Document Library: http://snomed.org/doc

Executive Summary

SNOMED CT is a clinical terminology with a global scope covering a wide range of clinical specialties and requirements. Each concept in SNOMED CT represents a clinical idea to which a unique concept identifier has been assigned, but additional meanings can also be represented in SNOMED CT through expressions.

A SNOMED CT expression is a structured combination of one or more concept identifiers that represents a single clinical idea. A precoordinated expression is the simplest form of an expression. It contains a single concept identifier, and optionally one of the terms associated with the concept to make the expression human readable. A postcoordinated expression combines two or more concepts in accordance with rules expressed in the SNOMED CT Concept Model.

Postcoordination enables a wide range of clinical meanings to be captured in an EHR in a coded form, without requiring the terminology to include a separate concept for every detailed combination of ideas that may potentially need to be recorded.

Postcoordination supports a range of use cases, such as capturing clinical information in an EHR, sharing information across systems, supporting Natural Language Processing, and supporting mappings between SNOMED CT and other code systems.

Allthough, postcoordination may assist this range of use cases, the decision to use postcoordination requires careful consideration and preparation, because the implementation and support for expressions relies on the availability of human skills and dedicated tools supporting the safe creation, storage and processing of expressions.

Enabling postcoordination in an EHR requires an EHR design that is prepared to support the use of expressions in the same way as the released, precoordinated SNOMED CT content. Furthermore, it requires an EHR which uses a terminology server designed to enable the storage, processing and use of expressions.

Within a terminology server, the postcoordinated content lives alongside and is dependent on a specific versioned edition of SNOMED CT. The terminology server should, therefore, be designed to hold a given versioned SNOMED CT Edition, and an expression repository to hold the postcoordinated expressions. Furthermore, in addition to the services required by the EHR to use the content and derivatives of released SNOMED CT Editions, services are required to support the use of the postcoordinated content.

A terminology server may apply various levels of support for postcoordination. At the lowest level (level 0), the support for postcoordination involves the ability to accept only compositional grammar expressions that are fully compliant with the SNOMED CT Machine Readable Concept Model. At higher levels (level 1+), various patterns for non-mrcm compliant expressions are accepted and a higher degree of flexibility for the provided expressions are supported.

At present, it is recommended to only support postcoordination at either level 0 or 1, where the behavior and accepted expression types have been clearly specified. Further development is required to ensure that postcoordination at a higher level can be done without compromising clinical data quality.

1. Introduction

Background

SNOMED CT is a clinical terminology with a global scope covering a wide range of clinical specialties and requirements. Each concept in SNOMED CT represents a clinical idea to which a unique concept identifier has been assigned. Some concepts represent atomic, or simple, meanings, such as |Edema (finding)| , or |Foot structure (body structure)| , while other concepts represent more complex clinical meanings, such as |Atypical mycobacterial infection of lung (disorder)| , or |Positron emission tomography with computed tomography of heart (procedure)| . SNOMED CT concepts are also referred to as precoordinated concepts or Precoordinated expression.

Aside from the unique concepts, meanings can also be represented in SNOMED CT through postcoordinated expressions, which are structured combinations of two or more concept identifiers. The ability of SNOMED CT to support postcoordination is important because it enables a wide range of clinical meanings to be captured in a record, without requiring the terminology to include a separate concept for every detailed combination of ideas that may potentially need to be recorded. Thus, postcoordination greatly increases the depth of detail that SNOMED CT can represent, while avoiding a combinatorial explosion of precoordinated concepts. The SNOMED CT concept model provides the rules for the kinds of relationships that can be specified between particular types of concepts when creating postcoordinated expressions.

Objective and Scope

The objective of this guide is to provide a practical point of reference for the best practice principles for enabling postcoordination with SNOMED CT.

The chapters of this guide aim to answer the following questions:

- What characterizes a postcoordinated expression?
- What are the use cases for postcoordination?
- What are the prerequisites for enabling postcoordination?
- What approaches can be taken to enable postcordination in an Electronic Health Record?
- What are the steps involved in the creation, implementation, and use of postcoordinated expressions?
- What is required by a terminology server designed to support postcoordination?

Status

The guide is intended to provide an overview of the preliminary work done to facilitate the safe use of postcoordination with SNOMED CT within Electronic Health Records. It offers guidance on the best practices and considerations for implementing postcoordination effectively, and outlines potential risks and benefits associated with this approach. We encourage initiatives to follow the implementation approaches outlined in this guide on a trial basis to evaluate their effectiveness and suitability for your specific use case.

The document is a dynamic resource, and it will continue to advance as implementation experience and maturity expand. Its content will be regularly updated to reflect the latest advancements and insights in the domain. Users are encouraged to utilize this guide as a source for continuous learning and enhancement of their use of SNOMED CT and postcoordination.

Out of Scope

This document presents the design and technical implementation of an expression repository and terminology services required when enabling postcoordination in an EHR. However, some areas are not included in the current version, as they are still under investigation and will be added in the future as knowledge and experience in this are increase.

The does not offer detailed maintenance documentation for the expression repository, and implementing postcoordinated expressions in a production environment, therefore, requires careful analysis and establishment of required processes. The guide also lacks information on governance structures, work processes, and clinical validity beyond the rules specified in the Machine Readable Concept Model. Therefore, anyone attempting to support postcoordination in their EHR should follow the editorial guidelines to ensure logical consistency and clinical meaningfulness of meanings expressed through SNOMED CT.

Audience

This guide is targeted towards SNOMED CT Members and Affiliates who are involved in the creation, implementation, and maintenance of postcoordinated expressions.

- People involved in the design and development of SNOMED CT enabled solutions
  - This includes designers and developers of EHR systems, information models, data entry interfaces, storage systems, decision support systems, retrieval and analysis systems, communication standards, and terminology services

Document Overview

This document presents a practical guide to postcoordination with SNOMED CT and is structured as follows:

- 1. Introduction: This chapter explains the background, purpose, scope, audience and overview of the document
- 2. SNOMED CT Expressions: This chapter introduces SNOMED CT expressions, describing the syntax, structure and forms of expressions. It also outlines the benefits, challenges and considerations associated with the decision to support postcoordinated expressions
- 3 System Components: This chapter introduces the components included in the design of systems enabled to support postcoordination
- 4. Expressions in the EHR: This chapter introduces and exemplifies how expressions may be used within an Electronic Health Record to support key tasks including clinical data entry, storage, display, exchange and retrieval
- 5. Expressions in a Terminology Server: This chapter describes the requirements and design of a terminology server supporting postcoordination. It introduces various levels of implementations that have been specified to ensure the safe implementation of postcoordination, and it documents how the design can be met complying with RF2 or HL7 FHIR

Feedback

We invite SNOMED CT Members and Affiliates to try this initial version of the guide. It will be extended and enhanced following feedback. Please send any comments or questions to info@snomed.org.

2. SNOMED CT Expressions

This chapter provides an introduction to SNOMED CT expressions and covers several aspects related to their representation and use.

It begins by outlining the key differences between pre- and postcoordination, including their relative benefits and limitations. It covers the syntax used to represent expressions, providing a detailed overview of the various expression parts and operators that are used. The chapter introduces the different forms that expressions can take, and it discusses the role of the concept model and editorial guidance required to ensure the accuracy and consistency of expressions.

2.1 Precoordination and Postcoordination

SNOMED CT can be considered as a knowledge graph (direct acyclic graph), where each concept is represented as node in the graph structure, and the position of each node corresponds to its meaning and relationship to other concepts in the graph. These positions can be thought of as coordinates. Precoordinated concepts are predefined and accurately placed in the graph, while postcoordinated expressions require coordination into the graph by a description logic classifier. This is necessary to ensure that the combined concepts are positioned correctly in the graph to represent their relationships, and to allow for effective search and retrieval of the resulting concepts. In this context, the use of coordinates refers to the position of nodes in the SNOMED graph, and precoordination and postcoordination refer to the placement of concepts in the graph at a certain moment in time.

Concepts and Expressions

A SNOMED CT concept represents a clinical idea with a unique SNOMED CT identifier that is distributed as part of a SNOMED CT release. Concepts are linked to terms by descriptions, which provide a human-readable representation of the meaning of the concept. Additionally, they are related to each other by relationships, where each relationship represents a defining property of the concept being the source of the relationship. These defining relationships provide a machine-processable representation of the concept's meaning.

Aside from the human-readable representation, all released SNOMED CT concepts can be represented in various ways, three of which are:

- their concept identifier, which uniquely identifies the concept
- their stated view, represented through the explicit representation of their defining properties as stated by the author of the concept
- their inferred view, which represents the formal definition of concepts as logically derived by applying a description logic classifier to the stated view

Table 2.1-1 provides an example of three different views that can be used to refer to the clinical meaning 'appendicitis' .

Concept: "Appendicitis"				
Concept Identifier	**Stated view**		**Inferred view**	
74400008	**Relationship type**	**Value**	**Relationship type**	**Value**
	116680003 |Is a|	64572001 |Disease|	116680003 |Is a|	18526009 |Disorder of appendix|
	363698007 |Finding site|	66754008 |Appendix structure |	116680003 |Is a|	302168000 |Inflammation of large intestine|
	116676008 |Associated morphology |	409774005 |Inflammatory morphology|	363698007 |Finding site|	66754008 |Appendix structure |

Concept: "Appendicitis"				
			116676008 |Associated morphology|	409774005 |Inflammatory morphology|

Table 2.1-1: Example of ways of representing the meaning conveyed by the concept 'Appendicitis'.

SNOMED CT Expressions

To support a standardized and computer-processable way of representing and referring to clinical meanings in SNOMED CT, irrespective of whether this involves a single concept identifier, or a combination of concept identifiers, SNOMED CT supports expressions.

A SNOMED CT expression is a structured combination of one or more concept identifiers that represents a single clinical idea. SNOMED International distinguish between two main types of expressions, i.e. precoordinated expressions and postcoordinated expressions. Table 2.1-2 shows examples of both types of expressions.

SNOMED CT Expression	
Precoordinated expression	**Postcoordinated expression**
An expression containing a single concept identifier	An expression containing two or more concept identifiers
31978002 |Fracture of tibia (disorder)|	64572001 |Disease (disorder)| : { 363698007 |Finding site (attribute)| = 12611008 |Bone structure of tibia (body structure)| , 116676008 |Associated morphology (attribute)| = 72704001 |Fracture (morphologic abnormality)| }

Table 2.1-2: SNOMED CT expression types.

Precoordinated Expression

A precoordinated expression is the simplest form of an expression. It contains a single concept identifier, and optionally one of the terms associated with the concept to make the expression human readable. For example, a precoordinated expression, which means “fracture of tibia” can be as represented as “31978002”, optionally followed by the term ‘fracture of tibia’ placed in vertical bars known as ‘pipes’. This means that all available concepts in SNOMED CT can be represented as precoordinated expressions, by using the assigned concept identifier.

Examples of precoordinated expressions are:

- 765472003 | Excision of left kidney (procedure)|
- 108365000 | Infection of skin (disorder)|
- 387458008 | Aspirin (substance)|
- 73211009 | Diabetes mellitus (disorder)|
- 423827005 | Endoscopy (procedure)|

Postcoordinated Expression

Postcoordinated expressions contain two or more concept identifiers and are used to specify a clinical meaning by referencing existing SNOMED CT concepts.

For example, the clinical meaning "Pain in left arm" can be represented by the following expression including three SNOMED CT concepts:

```
102556003 |Pain in upper limb| : 272741003 |Laterality| = 7771000 |Left|
```

Another example is the clinical meaning of "Edema of right lung", which can be represented as:

```
267038008 |Oedema| : 363698007 |Finding site| = 3341006 |Right lung structure|
```

The SNOMED CT Compositional Grammar is used as the syntax for expressions, and the SNOMED CT Concept Model rules determine how concepts can be combined. These topics are elaborated in the subsequent pages of this guide.

Postcoordinatied expressions are useful to support the authoring of new concepts in SNOMED CT, but they may also be created to support specific implementation use cases, e.g. when no precoordinated concept is available for a specific clinical meaning, or when an information model or implementation strategy requires a certain representation of the clinical meanings.

The ability to create postcoordinated expressions greatly increases the depth of detail that SNOMED CT can represent without having to include every possible specific site for every possible disorder via a released concept.

2.2 Expression Syntax

Compositional Grammar

The syntax used to represent SNOMED CT expressions is called the Compositional Grammar. This is a formal grammar describing the symbols you can use and how the different parts of an expression can be constructed. Please refer to Compositional Grammar - Specification and Guide for the full detail on this.

While alternative syntaxes are possible, the compositional grammar syntax is considered to be the normative standard for interoperability purposes. Therefore, the recommended practice for anyone supporting postcoordination within their clinical information system, is to represent expressions using this syntax.

Structure

Figure 2.2-1 illustrates the key parts of a SNOMED CT expression through an example. The expression used in this example represents the stated view of the concept 16554361000119106 | PET CT of brain| .

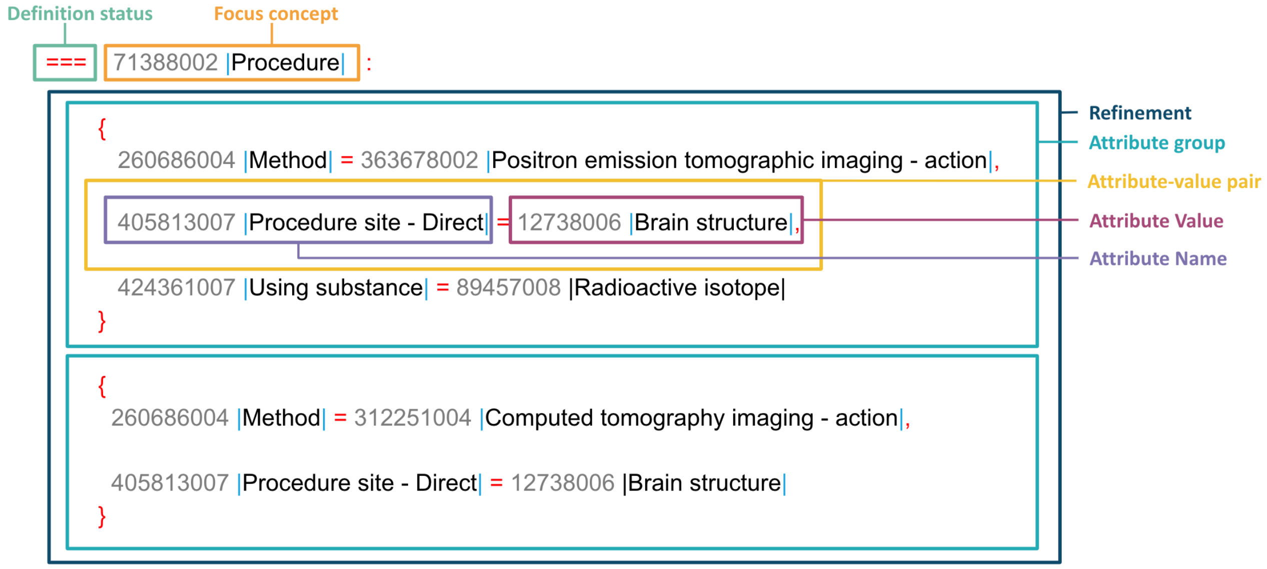

Definition status
Focus concept
=== 71388002 |Procedure| :
Refinement
Attribute group
Attribute-value pair
Attribute Value
Attribute Name
{
260686004 |Method| = 363678002 |Positron emission tomographic imaging - action|,
405813007 |Procedure site - Direct| = 12738006 |Brain structure|,
424361007 |Using substance| = 89457008 |Radioactive isotope|
}
{
260686004 |Method| = 312251004 |Computed tomography imaging - action|,
405813007 |Procedure site - Direct| = 12738006 |Brain structure|
}

Figure 2.2-1: Expression parts.

A SNOMED CT Compositional Grammar expression begins with an optional definition status, contains one or more focus concepts (represented by a concept identifier) and optionally has a refinement. Each refinement may contain grouped or ungrouped attributes (or both). An attribute consists of the attribute name (represented by a concept identifier) together with the value of the attribute. The attribute value is either an expression or a concrete value (i.e. string, integer, decimal or boolean). Please note that, in the context of an expression, the word 'Attribute' is used to refer to the name/value pair within a refinement. However, in the context of the SNOMED CT concept model, the word 'Attribute' is often used to refer specifically to the 'attribute name' (i.e. the concept used as the relationship type).

Expression part	Description	General	Recommendation for the implementation of postcoordination
Definition status	A SNOMED CT postcoordinated expression may be used to express a clinical meaning that is either equivalent to or a subtype of the given expression.	'===' is used to state that the expression (on the right-hand side) is semantically equivalent to the left-hand side. '<<<' is used to state expression (on the right-hand side) is a subtype of the left-hand side	For the scope of this guide, the default definition status for postcoordinated expressions is equivalent, i.e. '==='. This means that, unless stated otherwise, the expression represents the full semantics of what is being represented. This principle supports an unambiguous interpretation of the meaning of the expression.
Focus concept	The part of a SNOMED CT expression that represents the primary clinical idea.	According to the compositional grammar syntax, SNOMED CT expressions may contain multiple focus concepts. Often, the inferred view of a precoordinated concept includes multiple focus concepts, representing the direct parents of the concept.	For the scope of this guide, all expressions may only contain a single focus concept. The reason for this is that multiple focus concepts in a single postcoordinated expression can lead to ambiguity and confusion. Restricting expressions to one focus concept supports clarity and precision of the clinical meaning being represented, making it easier for healthcare providers to understand and use the information accurately.

Refinement	The part of a SNOMED CT expression that modifies or adds defining properties to the focus concept.	According to the compositional grammar syntax, the refinement of a SNOMED CT expression is everything to the right of the ':' .	Postcoordinated expressions may include a refinement to specify either a refinement or a qualification of the focus concept.
Attribute group	An association between a set of attribute value pairs that causes them to be considered together within a concept definition or postcoordinated expression.	An expression may contain multiple attribute groups, and each attribute group is surrounded by curly braces. When all attribute-value pairs in a refinement belong to the same attribute group the braces around the attribute group are optional.	Systems enabling postcoordination may support specific transformations of expressions to ensure that appropriate groups are applied for certain attribute-value pairs or set of attribute-value pairs. Such patterns may be defined to support specific use cases while ensuring the correct inferred view of the expression.
Attribute-value pair	A combination of an attribute name and an attribute value used to specify a defining characteristic of a clinical idea or a concept.	Each attribute-value pair represents a characteristic that applies to the meaning being expressed.	Postcoordinated expressions may contain one or more attribute-value pairs.
Attribute name	The concept that represents the attribute type in a defining relationship or postcoordinated expression.	Each attribute-value pair has a name which is represented by a concept. All of the concepts that can be used to name attributes are subtypes of the concept 410662002 | Concept model attribute (attribute)| . The SNOMED CT concept model specifies the domain for all attributes. The domain for an attribute is the set of concepts which may be defined or refined, using the particular attribute	Attribute-names used in the refinement of a postcoordinated expression should be valid for the domain determined by the focus concept. If implementations extend the domain for certain attributes, transformations should be supported to convert the expressions into a concept-model compliant format. *For example, adding a laterality to a procedure violates the concept model rules, but transformations may be applied to put the expression into a valid form where the laterality applies to the body structure being the site of the given procedure.*
Attribute value	A concept that represents the target of a relationship or the value of an expression refinement in a postcoordinated expression.	An attribute name is associated with a value (that creates an attribute-value pair) when used in the definition of a concept or in a postcoordinated expression. The permitted range of values for an attribute depends on the rules specified in the SNOMED CT concept model.	Attribute values should belong to the acceptable range for the selected focus concept and attribute. Note that although an attribute value fall within the range of the attribute, the focus concept of the expression may alter/ constraint the acceptable range. *For example, stating a |finding site (attribute)| of |eye structure| to the clinical finding |fracture of femur| will not be valid, despite the fact that |eye structure| falls within the allowed range of the |finding site (attribute)|. The definition of the focus concept (|fracture of femur|) constraints the allowed range to the concept 71341001 | Bone structure of femur|, or any subtype hereof.*

Refinement Types

Overall, postcoordination may be performed to either *refine* or *qualify* the meaning of a SNOMED CT concept. The refinement part of a postcoordinated expression may thus represent either a 'refinement' or a 'qualification' of the stated focus concept.

Refinement

Refining the meaning of a focus concept is done by applying more specific values to particular attributes of the concept. The definition of the focus concept, therefore, determines the refinements that are possible for the concept.

Consider the following expression representing the meaning 'Spiral fracture of tibia':

```
31978002 |Fracture of tibia (disorder)| : 116676008 |Associated morphology (attribute)| = 73737008 |
Fracture, spiral (morphologic abnormality)|
```

The focus concept of this expression has the following stated definition:

```
64572001 |Disease (disorder)| :
{ 363698007 |Finding site (attribute)| = 12611008 |Bone structure of tibia (body structure)| ,
  116676008 |Associated morphology (attribute)| = 72704001 |Fracture (morphologic abnormality)| }
```

The attribute value of used in the expression (|Fracture, spiral (morphologic abnormality)|) is a subtype of the attribute-value present in the definition of the concept (|Fracture (morphologic abnormality)|), as shown below.

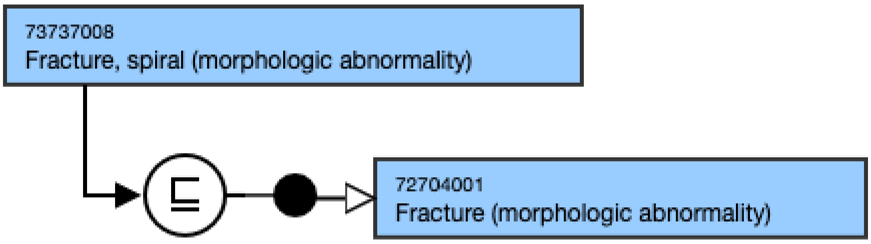

73737008
Fracture, spiral (morphologic abnormality)
⊑
72704001
Fracture (morphologic abnormality)

Figure 2.2-2: 73737008 |Fracture, spiral (morphologic abnormality)| is a subtype of 72704001 |Fracture (morphologic abnormality)|.

Qualification

Qualifying the meaning of a concept can be done by adding a qualifying characteristic to a concept, e.g. by applying and attribute-value pair to the focus concept, where that attribute-value pair does not refine an existing attribute-value pair present in the definition of the focus concept. If a particular qualifying characteristic is applied to a concept, the resulting expression represents a more tightly defined subtype of that concept. All attribute-value pairs stated to qualify a focus concept should fall within the acceptable domain and range determined by the SNOMED CT concept model.

Examples of qualifications include:

The concept 53084003 | Bacterial pneumonia (disorder)| may be qualified according to its clinical course (<< 288524001 |Courses (qualifier value)|) or severity (<< 272141005 |Severities (qualifier value)|)

```
53084003 |Bacterial pneumonia (disorder)| : 263502005 |Clinical course (attribute)| = 385315009 |Sudden
onset (qualifier value)|
```

The concept 71651007 |Mammography| may be qualified according to its priority (<< 272125009 |Priorities (qualifier value)|) or intent (<< 363675004 |Intents (nature of procedure values) (qualifier value)|)

71651007 |Mammography (procedure)| : 363703001 |Has intent (attribute)| = 360156006 |Screening - procedure intent (qualifier value)|

Examples

Expression	Type	Inferred view of focus concept	Description
125605004 |Fracture of bone (disorder)| : 363698007 |Finding site (attribute)| = 12611008 |Bone structure of tibia (body structure)|	Refinement	=== 284003005 |Bone injury (disorder)| :{ 363698007 |Finding site (attribute)| = 272673000 |Bone structure (body structure)| , 116676008 |Associated morphology (attribute)| = 72704001 |Fracture (morphologic abnormality)| }	This expression represents a refinement of the focus concept 125605004 | Fracture of bone (disorder)|, because the stated finding site value (12611008 | Bone structure of tibia (body structure)|) is a subtype of 272673000 |Bone structure (body structure)|
108022006 |Kidney excision (procedure)| : 405813007 |Procedure site - Direct (attribute)| = 18639004 |Left kidney structure (body structure)|	Refinement	=== 108189003 |Abdomen excision (procedure)| + 175898006 |Kidney operation (procedure)| + 118959001 |Removal from urinary tract (procedure)| :{ 260686004 |Method (attribute)| = 129304002 |Excision - action (qualifier value)| , 405813007 |Procedure site - Direct (attribute)| = 64033007 |Kidney structure (body structure)| }	This expression adds a more specific procedure site value than the one used in the definition of the focus concept
195967001 |Asthma (disorder)| : 263502005 |Clinical course (attribute)| = 424124008 |Sudden onset AND/OR short duration (qualifier value)|	Qualification	<<< 50043002 |Disorder of respiratory system (disorder)| :{ 363698007 |Finding site (attribute)| = 89187006 |Airway structure (body structure)| }	This expression adds an attribute-value pair which does not represent or refines a defining property of the focus concept.
40733004 |Infectious disease (disorder)| : 363698007 |Finding site (attribute)| = 39937001 |Skin structure (body structure)|	Qualification	=== 64572001 |Disease (disorder)| :{ 370135005 |Pathological process (attribute)| = 441862004 |Infectious process (qualifier value)| }	This expression adds an attribute-value pair which does not represent or refines a defining property of the focus concept.

2.3 Concept Model and Editorial Guidance

SNOMED CT Concept Model

The SNOMED CT Concept Model is a set of rules that govern the ways in which SNOMED CT concepts are permitted to be modelled using relationships to other concepts. These rules are critical to the consistent modelling of SNOMED CT content, which in turn determines the extent to which reproducible logical inferences can be drawn. These logical inferences are the foundation for effective use of SNOMED CT for retrieval and reuse of clinical information.

For each attribute (relationship type) defined in SNOMED CT, the concept model specifies a set of rules to support the proper application of SNOMED CT attributes, including:

- Attribute **Domain**: The set of concepts to which the attribute may be applied
- Attribute **Range**: The set of concepts which may be used as the value for the attribute
- Attribute **Cardinality**: The minimum and maximum number of times that the attribute may appear in a concept definition
- Attribute **Grouping**: Whether an attribute may or may not belong to a relationship group
- Attribute **In-group cardinality**: The minimum and maximum number of times that the attribute may appear in each relationship group

ⓘ In addition to the features presented above, the concept model includes additional information, such as specifying parent domains, rule strength and scope.for more information, please refer to the MRCM specification: http://snomed.org/mrcm

Machine Readable Concept Model

The Machine Readable Concept Model (MRCM) represents rules of the SNOMED CT concept model in a form that can be read by a computer and applied to test that concept definitions and expressions comply with the rules.

Table 2.3-1 shows a few example of concept model rules, as they can be derived from the MRCM reference sets.

Domain	Attribute	Grouped	Cardinality	In Group Cardinality	Range Constraint	Rule Strength	Scope
<< 404684003 |Clinical finding|	116676008 |Associated morphology|	1	0..*	0..1	<< 49755003 |Morphologically abnormal structure|	Mandatory concept model rule	All SNOMED CT content
<< 404684003 |Clinical finding|	246075003 |Causative agent|	1	0..*	0..1	<< 105590001 |Substance| OR << 260787004 |Physical object| OR << 373873005 |Pharmaceutical / biologic product| OR << 410607006 |Organism| OR << 78621006 |Physical force|	Mandatory concept model rule	All SNOMED CT content
<< 404684003 |Clinical finding|	363698007 |Finding site|	1	0..*	0..1	<< 442083009 |Anatomical or acquired body structure|	Mandatory concept model rule	All SNOMED CT content
<< 71388002 |Procedure|	260686004 |Method|	1	0..*	0..1	<< 129264002 |Action|	Mandatory concept model rule	All SNOMED CT content
<< 71388002 |Procedure|	260870009 |Priority|	1	0..*	0..1	<< 272125009 |Priorities|	Mandatory concept model rule	All SNOMED CT content
<< 387713003 |Surgical procedure|	424876005 |Surgical approach|	1	0..*	0..1	<< 103379005 |Procedural approach|	Mandatory concept model rule	All SNOMED CT content

Table 2.3-1: Example of concept model rules.

Examples of Use

Given the machine processable nature of SNOMED CT, the Compositional Grammar and the MRCM, postcoordinated expressions may be tested against the rules specified in the MRCM, for example to validate the expression, or to support the creation of MRCM-compliant expressions.

To support the creation and validation of SNOMED CT postcoordinated expressions, the following MRCM process can be used:

1. Determine which domains the expression belongs to using the |MRCM domain reference set|
   - An expression belongs to a given domain if any focus concept is either:
     i. Valid when tested against the associated *domainConstraint*; or
     ii. Valid when tested against the associated *proximalPrimitiveConstraint* and all required refinements in the *proximalPrimitiveRefinement* either match a defining relationship on the given focus concept, or match a refinement condition added to the expression being authored;
2. Determine the set of valid attributes for the given domains using the |MRCM attribute domain reference set| and allow refinements to be added using attribute concepts from this set;
3. For each attribute used to define the concept, ensure that the grouping and cardinality are valid according to the rules specified in |MRCM attribute domain reference set| for the given attribute and parent domain;
4. Determine the valid range for each attribute using the *rangeConstraint* in |MRCM attribute range reference set| .

Please note that in the above process only rules with a contentType = << 723595009 |All postcoordinated SNOMED CT content| should be used. Rules with a ruleStrength of |Mandatory concept model rule| should be enforced when authoring and cause an error during validation, while rules with a ruleStrength of |Optional concept model rule| should be used as a recommendation for authoring and result in a warning during validation.

An alternative approach to authoring and validating postcoordinated expressions is to use the *domainTemplateForPostcoordination* from the |MRCM domain reference set| to ensure compliance with the full set of attribute rules. The *domainTemplateForPostcoordination* can also be specialized into an expression authoring template that meets the needs of a particular use case, while still conforming to the overall rules of the domain.

Editorial Guidance

In addition to the concept model, SNOMED International also rely on editorial principles when authoring the content of SNOMED CT. The SNOMED CT Editorial Guide covers a wide range of topics, including the principles of SNOMED CT, and the structure and organization of the terminology. In addition, it documents guidelines for the modelling of content within specific domains and sub-domains that and it defines the role of the attributes available. The SNOMED CT Editorial Guide plays a critical role in ensuring the quality of SNOMED CT content, as it documents the authoring principles that require human analysis and interactions to ensure that concepts are accurate, consistently modelled, and clinically relevant.

Implementing postcoordination with SNOMED CT is similar to authoring SNOMED CT concepts, as it involves creating clinical meanings by combining existing concepts. However, it is crucial to ensure that the implementation of postcoordination does not impact clinical safety.

It is essential to involve relevant stakeholders such as clinicians, informaticians, and quality assurance personnel to ensure that the approach taken is appropriate and safe for use in clinical practice. Thorough testing and evaluation of the postcoordinated expressions can help identify potential safety concerns early in the implementation process, ultimately leading to safer and more effective healthcare.

2.4 Expression Forms

Semantic Forms

Similar to the stated and inferred view of precoordinated concepts, as described in 2.1 Precoordination and Postcoordination, the meaning represented by postcoordinated expressions can also be represented in different ways. When enabling postcoordinated expressions within a system, it is important to be support these different views and understand why they are needed.

Close to User Form

The Close to User Form (CTU) Expression is the faithful representation of the clinical meaning as it was entered by the user. It is the{ primary stored and communicated view of the clinical information that's encoded using SNOMED CT. The CTU Expression includes any refinement that was applied by the system, based on the selections made in a data entry form, or those made explicitly.

And, importantly, it does **not** include any additional relationships that are added, based on classifier rules to make the expression complete or to normalize it. It only includes the parts of the clinical meaning that were specifically intended by the user.

Classifiable Form

The Classifiable Form (CF) expression is a syntactically valid and concept model compliant representation of the CTU expression. It serves as the input to the classifier, enabling expressions to be classified together with other SNOMED CT content.

And in the classifiable form, any ungrouped attributes and role groups have been validated and appropriately applied to the definition of the focus concept.

Necessary Normal Form

The Necessary Normal Form (NNF) expression is the inferred view of the expression, and it includes all the relationships that are necessarily true, with redundancy removed. This is the output of the process that took the classifiable form expression and classified it with a given SNOMED edition. The NNF represents the necessary relationships used for querying, so this becomes part of your substrate when you're running an Expression Constraint query.

The NNF includes refinements that represent inferred relationships, without any redundant refinements or redundant relationship groups.

Objective of Different Semantic Forms

Within the released SNOMED CT content, the *stated* definitions of SNOMED CT concepts are represented as Description Logic OWL axioms, and these are the clinical definition that SNOMED authors state when they define the meaning of a concept. For postcoordinated expressions, the CTU expression can be compared to the stated concept definition, as it represents the expression as it was created.

When querying SNOMED CT concepts, the inferred definitions are used. The *inferred* definition of a concept is derived (or inferred) by classifying the stated definitions using a Description Logic reasoner. The output of this classification process is represented in the release using the relationship table and includes the necessary normal form for each concept. The necessary normal form represents the set of 'necessary' inferred relationships. For more information on concept definitions, please refer to Appendix D of the release file specification: Concept Definition Illustrations. In the same way as for precoordinated concepts, the inferred view, or the NNF expressions, needs to be generated to support querying over postcoordinated expressions, see Figure 2.4-1.

To enable the classification of the CTU expression, a transformation is required to provide a form that can serve as input to a Description Logic reasoner. Such reasoner requires a form which is syntactically valid and complies to the concept model rules. This is why all support for postcoordination also requires the ability to generate the CF of each expression.

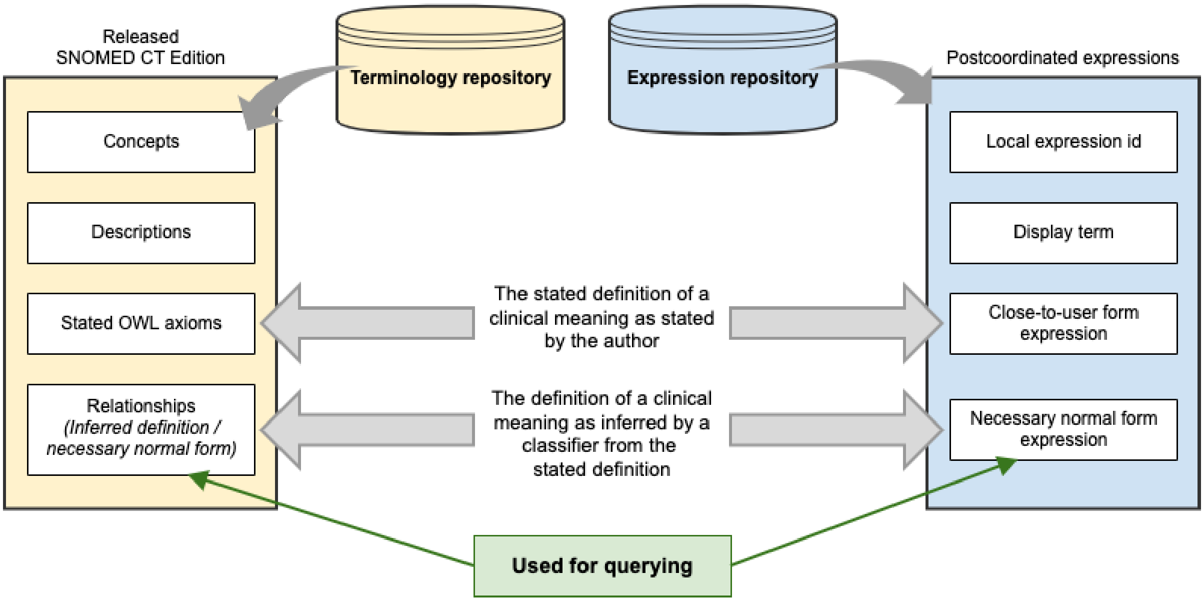

Released
SNOMED CT Edition
Terminology repository
Expression repository
Postcoordinated expressions
Concepts
Descriptions
Stated OWL axioms
Relationships
(Inferred definition / necessary normal form)
Local expression id
Display term
Close-to-user form expression
Necessary normal form expression
The stated definition of a clinical meaning as stated by the author
The definition of a clinical meaning as inferred by a classifier from the stated definition
Used for querying

Figure 2.4-1: Expression forms compared to the representations of precoordinated content.

Canonical Form

Although the Compositional Grammar syntax supports a standard way of representing SNOMED CT expressions, expressions that contain exactly the same concept identifiers and refinements, may still differ from one another in the following ways:

- Inclusion of whitespace between elements
- Inclusion of specific terms associated with identified concepts
- The order in which focus concepts, refinements, attributes, and attribute groups appear

The canonical form of an expression is a serialized representation produced by applying a set of rules that ensure a single unique representation for any expression.

The following rules should be followed to order the expression (click to show/hide)

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 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
- Definition status indicator
    - Equal to indicator [===] - to represent semantic equivalence
    - Subtype of indicator [<<<] - to represent a primitive clinical meaning

The syntax determines the general order of elements within an expression as follows:

- Focus conceptIds
    - 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.
- Attributes (expressed as name-value pairs)
    - 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.
- Groups (containing attributes)
    - Groups are sorted by alphabetical order of the combined set of previously sorted attributes.

Examples

Original Expression	Canonical Form
195967001 |Asthma| : 246112005 |Severity| = 24484000 |Severe|	195967001:246112005=24484000
417076003 |Dislocation of shoulder joint| : 272741003 |Laterality| = 24028007 |Right|	417076003:272741003=24028007
71388002 |Procedure| :{ 405815000 |Procedure device| = 122456005 |Laser device| , 260686004 |Method| = 129304002 |Excision - action| , 405813007 |Procedure site - direct| = 15497006 |Ovarian structure| }	71388002:{260686004=129304002,405813007=15497006,405815000=122456005}
64572001 |Disease| :{ 363698007 |Finding site| = 12611008 |Bone structure of tibia| , 116676008 |Associated morphology| = 72704001 |Fracture| }	64572001:{116676008=72704001,363698007=12611008}

Table 2.4-1: Examples of expressions and their canonical form.

2.5 Benefits and Challenges

Postcoordination with SNOMED CT offers significant benefits for users who need to capture and express clinical meanings accurately. Postcoordination enhances the flexibility and expressivity of SNOMED CT, making it more adaptable to current clinical practice. However, postcoordination also presents a variety of challenges, encompassing both human and technical issues that are crucial to handle as part of an implementation, ensuring that postcoordination is used judiciously and consistently to maximize its benefits while minimizing its potential drawbacks.

Benefits

Two overall benefits can be described for postcoordination:

Postcoordination supports the ability to express or capture new meanings or clinical ideas that are not already in SNOMED CT

SNOMED CT implementers without the means to create and manage a proper extension may require clinical meanings to be represented using SNOMED CT without being dependent on specific precoordinated concepts existing. In such cases, postcoordination may be considered as part of a SNOMED CT implementation.

Postcoordination supports the ability to compose clinical meanings from separate input values and then classify to identify existing precoordinated content

Some SNOMED CT implementations are dependent on a fixed information structure, user requirements may enforce a specific way of entering clinical data, or new clinical meanings are created dynamically (for example within a natural language processing tool). These are cases in which implementations may require clinical meanings to be represented as expressions.

Challenges

To realize the benefits of postcoordination, careful analysis of the following described challenges needs to be undertaken and solutions need to be implemented.

Tooling and guidance in the area of postcoordination are still in a developmental stage

A key challenge for the implementation of postcoordination is the lack of mature or well-developed tools and practical experience within this area.

Creating expressions, in a way that conforms to all the rules and consistent with existing content in order to classify correctly, is not simple!

As described in 2.2 Expression Syntax and 2.3 Concept Model and Editorial Guidance, expressions that represent the desired semantics must follow both terminological and editorial principles of SNOMED CT. In addition, expressions should be consistent with the modeling applied for the released SNOMED CT content. Therefore, the creation of expressions should be performed by people with the required knowledge, or assisted by tools that facilitate the proper creation of expressions.

Expressions are tied to a specific SNOMED CT release and upgrading that release requires maintenance

Maintenance processes need to be established to ensure that the expressions used in the system continue to work properly with the updated version of SNOMED CT.

Interpretation by humans is a challenge

Generated expressions may not use natural language

Task of manually assigning terms that accurately represent the meaning of expressions require training and is cognitively demanding

Interpretation by machines is a challenge

Advanced tooling is required to create, maintain, understand and use expressions

- When expressions are communicated to other systems:
    - Interpretation in the receiving system is not guaranteed

    - The burden of advanced tooling is put upon the receiving system

2.6 Use Cases

There is a range of use cases in which it **may** be helpful to use postcoordination, including:

Capture new clinical meanings in a health record, message structure, or query specification

- To support situations where clinicians need to record and share a clinical meaning, which has not been defined in any release of SNOMED CT. This applies to direct data entry, coding values in UIs, or other clinical situations. SNOMED CT postcoordinated expressions can be used to represent a new clinical meaning as an alternative to concept authoring.

Enable interoperability between different structural representations of the same clinical meaning

- To support transformation between different representations within a clinical system or while exchanging data between systems. The postcoordinated expression is a common representation of clinical meaning. It can be exchanged and compared between models and shared as part of interoperability standards like HL7 FHIR.

Represent maps from other code systems or from interface terms to an equivalent representation of the meaning in SNOMED CT

- To map other code systems to SNOMED CT may require the representation of new meanings or combinations of meanings already represented in SNOMED CT. In these cases, when a suitable precoordinated concept is not available, postcoordination offers an alternative to concept authoring.

Support Natural Language Processing

- To transform the meaning of free text into structured SNOMED CT expressions. Postcoordination allows the representation of specific context or a greater amount of detail in the free text than you can capture with a single precoordinated code

2.7 Alternatives to Postcoordination

Before you launch into a postcoordinated solution it’s important to understand that in some cases there are alternatives to postcoordination that may be easier to implement.

The two main alternatives to postcoordination are:

Create extension concepts:

- Firstly, and perhaps the most obvious alternative to postcoordination is to add the new clinical meanings to an extension, as precoordinated content. This approach enables the use of standard SNOMED authoring tools to classify the concepts and distribute them in a way that supports queries. However, this alternative is not an option for everyone, and in those cases, postcoordination may be considered.

Use information model structure:

- The second alternative is to define the postcoordination structure using the information model, using separate coded data elements for the focus concept, and each of the refinement values in the expression. For example, rather than using an expression for ‘fracture of the foot’, two separate data elements could be applied, one for the SNOMED CT concept for 'fracture' and the other for the SNOMED CT concept for 'foot'. This approach has good accuracy for data capture but introduces significant limitations for the execution of semantic queries over the clinical data, as the focus concept is separated from the refinements and is not combined into a single SNOMED expression. In some situations where data is captured separately and follows the SNOMED CT concept model, post-processing can be applied to convert information models into postcoordinated expressions, after data capture.

3. System Components

Enabling postcoordination in an Electronic Health Record (EHR) usually requires an EHR design that is prepared to support the use of expressions.

When developing EHRs, these are usually designed with both a health record data store to record health-related information about each patient, and a terminology server to store the codes and terms that may be used in the health records as illustrated in Figure 3-1.

A SNOMED CT-enabled terminology server stores the terminology content and provides the services required by the EHR to access the content. A terminology repository is included to hold the released terminology content, for example, a specific version of a national Edition of SNOMED CT plus any required SNOMED CT derivatives. When supporting postcoordination, the terminology server should support the storage of expressions, for example using an expression repository. In addition, dedicated services are required to enable the processing and use of the expressions in the same way as the released, precoordinated SNOMED CT content.

The recommended approach to ensure efficient use of a SNOMED CT-enabled terminology server is to use a service-oriented approach where dedicated terminology services are provided via an API enabling access and processing of the content within the terminology and expression repository.

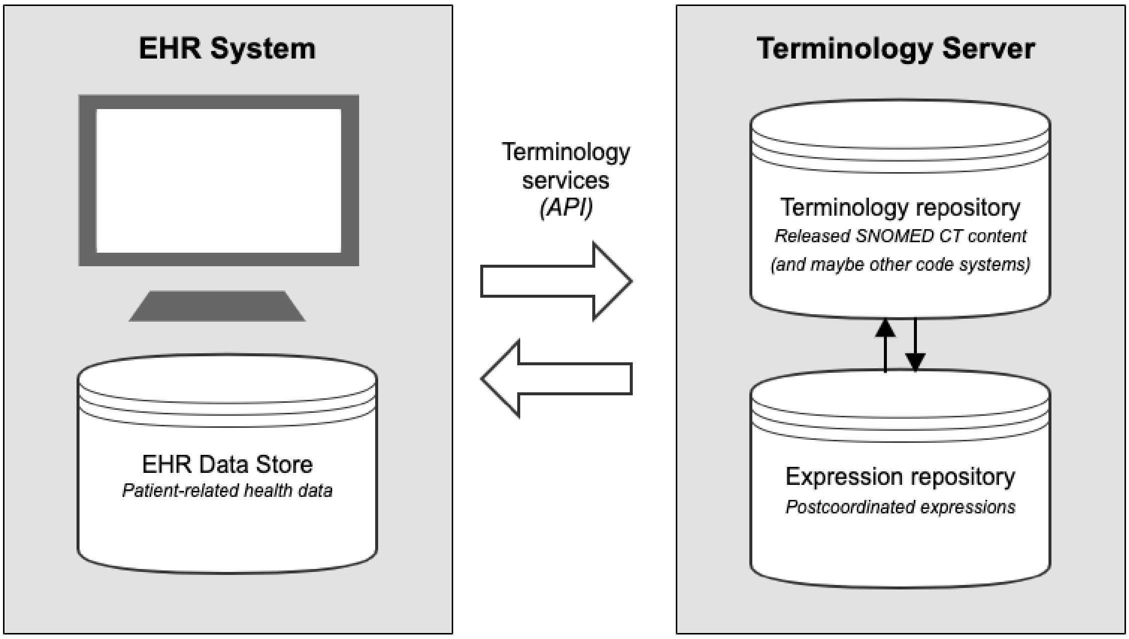

EHR System
EHR Data Store
Patient-related health data
Terminology services (API)
Terminology Server
Terminology repository
Released SNOMED CT content
(and maybe other code systems)
Expression repository
Postcoordinated expressions

Figure 3-1: Overall design of an EHR supporting postcoordination. The EHR data store is used to hold the health related information recorded about each patient, and the terminology server includes the terminology content required by the EHR. This includes both a terminology repository for the released SNOMED CT content and an expression repository for the postcoordinated expressions.

4. Expressions in the EHR

This part of the guide describes one way in which postcoordination might be applied in an EHR. This involves introducing the steps involved with setting up an expression repository, including approaches taken to create and add expressions to the repository.

This section also introduces how postcoordination can be implemented to support essential EHR tasks including entering, storing, displaying, querying, and communicating expressions. In addition, references to the terminology services required to support these approaches will be provided to function as a practical source of reference for implementers wishing to enable postcoordination in their EHR.

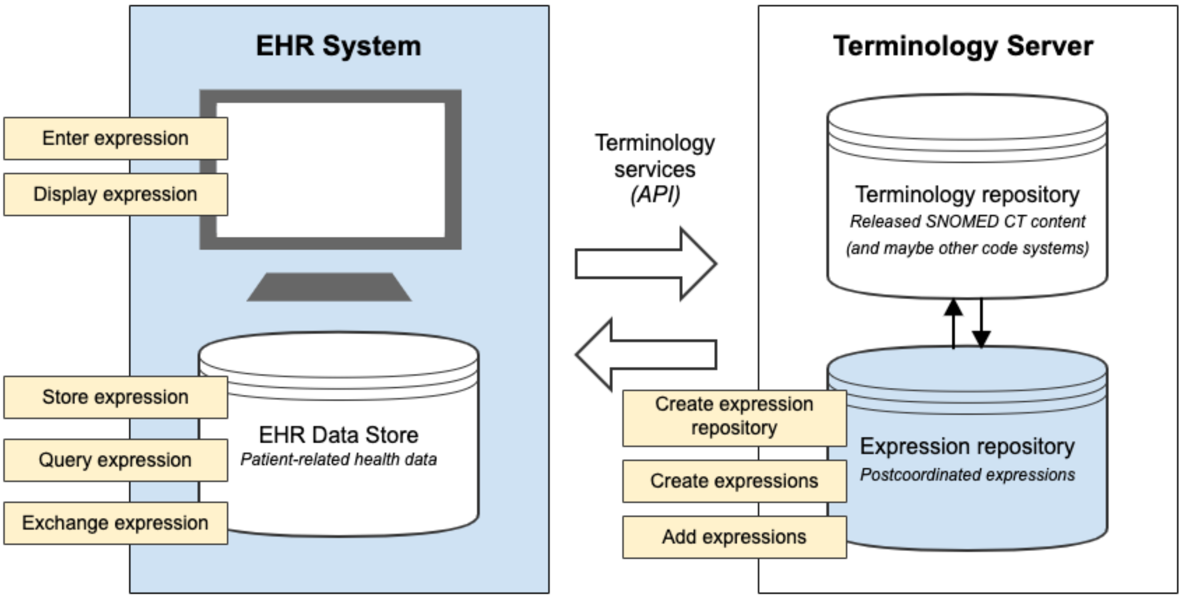


Figure 4-1: This section summarizes the how postcoordinated expressions may be implemented to support key EHR tasks, and it introduces key tasks required to prepare an expression repository.

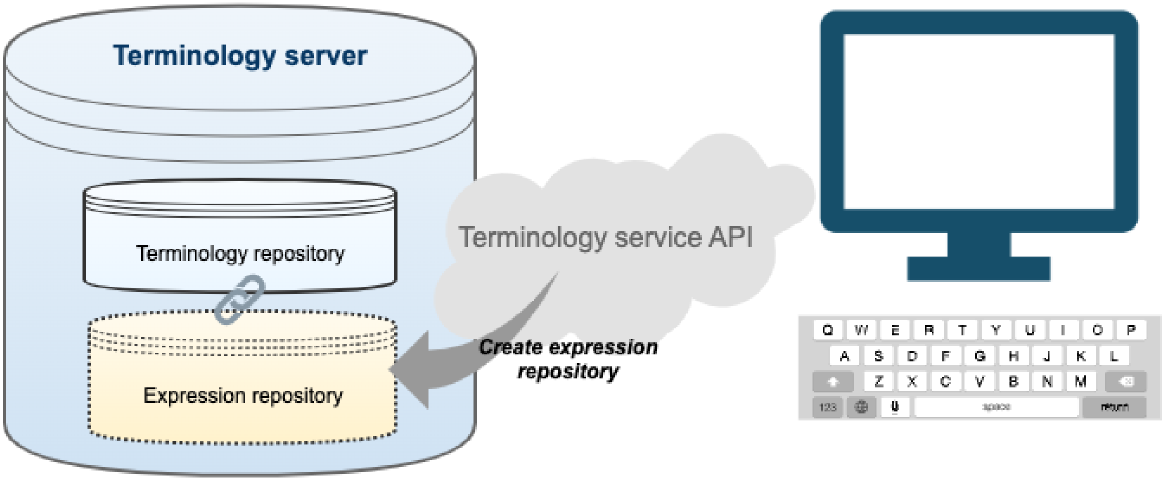
4.1 Preparing Expression Repository

Enabling postcoordination in an EHR may involve the preparation of an expression repository. This involves the actual creation of the expression repository and often the addition of a set of postcoordinated expressions to be included in the value sets of the EHR.

The following pages will elaborate on the steps involved with the setup of the expression repository.

4.1.1 Creating an Expression Repository

To enable expressions to be stored and accessed in the terminology server, the expression repository first needs to be created.



A SNOMED CT-enabled expression repository is not a standalone artifact. The expression repository is tightly coupled with the terminology repository in the sense that all expressions represented in the expression repository are composed of concepts included in the applied terminology repository, i.e. a specific version of a SNOMED CT Edition. The terminology content on which the expression repository depends is also referred to as the substrate. When a terminology repository is updated to include a new version of SNOMED CT, the expression repository will equally need to be updated to align with the applied terminology content.

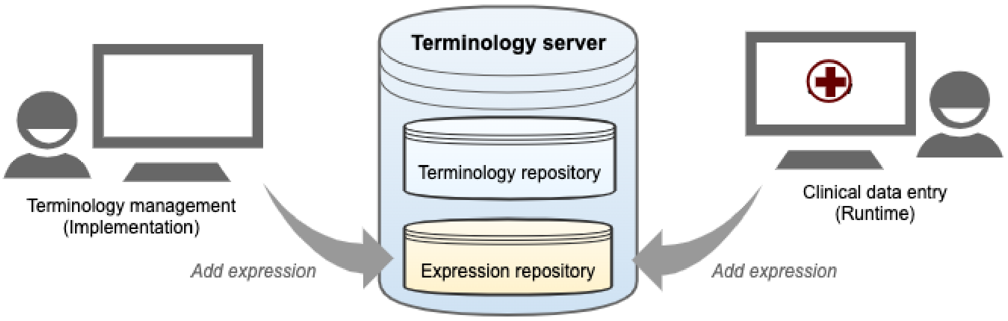
The services required by a terminology server to create an expression repository are described in 5.2.1 Create Expression Repository

The overall requirements for updating an expression repository are described in 5.2.3 Update Expression Repository

4.1.2 Adding Expressions to Repository

Depending on the choices made as part of a SNOMED CT implementation project, the addition of expressions to an expression repository may be done at various stages of implementation and by different users. For example:

- Some implementations may restrict the creation of expressions to only during the implementation stage, where
  - Expressions are created by people involved with the implementation
  - Expressions are added to the expression repository to be made available for end users in the same way as precoordinated SNOMED CT content. By this approach, the end users can not themselves directly create new postcoordinated expressions
- Some implementations may enable runtime creation of postcoordinated expressions by end users, where
  - Expressions are created by end-users as part of the data entry process
  - Expressions are added to the expression repository as part of saving patient data in the EHR record store.



The services required by the terminology server to add expressions to an expression repository are described in 5.2.4 Add Expression.

4.1.3 Creating Expressions

The creation of postcoordinated expressions can be performed using a variety of techniques, which are described in the following.

4.1.3.1 Mapping to Expression

Authoring of postcoordinated expressions may be conducted as part of an implementation project where a set of local interface terms or codes are mapped to SNOMED CT. Although it may be desired to map each local term to a precoordinated concept, mapping to expressions may be motivated by the following factors:

- An equivalent map is required but no precoordinated concept represents the semantics of the local term
- It is desired to map specific types of local terms to SNOMED CT using a consistent mapping structure
    - For example, when mapping terms representing disorders of lateralized body structures, but only some of these disorders are precoordinated in SNOMED CT, then it may be chosen to map all of these cases to an expression of the same structure

Semantic Equivalence

When mapping to an expression it is important to aim for semantic equivalence, meaning using a focus concept and refinement that represents the full meaning of the local term. If this is not achievable, consider updating the local term to reflect the meaning of the expression.

Table 4.1.3.1-1 provides an example of two different local terms that are mapped to the same expression, and thus resulting in different semantic correlations.

Source term	CTU Expression	Semantic correlation
Bleeding from larynx	131148009 |Bleeding| : 363698007 |Finding site| = 4596009 |Laryngeal structure|	Exact match
Bleeding from larynx requiring transfusion	131148009 |Bleeding| : 363698007 |Finding site| = 4596009 |Laryngeal structure| *Note in this case the phrase "requiring transfusion" can't be modeled with the available attributes and, therefore, semantic equivalence is not achievable.*	Narrow to broad

Table 4.1.3.1-1: Semantic correlations between a source term and a target expression.

Alternatives to Mapping to a Postcoordinated Expression

Before deciding to map local codes or terms to SNOMED CT expressions, please consider the following alternatives:

- If you can't achieve an equivalent map from the local term to a precoordinated SNOMED CT concept, you may instead choose to map to a broader concept. Although this approach may lead to information loss because some of the meaning represented in the original local code will be degraded or lost in the map, the target concept may still retain enough of the critical original semantics required to support the subsequent use of the data
- If the local term is nationally or internationally relevant, a request for a new concept to be added to the national extension or the International Edition of SNOMED CT representing the meaning of the local term
- If the local term represents a meaning only just within the given country or organization, a concept may be added to the national/local extension representing the meaning of the local term

4.1.3.2 Expression Creation Techniques

The following pages describe the techniques that may be applied to create postcoordinated expressions.

Regardless of the technique applied the key requirement is to ensure that expressions are created in accordance with the concept model rules or specified patterns for postcoordination.

4.1.3.2.1 Compositional Grammar

Authoring expressions directly using the compositional grammar, even by hand in a text editor, is useful for situations in which expressions must be defined which don't necessarily conform to a consistent structure. To use this technique, the user must be familiar with the basic features of the Compositional Grammar syntax, please see Compositional Grammar - Specification and Guide.

There are, however, a number of ways in which a dedicated Compositional Grammar editing tool rather than a generic text editor can better support the user while creating expressions, including:

- Validate the syntactical correctness of the expression as it is authored
- Check the expression for conformance against the concept model
- Automatically populate or correcting the terms used for each part of the expression
- Provide integrated tools to search the SNOMED CT hierarchy for concepts to include in the expression
- Suggest the set of valid operators or characters that may be used at a given point in the expression
- Show any inferable relationships, parents, and/or child concepts
- Indicate if there is an equivalent pre-coordinated concept

When creating expressions using the compositional grammar, it is the responsibility of the author of the expression to ensure the clinical validity of the expression, and ensure that the expression syntax, the concept model and the editorial rules are followed.

4.1.3.2.2 Form-based Creation

Form-based creation enables users to construct a postcoordinated expression from the selections permitted by a dedicated form. This approach is useful for preventing errors when generating postcoordinated expressions, and is, therefore, applicable to a range of users, including:

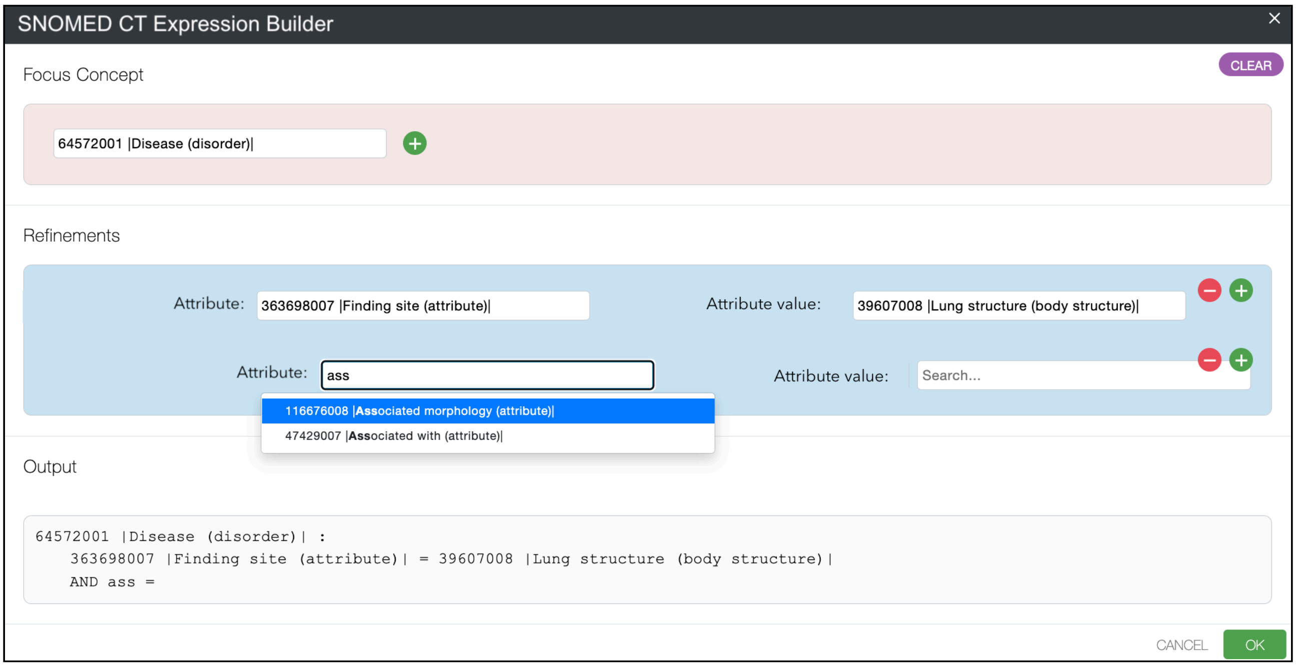
- Non-technical users, or users with limited knowledge of the SNOMED CT concept model
- Expert users with very detailed knowledge of SNOMED but where the expressions to be built or the rules that constrain them are complex and so errors are easy to make

Form-based authoring may involve the use of a generic SNOMED CT expression builder, (see Figure 4.1.3.2.2-1) , or the form may populate a predefined expression template (see 4.1.3.2.3 Expression Templates).

A well-designed expression builder utilizes the features of the compositional grammar, the Machine Readable Concept Model, and the SNOMED CT Editorial Guide.

Key features of an expression builder include:

- Enable the selection of the focus concepts to be wither refined or qualified
    - Either via the selection from a pre-defined set of options or from a less constrained search across the hierarchies
- Enable the addition of one or more refinements
    - The attributes available should be constrained to only those attributes that apply to the selected domain
    - The attribute values available for selection should be limited to the concepts within the permitted range determined by the selected attribute and focus concept
- Enable the addition of attribute groups, when these are required to represent the desired semantics



Focus Concept

64572001 |Disease (disorder)|

Refinements

Attribute: 363698007 |Finding site (attribute)| Attribute value: 39607008 |Lung structure (body structure)|

Attribute: ass Attribute value: Search...

116676008 |Associated morphology (attribute)|
47429007 |Associated with (attribute)|

Output

```
64572001 |Disease (disorder)| :
    363698007 |Finding site (attribute)| = 39607008 |Lung structure (body structure)|
    AND ass =
```

Figure 4.1.3.2.2-1: Example of an expression builder which supports implementers in constructing valid SNOMED CT postcoordinated expressions.

When using a form to create postcoordinated expressions it is the responsibility of the developers of the form to ensure that the generated expressions are valid considering the syntax and concept model rules. Users of the form should, however, be aware of the editorial guidance that apply to the domain of the expression.

4.1.3.2.3 Expression Templates

The authoring of postcoordinated expressions can also be performed using expression templates. These templates provide reusable patterns for the representation of expressions that express similar meanings. Expression templates can define the set of attributes and attribute values that must be used to obtain the expected classification results, and assist the users in authoring expressions in a consistent way. To ease the authoring of postcoordinated expressions using expression templates, the templates may be populated in a data entry form where the slots provide searchable fields constrained as specified by the template.

For example, there may be a preference to always represent fractures using an expression in which the body site is explicitly defined as the value of the |finding site| attribute. Given an appropriate expression template and a predefined list of body structures, a set of template-compliant postcoordinated expressions can be authored.

SNOMED International specifies the SNOMED Template Syntax, which provides the formal rules for representing slots in SNOMED CT expressions. A slot can represent a placeholder for a value that is not known at the time of authoring. These placeholders can be completed at a later time using data recorded elsewhere (such as in an information model or entered into a data entry form). Expression templates can thus be used to enable the easy and consistent creation of postcoordinated expressions from data entered into a user interface.

Example

Consider the following expression template for creating expressions representing family history disorders:

Expression Template for Family History Disorders

```
243796009 |Situation with explicit context| :
  { 246090004 |Associated finding| = [[+id (< 404684003 |Clinical finding| ) @finding]],
    408731000 |Temporal context| = 410511007 |Current or past| ,
    408729009 |Finding context| = 410515003 |Known present| ,
    408732007 |Subject relationship context| = 303071001 |Person in the family| }
```

This template includes only one slot to capture the value of the |Associated finding| attribute, i.e. the **@finding** slot. All other parts of the expression are fixed to the following attributes required to represent the meaning of the described meaning:

- The temporal context, to state that the finding occur now, or in the past:

```
40873100 |Temporal context| = 410511007 |Current or past|
```

- The finding context, to state that the finding exists, opposed to being absent:

```
408729009 |Finding context| = 410515003 |Known present|
```

- The subject relationship context, to state that the situation relates to a person in the family:

```
408732007 |Subject relationship context| = 303071001 |Person in the family|
```

Although this information may not be explicitly selected by the end user, this meaning may implied by the context in which it is selected, for example:

- If the user selects an option in a pick list with a term saying "Family history of **@finding**", e.g. "Family history of breast cancer"
- If the user enters a disorder from a list or search field belonging to a 'Family history' section of a user interfac

If a user selects the value 56265001 | Heart disease| to the finding slot (e.g. finding = 56265001 | Heart disease|), then the following expression can be derived based on that single selection:

Resulting Expression

```
243796009 |Situation with explicit context| :
  { 246090004 |Associated finding| = 56265001 |Heart disease| ,
    40873100 |Temporal context| = 410511007 |Current or past| ,
    408729009 |Finding context| = 410515003 |Known present| ,
    408732007 |Subject relationship context| = 303071001 |Person in the family| }
```

ⓘ Detailed information about the template syntax can be found here: SNOMED Template Syntax

4.1.3.2.4 Automatic Generation

Some applications allow free text to be entered into patient records, which can either be manually or automatically processed to transform into SNOMED CT Compositional Grammar expressions. This approach relies on either trained staff that are available to do the processing in a timely manner, or NLP (Natural Language Processing) software which is able to generate reliable postcoordinated expressions for clinician review.

4.2 Entering Expressions in an EHR

When enabling postcoordination in an EHR, it is important to be aware of how the approach selected for entry of the expression impacts the storage and the display of the expression.

There are three main approaches to entering postcoordinated clinical data into an electronic health record.

- Prepopulated expressions
- Form-based selection of expression parts
- Free text with Natural Language processing to encode the free text with SNOMED CT

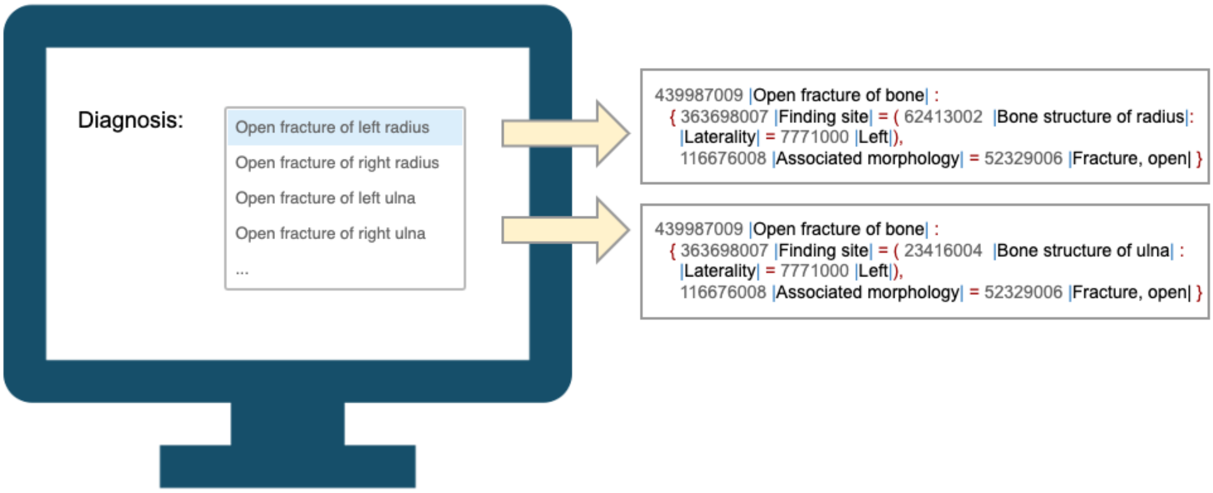
The following pages will elaborate on each of these approaches and clarify key requirements for each. Furthermore, best practice approaches to the storage and display of expressions will also be presented.

4.2.1 Prepopulated Expressions

When using pre-populated expressions, the user is presented with a list of clinical phrases (‘interface terms’), using a pick list, checkboxes, or another type of data entry control.

The essence of this approach is that the user is presented with a clinical phrase representing a specific clinical meaning, and in the backend of the system, this clinical phrase is associated with a postcoordinated expression (instead of a precoordinated concept).

In the example shown in Figure 4.2.1-1, the user interface includes a list of phrases such as “Open fracture of left radius”, and “Open fracture of right radius”. Some of these phrases may be associated with precoordinated content (either via mapping or the released SNOMED CT descriptions), while other phrases may be associated with postcoordinated expressions. In this example, the phrase “Open fracture of left radius” and “Open fracture of left ulna” is associated with postcoordinated expressions.


Diagnosis:
Open fracture of left radius
Open fracture of right radius
Open fracture of left ulna
Open fracture of right ulna
...

```
439987009 |Open fracture of bone| :
  { 363698007 |Finding site| = ( 62413002  |Bone structure of radius|:
    |Laterality| = 7771000 |Left|),
    116676008 |Associated morphology| = 52329006 |Fracture, open| }
```

```
439987009 |Open fracture of bone| :
  { 363698007 |Finding site| = ( 23416004  |Bone structure of ulna| :
    |Laterality| = 7771000 |Left|),
    116676008 |Associated morphology| = 52329006 |Fracture, open| }
```

Figure 4.2.1-1: Options displayed in a user interface may be associated with a postcoordinated expression

This approach requires the expressions to be created and mapped to an interface term by a SNOMED CT implementer. This occurs as part of the terminology binding process in the design stage of the implementation. Different approaches to the creation of the expressions exist and depend on the tools available. These approaches are described in the following pages.

Using postcoordinated expressions for items in a prepopulated list, or predefined values in a UI, each expression is associated with one field on the user interface. For precoordinated concepts, the best practice is to record the concept identifier as the code, and the term that was displayed or selected by the user as the original text.

The recommended approach to the storage of predefined postcoodinated expressions is similar to the storage of precoordinated concepts. This means that the display term used for the expression should be recorded in the health record, but the ‘code’ should either be the terse form of the full expression (i.e. the expression with the terms and spaces removed), or the unique identifier associated with the expression when this was added to the expression repository.

Terminology services

Enabling prepopulated expressions in the EHR requires the EHR to access expressions already created in the expression repository.
For this purpose, the following services may be required:

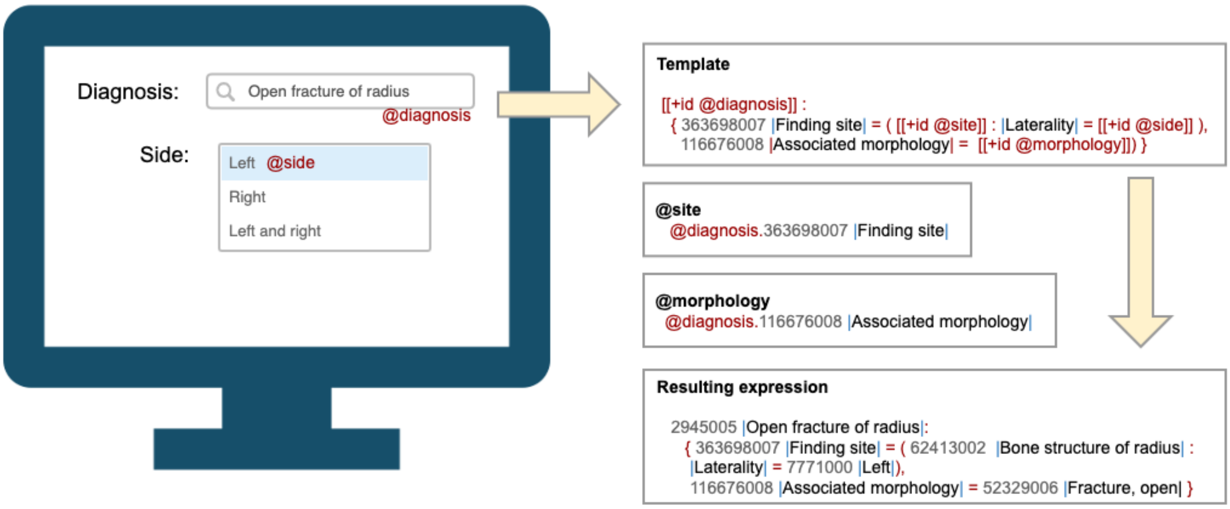
- 5.2.7 Lookup Expression
- 5.2.8 Search Expression
- 5.2.9 Get Display Term

4.2.2 Form-based Entry of Expressions

With the form-based approach, the combination of values recorded in different data entry controls results in specific expressions to be formed.

With this approach, using an expression template can help to ensure that the selected values are added in the appropriate places within the expression.

The example shown in Figure 4.2.2-1 illustrates how an expression template is used to apply a laterality to a selected diagnosis. In this example, if a user enters the values *Open fracture of radius* as the diagnosis, and the value *left* in the side field, the values entered are then used to populate the slots in the expression template and form an expression representing the meaning of an 'open fracture of radius with a laterality of left'.



Template

```
[[+id @diagnosis]] :
  { 363698007 |Finding site| = ( [[+id @site]] : |Laterality| = [[+id @side]] ),
    116676008 |Associated morphology| =  [[+id @morphology]]) }
```

@site

```
@diagnosis.363698007 |Finding site|
```

@morphology

```
@diagnosis.116676008 |Associated morphology|
```

Resulting expression

```
2945005 |Open fracture of radius|:
  { 363698007 |Finding site| = ( 62413002  |Bone structure of radius| :
    |Laterality| = 7771000 |Left|),
    116676008 |Associated morphology| = 52329006 |Fracture, open| }
```

Figure 4.2.2-1: In this example, the value of the 'diagnosis' field and the value of the 'side' field are used to represent different parts of the resulting expression.

Expression templates associated with a data entry form should be created during the design phase of the implementation, and then populated with values at run time, when the user selects values in each field. Please see section 4.1.3.2.3 Expression Templates for details on expression templates.

When using the form-based approach for entering postcoordinated expressions in an EHR, it is recommended to store the data in an information structure that matches the fields on the form and to record the concept identifier and display term for each of the fields, within that information structure. This enables the system to efficiently display health records that have been recorded previously, and also provides full traceability of what the user selected at the point of data entry. It also enables the possibility of re-generating a postcoordinated expression at a later date that is potentially different from that which was originally generated, e.g. if an error were found in the original template or if structural changes in SNOMED force a redesign of the expression template.

In addition to the storage of the individual concept identifiers, the derived expression (in terse form) may also be stored to meet specific EHR-requirements. Generating the derived expression may be done using an expression template.

Terminology services

Enabling form-based entry of expressions in the EHR requires the EHR to dynamically access at runtime SNOMED CT concepts as determined by the terminology binding specified for each data element. At the point of data entry, no specific services for postcoordination are required, however, if the resulting expression is stored as a single expression, or subsequently communicated, terminology services are required to add the expression to the expression repository.
For this purpose, the following services are required:

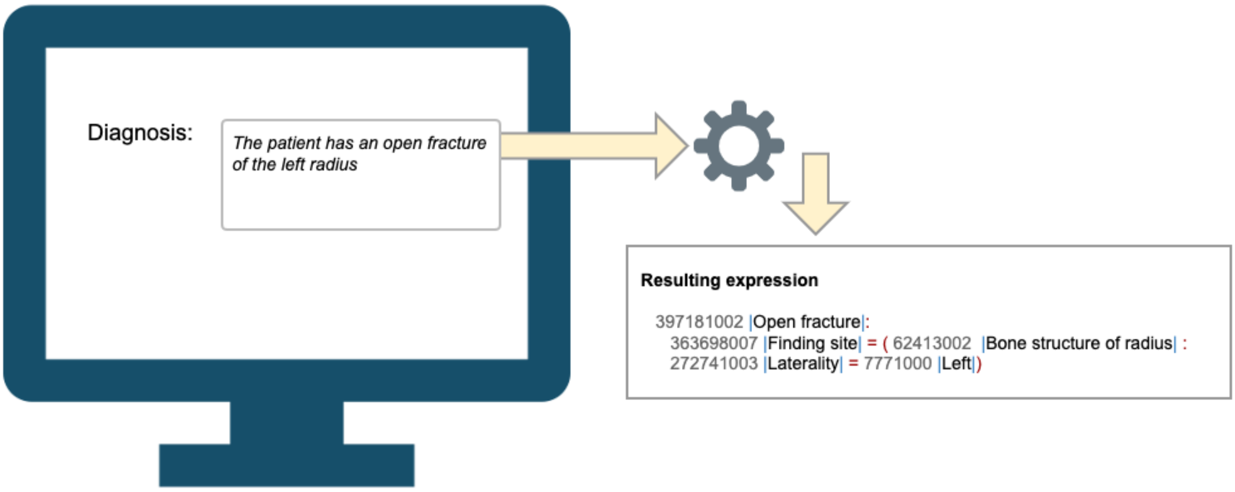
- 5.2.4 Add Expression (including required sub-services)

4.2.3 Free-text with Natural Language Processing

Except where agreed business processes require structured and standardised recording, using free text fields are often a necessity to support the clinical documentation and established work processes. Free text in Electronic Health Records enables the user to enter information using the terms and phrases of his/her own choice.

With this approach, the clinical user enters the information in a dedicated text box, and the text is subsequently analyzed by a natural language processing (NLP) tool. The NLP tool is designed to codify the key terms as SNOMED CT concepts, and the rules specified in the Machine Readable Concept Model (MRCM) may be applied to help identify the relationships between these key terms, to form SNOMED CT expressions.

Figure 4.2.3-1 illustrates how the phrase *"The patient has an open fracture of the left radius"* is processed by an NLP tool to form the expression 42945005 |Open fracture of radius| : 272741003 |Laterality| = 7771000 |Left| .



```
397181002 |Open fracture|:
  363698007 |Finding site| = ( 62413002  |Bone structure of radius| :
  272741003 |Laterality| = 7771000 |Left|)
```

Figure 4.2.3-1: Using free text to enter postcoordinated expressions involves the free text to be transformed into a SNOMED CT expression using a SNOMED CT-enabled Natural Language Processing tool.

With this approach, the expressions are usually created at run-time, as the user types in the free text. This requires the system to incorporate a SNOMED CT- enabled NLP service that works in the local dialect.

Using the free-text approach with an NLP tool to generate the expressions, it is recommended, and often required for medicolegal reasons, to store both the free text as it was entered by the user, and also the coded expression that was generated by the NLP tool and confirmed by the user. You should also store the terms associated with the expression that was displayed to the user when they confirmed that it accurately reflects the meaning of the free text.

Terminology services

Enabling NLP with expressions in the EHR requires the EHR to add newly created expressions derived from the NLP algorithm.
For this purpose, the following services may be required:

- 5.2.7 Lookup Expression
- 5.2.4 Add Expression
- 5.2.9 Get Display Term. This may be required to display a human readable representation of the generated expression, to enable the end-user to confirm the result of the NLP analysis.

So, in this approach the expressions are usually created at run-time, as the user types in the free text. This requires the system to incorporate a SNOMED CT enabled NLP service that works in the local dialect.So, in this approach the expressions are usually created at run-time, as the user types in the free text. This requires the system to incorporate a SNOMED CT enabled NLP service that works in the local dialect.So, in this approach the expressions are usually created at run-time, as the user types in the free text. This requires the system to incorporate a SNOMED CT enabled NLP service that works in the local dialect.

4.3 Storing Patient Data using Expressions

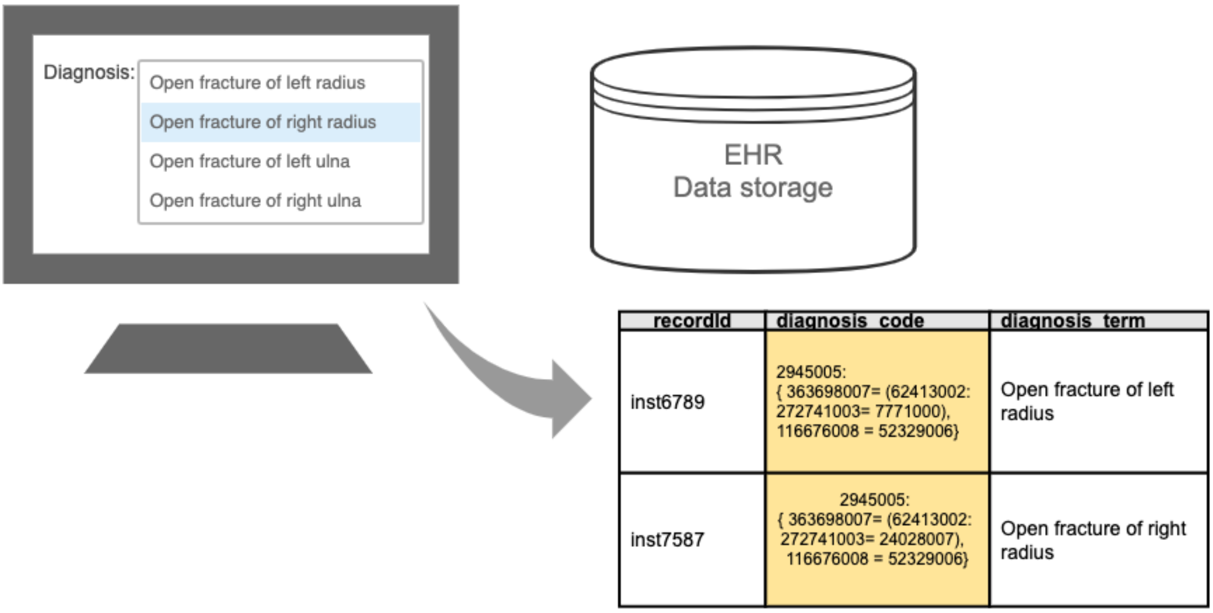
When implementing postcoordinated SNOMED CT expressions, there are different approaches to storage that may be considered.

- Store the full expression in the same way as you would store any coded data item (section 4.3.1 Store the Full Expression)
- Reference the expression in the record storage using a local expression identifier, and hold the expressions in a separate expression repository (section 2.2.2 Store Expression Identifier)
- Store the expression parts individually in separate fields (section 2.2.3 Store Expression Parts Individually)

Each of these approaches are described in the following pages:

4.3.1 Store the Full Expression

With this approach, the full expression is stored in a single field, in the same way as the storage of single concept identifiers.


Diagnosis:
Open fracture of left radius
Open fracture of right radius
Open fracture of left ulna
Open fracture of right ulna
EHR
Data storage

recordId	diagnosis_code	diagnosis_term
inst6789	2945005: { 363698007= (62413002: 272741003= 7771000), 116676008 = 52329006}	Open fracture of left radius
inst7587	2945005: { 363698007= (62413002: 272741003= 24028007), 116676008 = 52329006}	Open fracture of right radius

The canonical version of the compositional grammar expression (see 2.4 Expression Forms) is recommended for the storage of the full expression, because this has a minimum of syntactic noise and specified sort order for the elements within the expression.

To use this approach, the health record must be able to record codes that are as long as the longest expression linked to one of the interface terms. The codes must also allow non-numeric values to be included, such as the colon and equal signs that are used in most expressions.

Some clinical systems, however, may have limits on the length or datatype of the codes that may be recorded in their health records. So, in these cases, a unique local identifier can be assigned to each expression, so that this shorter expression id can be stored in the health record in the place of the full expression (see 4.3.2 Store Expression Identifier).

Terminology services

Storing the full expression in the EHR record requires the EHR to ensure that the stored expression represents a valid SNOMED CT expression.
For this purpose, the following services may be required:

- 5.2.5 Validate and Transform Expression (note, that for this purpose the CTU expression form of the expression should be used for data storage)

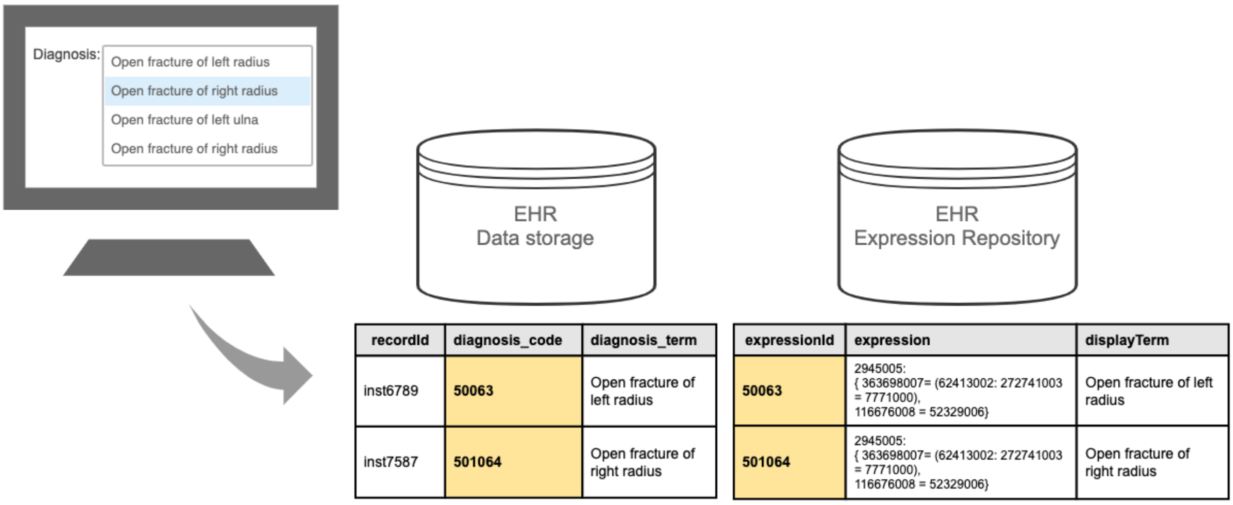
In conjunction to the EHR data storage, this process may also require the expression being added to an expression repository, which require the following services

- 5.2.7 Lookup Expression
- 5.2.4 Add Expression

4.3.2 Store Expression Identifier

In some cases, it may be preferred or required to use a single identifier to represent the stored expression. With this approach, a unique local identifier is assigned to each expression, so that this can be stored in the health record in the place of the full expression. This approach may be selected if the clinical system has a limit on the length of the codes that are recorded in their health records.

When local expression identifiers are used, these are then linked to the corresponding full expression and display term within an Expression Repository, as illustrated in Figure 4.3.2-1.


Diagnosis:
Open fracture of left radius
Open fracture of right radius
Open fracture of left ulna
Open fracture of right radius
EHR
Data storage
EHR
Expression Repository

recordId	diagnosis_code	diagnosis_term
inst6789	**50063**	Open fracture of left radius
inst7587	**501064**	Open fracture of right radius

expressionId	expression	displayTerm
50063	2945005: { 363698007= (62413002: 272741003 = 7771000), 116676008 = 52329006}	Open fracture of left radius
501064	2945005: { 363698007= (62413002: 272741003 = 7771000), 116676008 = 52329006}	Open fracture of right radius

Figure 4.3.2-1: This image illustrates the principles of storing postcoordinated expressions in an expression repository. The EHR data store uses local expression identifiers which are linked to a SNOMED CT expression and display term within the expression repository

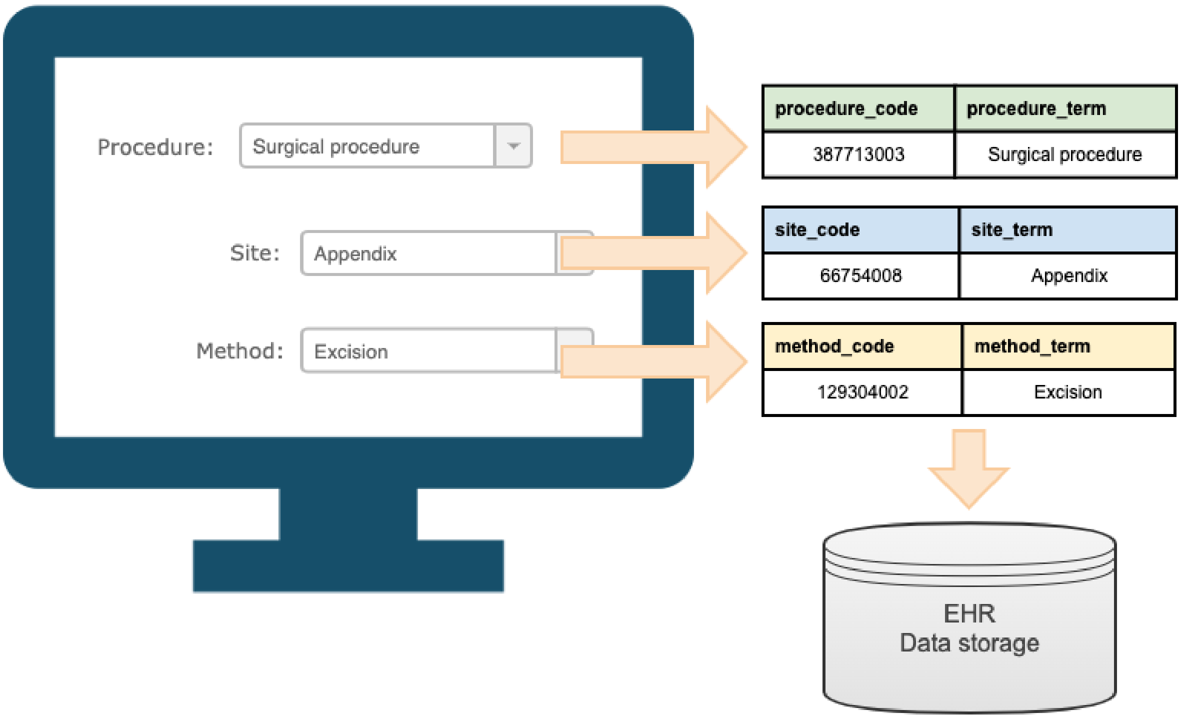
Terminology services

Storing an expression identifier requires the expression to first be added to the expression repository, and assigned a unique identifier. The resulting identifier will be stored in the EHR data store.
For this purpose, the following services may be required:

- 5.2.7 Lookup Expression - to get the identifier of the expression to be stored.
- 5.2.4 Add Expression. This is only required in cases where the expression does not already exist in the expression repository.

4.3.3 Store Expression Parts Individually

When clinical data is entered in an EHR using a number of interconnected data elements, the composition of the individual data items may derive a postcoordinated expression. This is exemplified in the image below where three data entry controls are used to form the clinical meaning of "an excision of the appendix structure", see Figure 4.3.3-1.



Procedure: Surgical procedure

Site: Appendix

Method: Excision

procedure_code	procedure_term
387713003	Surgical procedure

site_code	site_term
66754008	Appendix

method_code	method_term
129304002	Excision

Figure 4.3.3-1: Form-based approach to data entry, where the expression parts are stored in separate fields.

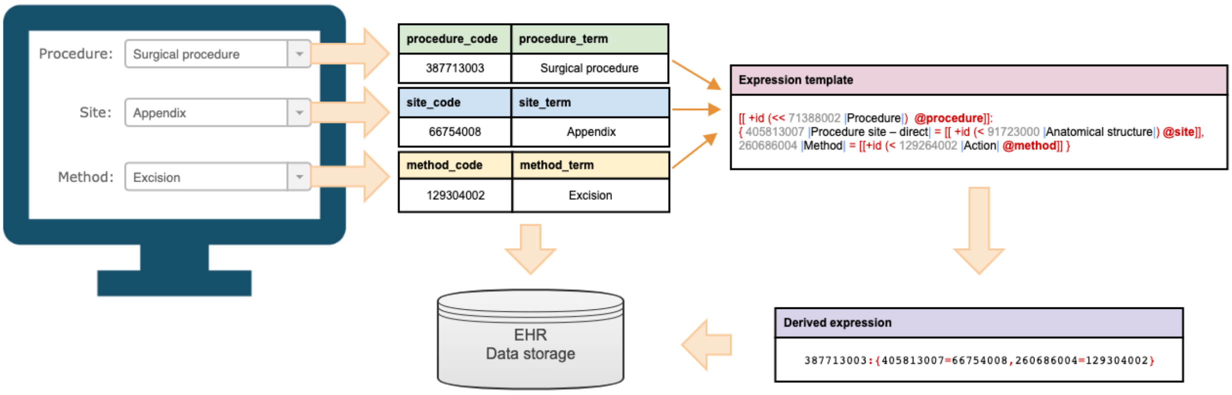
With this approach, it is recommended to store the data in an information structure that matches the fields on the form and to record the concept identifier and display term for each of the fields, within that information structure. This enables the system to efficiently display health records that have been recorded previously, and also provides full traceability of what the user selected at the point of data entry.

Store Derived Expression

Aside from storing the individual data items, the associated postcoordinated expression may also be generated using an expression template and stored, if this is needed for other EHR tasks. The principle of this approach is illustrated in the diagram below where the derived expression represents the combined meaning of the selected values. In this example, the concept id for |surgical procedure| replaces the @procedure slot in the expression template. The concept id of |Appendix| replaces the @site slot, and the concept id for |Excision| replaces the @method slot.

The benefit of this approach is that changes to the modelling of a particular branch in SNOMED CT will only require an update to the applied expression, instead of every single expression.

Some implementations may choose to store this derived expression, in addition to the individual values of each field, to support run-time querying or exchange.



procedure_code	procedure_term
387713003	Surgical procedure

site_code	site_term
66754008	Appendix

method_code	method_term
129304002	Excision

```
[[ +id (<< 71388002 |Procedure|)  @procedure]]:
{ 405813007 |Procedure site – direct| = [[ +id (< 91723000 |Anatomical structure|) @site]],
260686004 |Method| = [[+id (< 129264002 |Action| @method]] }
```

```
387713003:{405813007=66754008,260686004=129304002}
```

Table 4.3.3-1: Illustration of the principle of using an expression template to generate a postcoordinated expression from the data recorded using the form-based approach to data entry.

4.4 Displaying Expressions in an EHR

The approach to the display of postcoordinated expressions will depend on how they are created, stored, and exchanged.

But the general recommendation is to display the expressions in the same way as they were entered or confirmed by the user.
For example, if the expression has been constructed from the selections made in a form, the display of the expression would involve rendering the selected terms with an indication of the context in which the value was selected. Some situations may, however, require a single term to be constructed for an expression, e.g. to support data exchange.

The following pages introduce general approaches to the creation of display terms for postcoordinated expressions.

4.4.1 Manual Creation of Display Terms

When a postcoordinated expression is created during the implementation process, this may involve the manual creation of a display term for each expression. Two scenarios for the manual creation of display terms can be outlined:

1. The list of clinical phrases is developed first, based on user requirements or an existing library of clinical phrases, and then these will be mapped to SNOMED expressions
2. A set of postcoordinated expressions are created first to represent the clinical meanings that are required, and a display term is created for each expression
    a. *This approach may be assisted by Natural Language Generation technologies and techniques to automatically suggest appropriate terms for each expression*

The challenge associated with the creation of terms for expressions is that the meaning represented in the term may differ from the meaning of the expression itself. This deviation may arise due to limitation of the concept model or implementation constraints. Implementers should, therefore, consider if they want to support the creation of terms for expressions and in which case they should implement a process to assure the quality of the created terms.

The goal of assigning a term to an expression is to select a term that is an accurate synonym for the meaning conveyed by the expression, ensuring semantic equivalence between the two. If the display term does not fully capture the meaning expressed in the expression, the expression should be labeled as a primitive ('<<<') to indicate that some aspects of the term's semantics are not reflected in the expression.

Source term	Target expression	Definition status
Bleeding from larynx	=== 131148009 |Bleeding| : 363698007 |Finding site| = 4596009 |Laryngeal structure|	Equivalent ('===')
Bleeding from larynx requiring transfusion	<<< 131148009 |Bleeding| : 363698007 |Finding site| = 4596009 |Laryngeal structure| *Note in this case the phrase "requiring transfusion" can't be modeled with the available attributes and, therefore, semantic equivalence is not achievable.*	Primitive ('<<<')

To avoid inconsistency between the display term and the expression, the recommended approach for providing a human-readable term for an expression is to automatically generate a display term. This approach, even though it may require human validation, ensures that the terms used in the display term are derived from terms approved as part of the authoring of the concepts referred to by the expression.

4.4.2 Automatic Creation of Display Terms

Use Cases

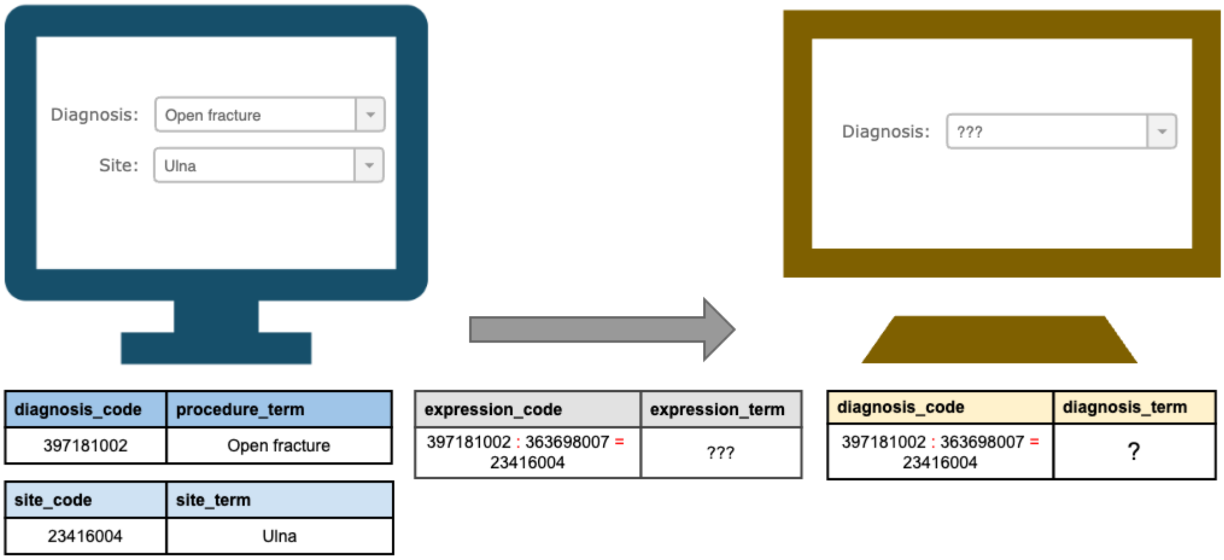
Some situations may require expressions to be displayed differently from how they were created. For example

- **Communication and display of expressions**
  When the information structures used to create the expression are different to those available for display or exchange
- **Design-time authoring of expressions**
  When postcoordinated expressions are created during the design of an EHR implementation

Communication and Display of Expressions

Autogeneration of display terms for postcoordinated expressions may be required when an expression is created in a different clinical system, or a different form on the same system, or it needs to be exchanged using a different information structure. In these cases, the information structure used to create the expression may be different to the ones available to display or exchange the expression and, therefore, the autogeneration of terms may be required.

This situation is illustrated in Figure 4.4.2-1 where the meaning of "open fracture of ulna" is captured in an information structure using two data elements and communicated to a target structure with only a single data element for the same meaning. In some settings, it may be applicable to support the display of the expression in the target system by automatically generating a term.



Diagnosis: Open fracture

Site: Ulna

Diagnosis: ???

diagnosis_code	procedure_term
397181002	Open fracture

site_code	site_term
23416004	Ulna

expression_code	expression_term
397181002 : 363698007 = 23416004	???

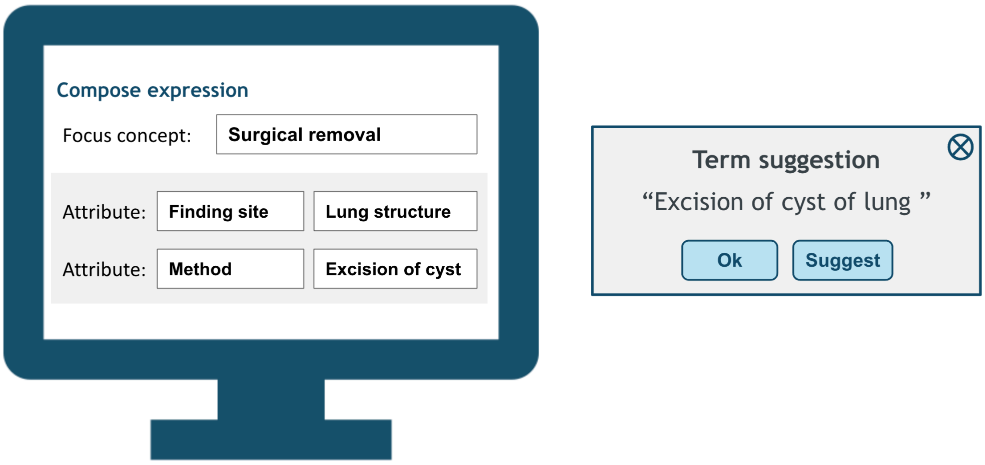
diagnosis_code	diagnosis_term
397181002 : 363698007 = 23416004	?

Figure 4.4.2-1: Some situations require a term to be generated for an expression, e.g. when the expression parts are captured as individual items but communicated to a target system populating the meaning in a single item.

Design-time Authoring of Display Terms for Expressions

Another situation where autogenerated terms may be useful is in systems used to create postcoordinated expressions during the design of an EHR implementation. At design time, an expression could be created using a form-based expression template, and then the system could suggest a display term that captures the meaning of the whole expression.

This scenario is illustrated in Figure 4.4.2-2.



Compose expression

Focus concept: Surgical removal

Attribute: Finding site | Lung structure

Attribute: Method | Excision of cyst

Term suggestion

“Excision of cyst of lung ”

Figure 4.4.2-2: Systems supporting the creation of postcoordinated expressions may automatically suggest a display term for the expression.

Terminology services

The techniques to generate a display term for the expressions may differ between implementations, but the following service is required by a terminology server supporting the automatic generation of a display term for an expression:

- 5.2.9 Get Display Term

4.5 Query Expressions

When postcoordinated expressions are used to represent clinical information, this data needs to be queried and analyzed in the same way as data recorded using precoordinated concepts.

Systems supporting postcoordination should thus ensure that analytics features are enabled to take advantage of SNOMED CT's hierarchy and explicit representation of defining properties. Furthermore, this facility should allow queries to be generated that combine SNOMED CT-specific selection criteria with other health record criteria.

Key analytics features that should be supported include:

- **Subsumption** which involves the determination of whether one concept (or expression) is a kind of another concept (or expression) is the fundamental capability enabled by SNOMED CT. For example, answering the question 'Which patients have an infectious disease?' involves finding all the patients with *any kind of* infectious disease (e.g. viral pneumonia, tuberculosis).
- **Selective retrieval** which involves using the properties (relationships) derived from classification to retrieve concepts (or expressions) matching specified criteria.

Subsumption

Subsumption testing between expressions tests to see if the *candidate expression* (often recorded in a patient record) is subsumed by a *predicate expression* (typically part of the query being run across the patient record). To enable subsumption testing it is necessary to ensure that the expression is processed in the same way as the precoordinated content. This requires the expression to be positioned in the right coordinates of the SNOMED CT hierarchy (aka coordinated), as illustrated in Figure 4.5-1.

Consider the expression:

```
372244006 |Malignant melanoma| :
{ 363698007 |Finding site| =
( 16982005 |Shoulder region structure| : 272741003 |Laterality| = 24028007 |Right| ) }
```

Subsumption testing may be performed to test if this expression is a subtype of the concept 300848003 |Mass of body structure (finding)| . This test could be required to identify a cohort of patients who have been diagnosed with some kind of lump in a body structure.

However, to enable systems to perform this test, the expression first needs to be classified with the precoordinated content to generate an inferred subtype hierarchy including both the postcoordinated expression and the query concept. If a classification is not performed, the software will not be able to automatically derive where the expression belongs in the SNOMED CT hierarchy relative to the query concept.

As illustrated in the image below, the candidate expression *is* subsumed by the predicate expression.

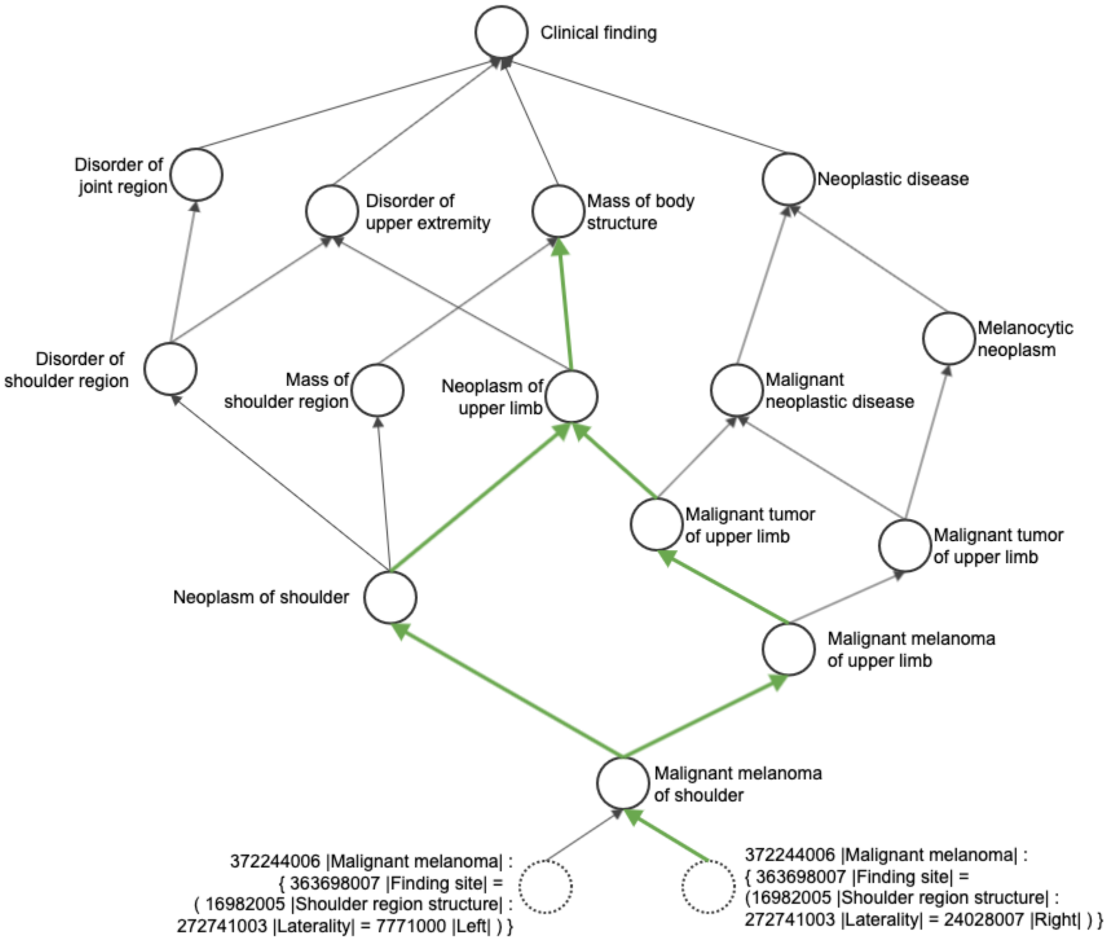

Clinical finding
Disorder of joint region
Disorder of upper extremity
Mass of body structure
Neoplastic disease
Melanocytic neoplasm
Disorder of shoulder region
Mass of shoulder region
Neoplasm of upper limb
Malignant neoplastic disease
Malignant tumor of upper limb
Malignant tumor of upper limb
Neoplasm of shoulder
Malignant melanoma of upper limb
Malignant melanoma of shoulder
372244006 |Malignant melanoma| : { 363698007 |Finding site| = (16982005 |Shoulder region structure| : 272741003 |Laterality| = 7771000 |Left|) }
372244006 |Malignant melanoma| : { 363698007 |Finding site| = (16982005 |Shoulder region structure| : 272741003 |Laterality| = 24028007 |Right|) }

Figure 4.5-1: Subsumption testing of expressions requires the expressions to be processed as if they were part of the released (precoordinated) SNOMED CT content.

Selective Retrieval

Systems supporting postcoordination should provide a mechanism to allow users to specify retrieval requirements based on the inferred properties of an expression. Attribute relationships in conjunction with the subtype hierarchies provide an efficient mechanism for specifying query criteria, e.g. a query for *"all diseases with a* | Finding site| *of* | Shoulder region structure| *and an* | Associated morphology| *of* | Edema| *"*. As illustrated in Figure 4.5-2, the inferred properties can be used to identify the concepts and expressions matching these query criteria. However, as for subsumption testing, this kind of query feature demands for the expressions to be classified with the precoordinated concepts in order for the expressions to infer all necessary properties correctly.

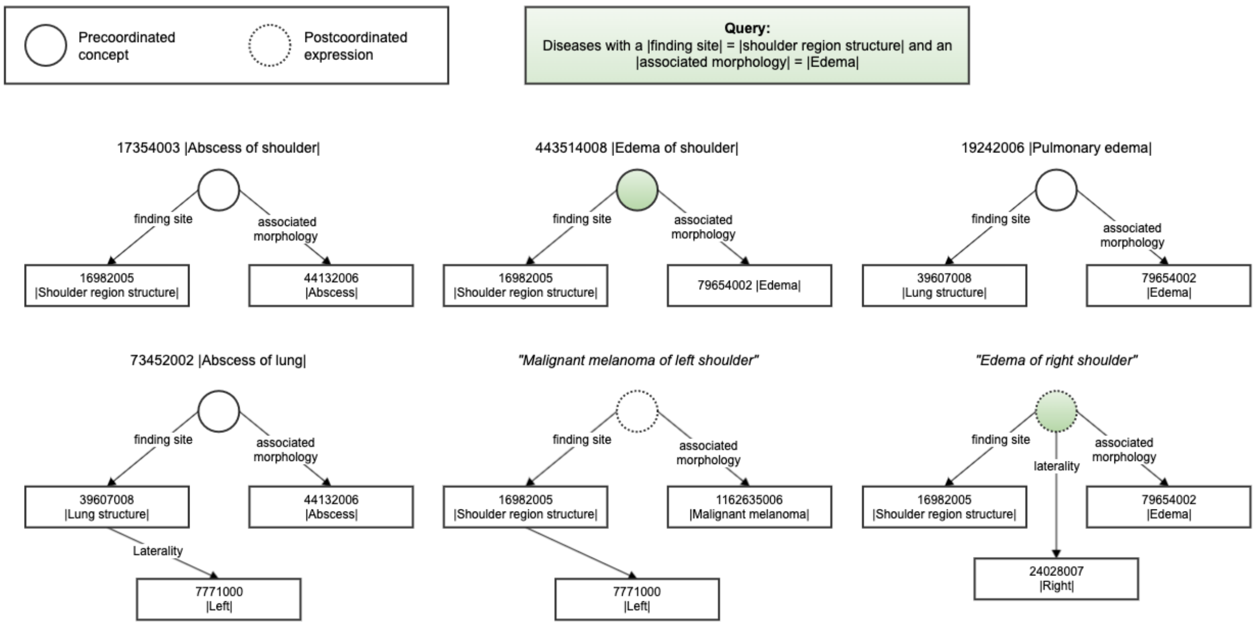

Precoordinated concept
Postcoordinated expression
Query:
Diseases with a |finding site| = |shoulder region structure| and an |associated morphology| = |Edema|
17354003 |Abscess of shoulder|
finding site
associated morphology
16982005 |Shoulder region structure|
44132006 |Abscess|
443514008 |Edema of shoulder|
finding site
associated morphology
16982005 |Shoulder region structure|
79654002 |Edema|
19242006 |Pulmonary edema|
finding site
associated morphology
39607008 |Lung structure|
79654002 |Edema|
73452002 |Abscess of lung|
finding site
associated morphology
39607008 |Lung structure|
44132006 |Abscess|
Laterality
7771000 |Left|
"Malignant melanoma of left shoulder"
finding site
associated morphology
16982005 |Shoulder region structure|
1162635006 |Malignant melanoma|
7771000 |Left|
"Edema of right shoulder"
finding site
laterality
associated morphology
16982005 |Shoulder region structure|
79654002 |Edema|
24028007 |Right|

Figure 4.5-2: Illustration of the principles of how the inferred properties of concepts and expressions can be utilized to specify query criteria selectively.

Terminology services

Enabling queries over expressions to be performed requires the availability of the following terminology services:

- 5.2.10 Query Expression - Including both services to perform subsumption testing and selective retrieval of expressions

4.6 Exchanging Expressions

When expressions are used to record clinical data, it is important to establish effective ways of communicating these expressions with other systems.

The main options for sharing postcoordinated expressions are:

- To exchange the terse expression as the ‘code’ of the relevant field, with the displayed or auto-generated term
    - For example, the HL7 version 3 CD data type (Release 2) allows the 'code' property to include SNOMED CT expressions (see HL7 TermInfo guide)
- To decompose the expression into separate fields - one for a focus concept and then one for each of the attribute values in the expression
    - In each field, the concept identifier may be used as the code, and one of the synonyms as the display text
    - This approach assumes alignment between the message structure and the relevant parts of the SNOMED CT concept model
- To use an expression identifier from your expression repository as the ‘code’, with the display term.
    - This approach requires the use of a shared expression repository enabling the recipient system to look up and resolve this identified to the relevant expression

Example: A FHIR Resource including a postcoordinated expression

```
{
  "resourceType": "Condition",
  "id": "f205",
  "text": {
    "status": "generated",
    "div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">...</div>"
  },
  "clinicalStatus": {},
  "verificationStatus": {},
  "code": {
    "coding": [
      {
        "system": "http://snomed.info/sct",
        "code": "301354004 |Pain of ear| :  272741003 |Laterality| = 7771000 |Left|",
        "display": "Left ear pain"
      }
    ]
  },
  "subject": {
    "reference": "Patient/f201",
    "display": "PP"
  },
  "recordedDate": "2013-04-04",
  "asserter": {
    "reference": "Practitioner/f201"
  }
}
```

Terminology services

Enabling expressions to be included in communication messages requires:

- 5.2.7 Lookup Expression. This will be used to get either the CTU form of the expression or the expression identifier to be included as the code for the coded data element in the message
- 5.2.9 Get Display Term. This is required in cases where a single display term is required for the expression, and this approach is acceptable in the given setting.

Note that additional terminology services will be required to decompose expressions to enable the separation of values into specific fields of the message structure.

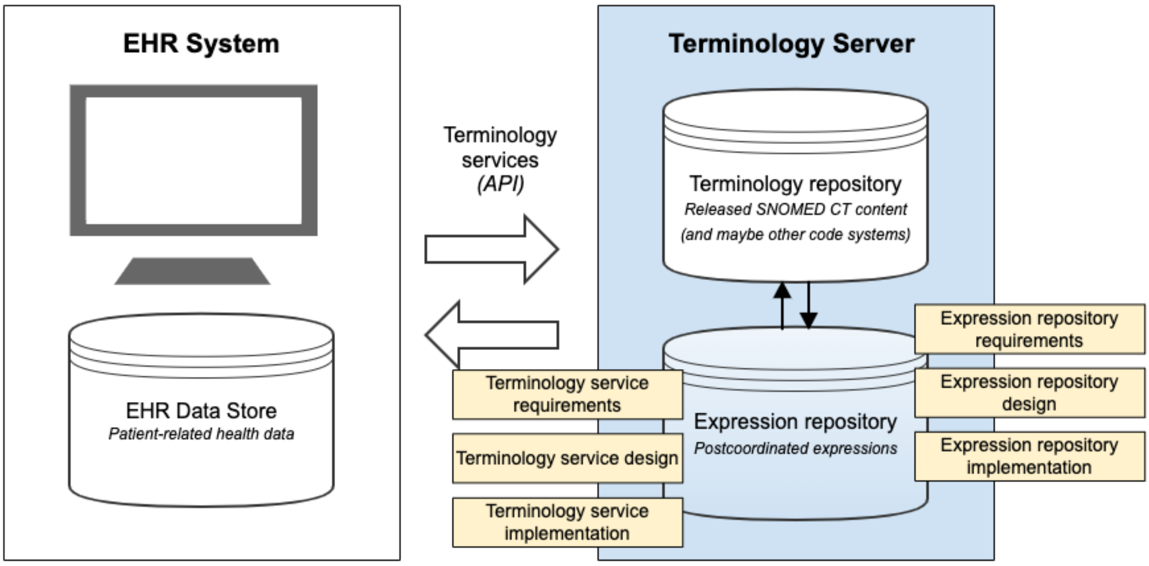
5. Expressions in a Terminology Server

To enable postcoordination within electronic health records a terminology server is required that provides the services needed to validate and process expressions. These services are similar to the ones used when authoring content within a published SNOMED CT edition.

When submitting postcoordinated expressions to a terminology server, the expressions can be seen as a derivative, or a supplement, to the SNOMED CT Edition content available in the terminology server. Postcoordinated content lives alongside and can only be interpreted or understood in the context of a specific SNOMED CT Edition version.

Because of the dependency on a versioned SNOMED CT Edition, the successful implementation of expressions involves services to continuously ensure validity and traceability. Similar to the approach to creating SNOMED CT extensions containing clinical concepts, postcoordinated expressions, therefore, need to follow the general principles for extensions, including validation, classification, and versioning.

As illustrated in Figure 5-1, the terminology server should, therefore, be designed to hold the given versioned SNOMED CT Edition, and an expression repository to hold the postcoordinated expressions. Furthermore, in addition to the services required by the EHR to use the content and derivatives of released SNOMED CT Editions (see Terminology Services Guide), services are required to support the use of the postcoordinated content.



EHR System
EHR Data Store
Patient-related health data
Terminology services (API)
Terminology Server
Terminology repository
Released SNOMED CT content
(and maybe other code systems)
Expression repository
Postcoordinated expressions
Terminology service requirements
Terminology service design
Terminology service implementation
Expression repository requirements
Expression repository design
Expression repository implementation

Figure 5-1: Overall EHR design with a terminology server supporting postcoordination.

The following pages will elaborate on the requirements, overall design, and implementation of a SNOMED CT-enabled expression repository, and present the services needed for supporting expressions in a terminology server.

5.1 Requirements

A terminology server supporting postcoordination should support the general features of a SNOMED CT-enabled terminology server (Terminology Services Guide), and additionally, support the features required to manage and use postcoordinated expressions. This involves an expression repository to store the postcoordinated expressions, and a set of services to access, process and query the expressions in the same way as pre-coordinated SNOMED CT content. This section provides an overview of the general requirements of an expression repository and the associated services.

5.1.1 Expression Repository Requirements

Overview

Table 5.1.1-1 provides an overview of key requirements for an expression repository. Additional requirements may be relevant for certain use cases, and this overview should, therefore, be used as a minimum set. The definition of the terms used in this table can be found below.

Requirement			Priority
Expression forms	Close-to-user form expression	The expression repository must store the expression as it was entered by the user	Mandatory
	Classifiable form expression	The expression repository should store the expression in a form that can be used as input to the classifier when generating the necessary normal form	Optional
	Necessary normal form expression	The expression repository should store the expression in a form that is suitable for querying	Optional
Identification	Expression id	The expression repository should store a unique identifier that can provide an easy reference to the expression	Optional
Versioning	Effective Time	The expression repository must store an effective time for each version of an expression. Although the CTU will never change, the CF and NNF may.	Mandatory
	Substrate	The expression repository must include a reference to the SNOMED CT content used to generate the expression.	Mandatory

Table 5.1.1-1: Expression repository requirements.

Requirements

Expression Forms

The following three expression forms may be supported by the expression repository:

- **Close-to-user form**
  - The CTU represents the expression as it was entered or created
  - The CTU is the primary stored and communicated view of the expression
  - The CTU is immutable, i.e. it will not change even though the repository is updated to new versions of SNOMED CT
- **Classifiable form**
  - The CF expression is a syntactically valid and concept model compliant representation of the close-to-user form expression
  - The CF serves as the input to the classifier, enabling expressions to be classified together with other SNOMED CT content
  - The CF may change when an expression repository is updated to a new version of SNOMED CT
  - When multiple classifiable forms exists for a CTU, these will be considered as different axioms when generating the NNF
- **Necessary normal form**
  - The NNF is the output of classifying the CF expression with a given SNOMED edition
  - The NNF represents the necessary relationships used for querying, so this becomes part of your substrate when you’re running an Expression Constraint query

- The NNF includes refinements representing inferred relationships, without any redundant refinements or relationship groups
- For each CTU, there will be one NNF

When included, the CTU, CF, and NNF should be stored in the expression repository in their canonical form. The canonical form is produced by applying a set of rules to the SNOMED CT expression to ensure a single unique representation of the expression.

Identification

The expression repository should support the unique identification of all expressions in the expression repository. The Canonical Close-to-user Form of an expression may be used to identify itself, but some implementations may require the storage of a fixed length, unique identifier for expressions used in an operational environment.

Some clinical systems have a limitation on the number of characters that can be used to identify clinical meanings in their databases. In systems that limit the length of these clinical meaning representations to 18 digits or less, postcoordinated expressions that are longer than this must be stored using a globally unique and unambiguous identifier containing 18 digits or less. A maximum of 18 digits is considered appropriate for this requirement, as this is the maximum length of standard SNOMED CT concept identifier. Therefore, any clinical system which supports SNOMED CT precoordinated concepts must support this number of characters.

Versioning

The expression repository should support version history. Although there will only ever be one version of the CTU, there is likely to be many versions of the CF and NNF over time. Versioning is therefore required to allow lookup and analysis of previous versions of expressions. This involves the representation of the effective time and the substrate.

Effective Time

For each version of an expression, the date should be recorded to enable the lookup of an expression for a particular point in time.

Substrate

For each version of the expression, the *substrate* used to generate the expression should be stored.

The *substrate* is the SNOMED CT content used for creating the expression. Because medical knowledge is constantly changing, and SNOMED CT evolves to reflect these changes, the concepts used to create an expression may be impacted over time. Therefore, it is important that the substrate is kept current to ensure that the meaning of the expression can be interpreted. With this in mind, both the SNOMED CT edition and the specific version of that edition (released on a given date) need to be included when determining the SNOMED CT substrate for an expression.

The SNOMED CT URI Standard defines a common format of URIs for identifying various SNOMED CT artefacts, including components and RF2 releases. This includes URIs for formally identifying the SNOMED CT international edition, national editions, and any specific versions thereof and can, therefore, be used to represent the substrate for the repository.

5.1.2 Terminology Services Requirements

This page introduces the users, use cases and resulting service requirements for a terminology server supporting postcoordination.

Users

Table 5.1.2-1 provides an overview of the key users who interact with an expression repository.

User	Description
Information Manager	• This user represents the people involved with the implementation of SNOMED CT-enabled systems and the maintenance of postcoordinated expression repositories • This user group interacts with an Expression Management System, designed to support key tasks required to create and maintain postcoordinated expressions in an expression repository
Clinical user	• This user represents the people involved with the entry of clinical data in an EHR system (or another type of clinical information system) • This user group interacts with an Electronic Health Record system when capturing and viewing clinical information

Table 5.1.2-1: Overview of key users interacting with an expression repository.

Use Cases

This set of services will enable users interacting with an expression repository to perform required tasks.

Table 5.1.2-2 provides an overview of requirements for each group of users.

User	Functional Requirement	Service Requirements
Information Manager	Should be able to set up the expression repository when preparing for an implementation	Create expression repository
	Should be able to retrieve details about the expression repository to enable referencing the content of this specific expression repository, e.g. for adding and updating content in the expression repository.	Get repository details
	Should be able to add expressions to the repository when these are created prior to the implementation within an EHR	Add expression
	Should be able to check if an expression constraint conforms to the concept model rules	Validate expression
	Should be able to check if expressions exists in the repository, e.g. as part of exploring if new expressions should be created	Lookup expression
	Should be able to view the details of each expression in the expression repository	Lookup expression
	Should be able to search for expressions using term search	Search for expression
	Should be able to retrieve expressions matching specific search criteria, e.g. as part of bulk edit features, expression reviews, etc.	Query expressions

User	Functional Requirement	Service Requirements
	Should be able to update expressions in an expression repository to enable correction of errors or to support updates to a new substrate	Update expression
	Should be able to disable or inactivate expressions that are no longer recommended for use	Inactivate expression
Clinical user	Should be able to add expressions to the expression repository, e.g.when the EHR supports runtime creation of expressions using form-based data entry	Add expression
	Should be able to see a term for an expression, e.g. when this is rendered as an option in a clinical user interface	Get display term
	Should be able to search for expressions using text-based search, e.g. when expressions are included in a value set bound to a search field	Search for expression
	Should be able to store expressions in the EHR data store	Add expression
	Should be able to share details about an expression, e.g. to support communication and exchange	Lookup expression
	Should be able to query expressions in the same way as pre-coordinated concepts to support tasks such as reporting, clinical data analysis and decision support	Query expressions

Table 5.1.2-2: Overview of functional requirements for each user involving interactions with the expression repository.

Service Requirements

Create expression repository

Services should be available to support the creation of an expression repository, involving the creation of the structures required to hold the data of the expression repository, as described in 5.1.1 Expression Repository Requirements. These services are required to provide an 'empty' expression repository.

Get repository details

Services should be available for getting details about a generated expression repository. The details can be used when accessing content of this specific expression repository, e.g. when adding, updating or querying expressions.

Add expression

Services should be provided to support the addition of new expressions to the expression repository. These services should ensure that all data required to represent and process expressions are available.

Services for adding expressions will often include the services for validating the expression, transforming the expression and generating the necessary normal form.

Transform expression

Services should be provided to support transformation of the close-to-user form expression to a classifiable form which is required to generate the necessary normal form.

Considerations and approaches to transformation are described in 5.3 Implementation.

Validate expression

Services should be available to check if the close-to-user form of the expression is syntactically valid and conforms to the concept model rules, or any allowed rules for postcoordination.

Generate Necessary Normal form

Services should be provided to support the generation of the necessary normal form of the expression, which is a prerequisite for supporting expressions to be queried in the same way as pre-coordinated content.

Considerations and approaches to the generation of the necessary normal form are described in 5.3 Implementation.

Update expression

Services should be provided to support the updating of expressions while retaining full traceability of previous versions of the expression (similarly to the principles of updating precoordinated SNOMED CT content).

Inactivate expression

Services should be provided to support the indication that an expression is no longer recommended for use. Although inactivated, the expression should be retained in the repository and full traceability of previous versions (similarly to the principles of inactivating precoordinated SNOMED CT content).

Lookup expression

Services should be provided to support access to details about the individual expressions in the repository.

Search for expression

Services should be provided to support searches for expressions in the repository. Lexical searches should be supported based on the terms associated with the concepts included in the expression.

Get display term

Services should be provided to support a human-readable representation of the expression. This service is required to enable the display of expressions in user interfaces, or to support term searching.

Subsumption testing

Services should be provided to test the subsumption relationship between a pair of expressions given the semantics of subsumption

Query expressions

Services should be available to enable subsumption testing between expressions and enable the selective retrieval of postcoordinated expressions satisfying stated criteria. Query services should enable queries to be performed over the content of a specific expression repository plus the content of the versioned edition of SNOMED CT that the repository depends on.

5.2 Design

This part of the guide introduces the design features of a terminology server supporting postcoordination with SNOMED CT.

This design meets the requirements outlined in 5.1 Requirements, and includes:

5.2.1 Expression Repository Design

The logical design of an expression repository includes the entities required to hold the data relevant for an expression version.

As illustrated in Figure 5.2.1-1, an expression repository contains one or more expression versions.

Each expression version must contain the following data:

- *Versioning data*, to represent the time and substrate of the given version of the expression
- A *Close-to-user form expression*, to represent the expression as it was stated upon creation

Each expression version should contain the following data:

- An *expression identifier*, **if** this is required to uniquely identify the expression
- A *Classifiable form expression*, to enable the generation of the NNF expression
- A *Necessary Normal Form expression*, to support querying the expression

All SNOMED CT concepts referenced by expressions in the repository at a particular point in time should all be included, and active, in a specific versioned edition of SNOMED CT.

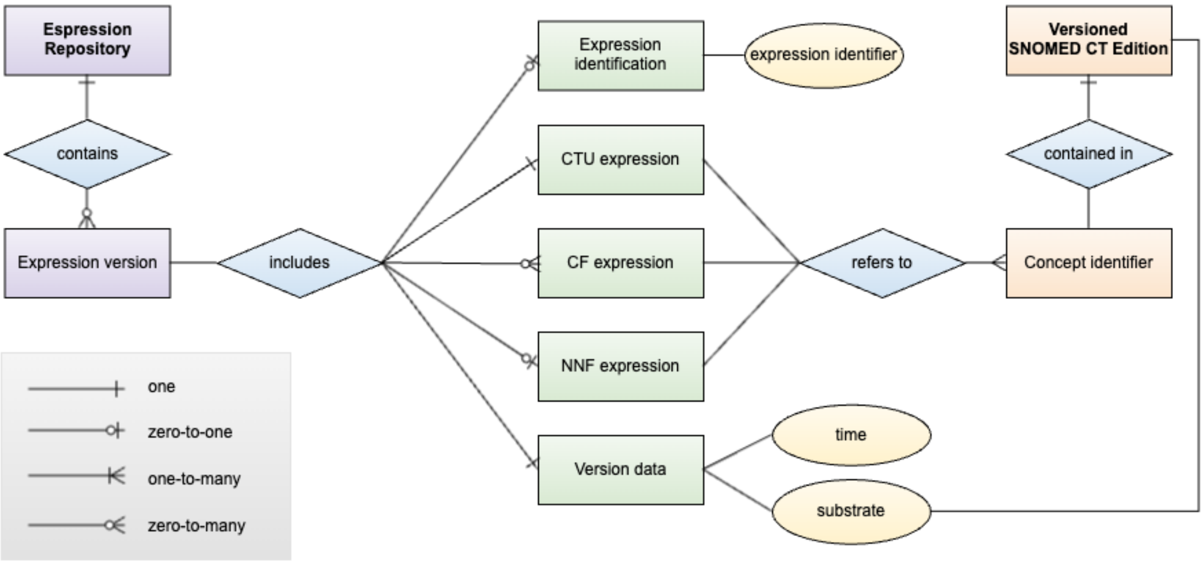


Figure 5.2.1-1: General expression repository design.

{

5.2.2 Terminology Services Design

Interacting with a terminology server is done via a set of supported services designed to meet key user requirements. The following pages provide details about the services required for a terminology server supporting postcoordination. Please note that the services designed to support an expression repository will often be used in conjunction with the services available to access precoordinated content. The general services of a SNOMED CT-enabled terminology server can be found in the Terminology Services Guide.

The following pages provides details on the services required by a terminology server supporting postcoordination with SNOMED CT.

5.2.1 Create Expression Repository

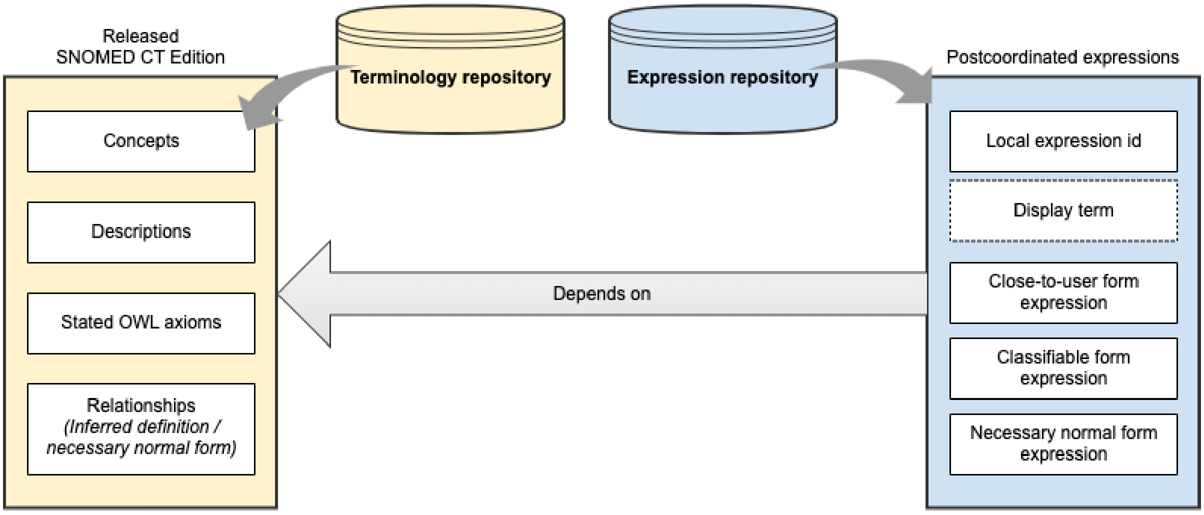
Objective

The objective of services for creating an expression repository is to set up the structures of the expression repository and provide an 'empty' expression repository.

Description

To set up an expression repository, it requires a terminology server with access to a SNOMED CT versioned edition. This is important because all concepts referenced by an expression must be interpreted in the context of a specific SNOMED CT Edition. To support processing of the expressions, access to information about the source concepts and to the particular version of the MRCM that is encoded within the version of the substrate is needed.

Note the MRCM does not change very frequently: several different consecutive versions of the substrate will contain different sets of concepts and their stated relationships, but may contain essentially the same MRCM content and version.


Released
SNOMED CT Edition
Concepts
Descriptions
Stated OWL axioms
Relationships
(Inferred definition / necessary normal form)
Terminology repository
Expression repository
Depends on
Postcoordinated expressions
Local expression id
Display term
Close-to-user form expression
Classifiable form expression
Necessary normal form expression

In addition to specifying the 'substrate' of the expression repository, it is important to specify the primary language, or languages, that should function as the default language for the repository. This is important to support the display of terms associated with the expressions. For the language configuration, it is recommended to specify which language reference set(s), available in the substrate, should be used as the default.

Summary

Service	Description	Input Parameters	Response
Create expression repository	Creates a new repository	- **substrate**: the base edition for this repository - **languages** (0-*): pairs of language refsetIds + language codes used to generate descriptions (optional)	**Success:** True, with identifier of the generated repository **Failure:** False, with appropriate error message

5.2.2 Get Details of an Expression Repository

Objective

To retrieve details about a generated expression repository. The details of the repository can be used when referring to the content of this specific expression repository, e.g. for adding and updating content in the expression repository.

Summary

Service	Description	Input Parameters	Response
Get details about an expression repository	Gets details of a repository	**- substrate:** the base edition for this repository **- repId:** a unique reference to the expression repository	**Success:** Return details of the selected expression repository and dependencies **Failure:** Return error message

5.2.3 Update Expression Repository

Objective

The objective of a service to update an expression repository is to support the ongoing maintenance and alignment with new versions of SNOMED CT used as the substrate for the expression repository.

Description

Updating an expression repository involves changing the substrate of an expression repository to a new, or another, substrate. Often, the new substrate will be to a new version of the expression repository substrate. Any changes made to the terminology between the SNOMED CT version being the current substrate and the SNOMED CT version that will become the new substrate may impact the content of the expression repository. For example, if a concept included in one of the expressions stored in the repository has been made inactive, the expression is no longer valid after applying the new substrate. Or, if concept model rules stated in the Machine Readable Concept model have been modified, expressions may no longer be concept model compliant.

The services for updating of the expression repository needs, therefore, need to support

1. Validate the CTU of all existing expressions against the new substrate
2. Resolve any validation issues, e.g. by
    a. replacing inactive concepts with proper replacements
    b. adjusting expression to conform to the MRCM
3. Transform the updated expressions (update the classifiable form expressions)
4. Classify the expression repository against the new substrate, and where relevant, generate new versions of the necessary normal form

Summary

Service	Description	Input Parameters	Response
Update expression repository	Creates a new version of a specific repository	- **substrate**: the new base edition for this repository - **repId**: identifier of the repository to be updated - **languages** (0-*): pairs of language refsetIds + language codes used to generate descriptions (optional)	**Success:** True **Failure:** False, with appropriate error message

5.2.4 Add Expression

Objective

The objective of the services described in this page is to enable the addition of a new expression in the expression repository.

Description

Adding an expression to an expression repository requires the validation of the expression and the storage of the expression as it has been entered by the user with the additional data and expression forms that are needed for subsequent access and use.

Workflow

The logical process of the add expression service is illustrated in the diagram below.

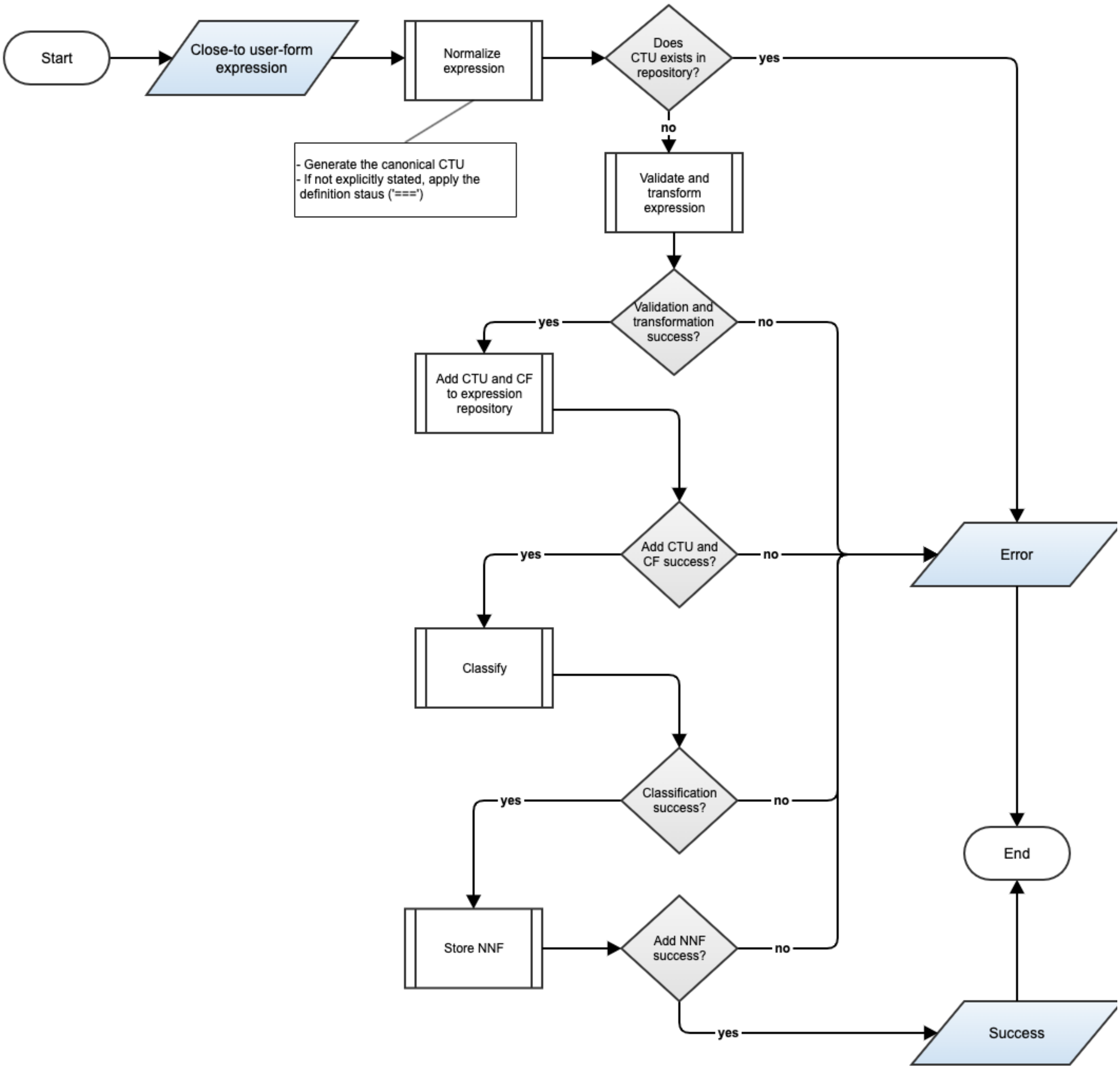

Start
Close-to user-form expression
Normalize expression
- Generate the canonical CTU
- If not explicitly stated, apply the definition staus ('===')
Does CTU exists in repository?
yes
no
Validate and transform expression
Validation and transformation success?
yes
no
Add CTU and CF to expression repository
Add CTU and CF success?
yes
no
Classify
Classification success?
yes
no
Store NNF
Add NNF success?
no
yes
Error
End
Success

Table 5.2.4-1: Add expression service

Summary

Service	Description	Input Parameters	Response
Add expression	Adds a new postcoordinated expression to the expression repository	**- substrate:** the base edition for this repository **- repId:** a unique reference to the expression repository **- CTU:** Close to user form expression	**Success:** Return data associated with the expression • Expression identifier, if generated • CTU expression • CF expression, if generated • NNF relationships, if generated **Failure:** *"Expression not added"* plus an indication of the reason (e.g. transformation issue, classification issue, term generation issue)

5.2.5 Validate and Transform Expression

Objective

The objective of this service is to validate a CTU expression and transform it into a classifiable form (CF) which can be used as input to the Description Logic classifier.

Description

This service involves both the validation and transformation of a CTU expression. Three types of validation are required, including syntactic validation, content validation, and MRCM validation. For expressions not fully compliant with the MRCM, a transformation is required to generate the classifiable form.

Syntactic Validation

The syntactic validation is performed to determine if the expression conforms to the SNOMED CT compositional grammar syntax (see Compositional Grammar - Specification and Guide section 7.2 Parsing).

Please refer to the Compositional Grammar - Specification and Guide (section 7.3 Validating) for more details on the validation of expression, and the Machine Readable Concept Model for details on the computer processable concept model rules.

Content Validation

The content validation is performed to determine if all concept references included in the expression exists and are active in the version and edition of SNOMED CT which is the substrate for the given expression repository.

MRCM Validation

The MRCM validation is performed to determine if the expression complies to the concept model rules as expressed in the SNOMED CT Concept Model and specified by the machine readable concept model (the version of the MRCM applying to the version and edition of SNOMED CT which is the substrate for the given expression repository)

An expression that does not conform to the *concept model* cannot be relied upon to correctly classify. Therefore, it cannot be reliably tested for subsumption by another *concept* or inclusion in (or exclusion from) the results of an expression constraint or analytics query.

Specific terminology services required to support MRCM validation are described in the general Terminology Services Guide here: SNOMED CT concept model

Transformation

Expressions that are fully compliant with the Machine Readable Concept will not need a transformation to be represented in the classifiable form. This means that for fully MRCM compliant expressions the close-to-user form and the classifiable form of the expression are the same.

An expression may be invalid when tested against the concept model but may contain refinements that would be valid if correctly structured. There are several situations where predictable structural adjustments enable a valid expression to be constructed from an informal expression that does not fully conform to the concept model.

Examples include:

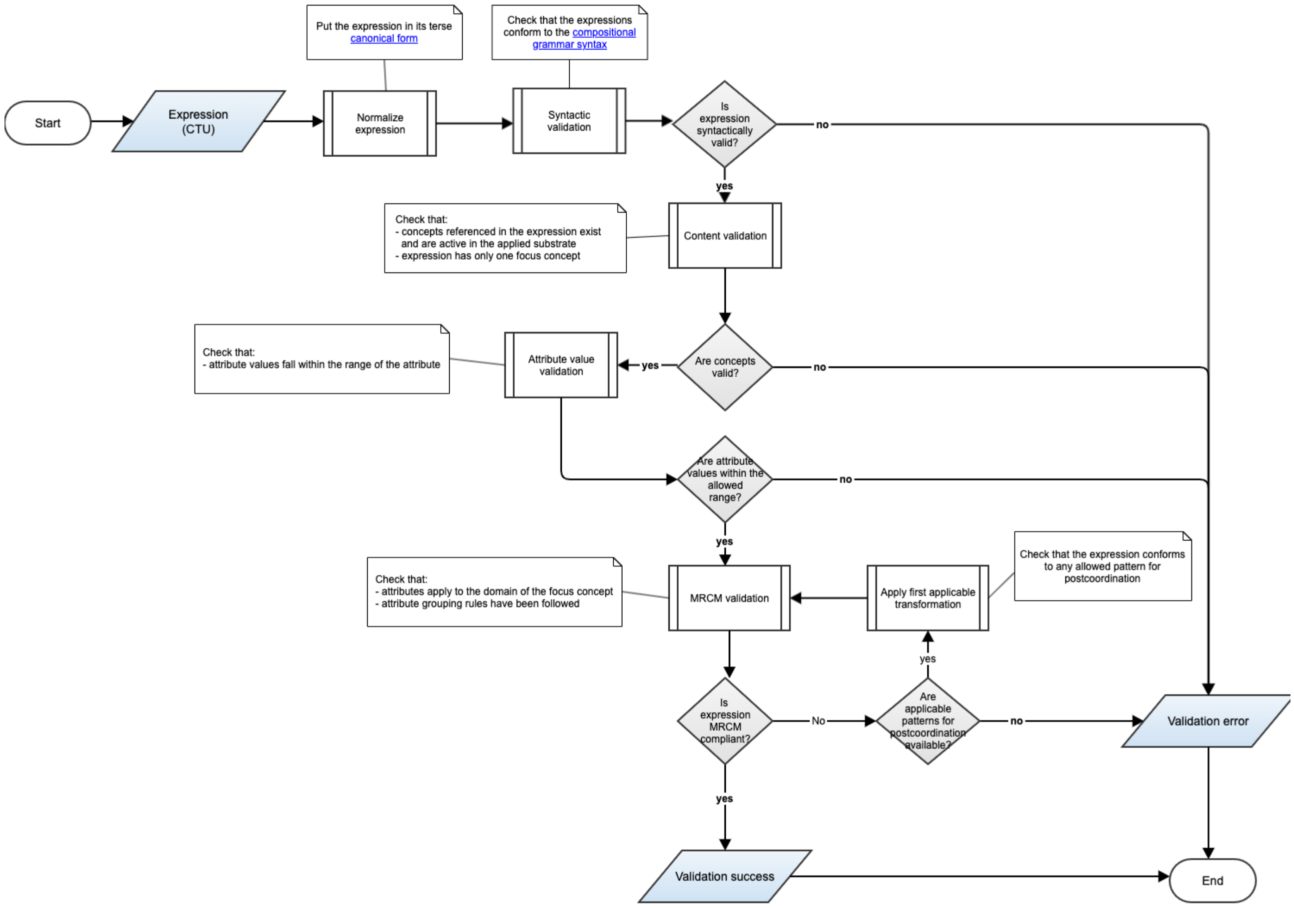
- Moving a loose attribute to an attribute group in which it is valid
- Moving a loose attribute so it provides a nested refinement to the value of another attribute

A loose attribute is an attribute that is included in the refinement of an expression without being placed in a group, although the concept model rules require it to be grouped.

The extent to which a terminology server supports the transformation of non-MRCM compliant expressions into MRCM-compliant expressions may vary between terminology servers. Each terminology server should, therefore, clearly state the level of transformations supported for expressions that are not fully MRCM compliant. A recommended practice for this it to support an API call by which it is possible to determine the transformation level. For more detail on these levels, see 5.3 Implementation

Workflow

The logical process of the service to transform and validate an expression is illustrated in the diagram below.


Put the expression in its terse canonical form
Check that the expressions conform to the compositional grammar syntax
Start
Expression (CTU)
Normalize expression
Syntactic validation
Is expression syntactically valid?
no
yes
Check that:
- concepts referenced in the expression exist and are active in the applied substrate
- expression has only one focus concept
Content validation
Check that:
- attribute values fall within the range of the attribute
Attribute value validation
yes
Are concepts valid?
no
Are attribute values within the allowed range?
no
yes
Check that:
- attributes apply to the domain of the focus concept
- attribute grouping rules have been followed
MRCM validation
Apply first applicable transformation
Check that the expression conforms to any allowed pattern for postcoordination
yes
Is expression MRCM compliant?
No
Are applicable patterns for postcoordination available?
no
Validation error
yes
Validation success
End

Summary

Service	Description	Input Parameters	Response
Validate and transform expression	Check that the CTU is syntactically valid and conforms to the concept model rules, or allowed rules for postcoordination. Transforms the close-to-user form expression to a classifiable form	**- id:** a unique reference to the expression repository **- expId:** CTU expression or expression id	**Success:** True **Failure:** False, with validation error indication

{

5.2.6 Generate Necessary Normal Form

Objective

The objective of this service is to support classification and Necessary Normal Form generation

Related Use Cases

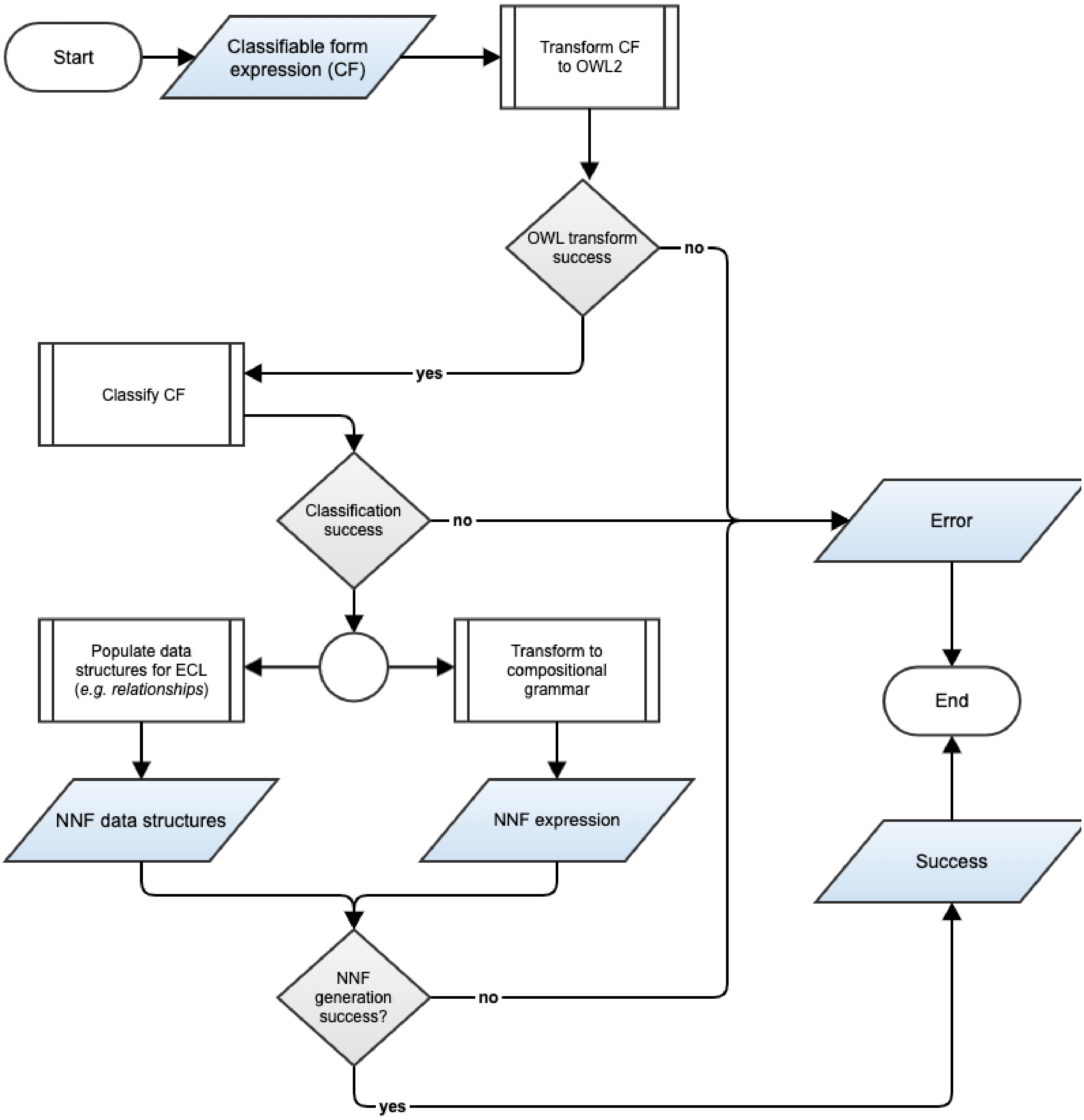
Classification is required to enable the following use case:

- 2.6 Query Expressions

Description

To enable postcoordinated expressions to be queried in the same way as pre-coordinated SNOMED CT concepts, the necessary normal form of the expression is required, as illustrated in the diagram below.

Workflow



Summary

Service	Description	Input Parameters	Response
Generate NNF	Classifies the expression (CF) and generates the necessary normal form of the expression	**- repId:** a unique reference to the expression repository **- CF:** Classifiable form	**Success:** As a minimum, returns the necessary normal form of the expression **Error:** Display an appropriate error message

5.2.7 Lookup Expression

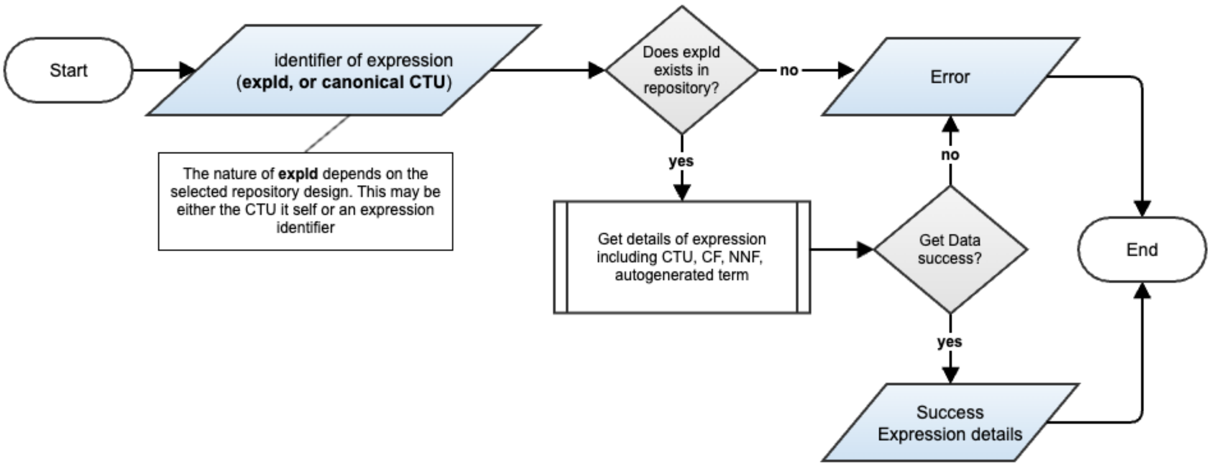
Objective

The objective of this service is to get data associated with an identified expression.

Description

Services for retrieving data associated with an expression rely on an input pointing to the expression of interest. For this purpose, a unique identifier associated with the expression within an expression repository should be input to this service. If the CTU of the expression itself is used to uniquely identify the expression, this will be used for the lookup. If multiple versions of the expression exists, the service will return the most recent version of the expression.

Workflow


Start
identifier of expression
(expId, or canonical CTU)
The nature of expId depends on the selected repository design. This may be either the CTU it self or an expression identifier
Does expId exists in repository?
no
yes
Error
Get details of expression including CTU, CF, NNF, autogenerated term
Get Data success?
no
yes
Success
Expression details
End

Summary

Service	Description	Input Parameters	Response
Lookup expression	Get expression by identifier	**- repId:** a unique reference to the expression repository **- expId:** CTU expression or expression id	**Success:** Return data associated with the expression • CTU expression • CF expression • NNF relationships

Service	Description	Input Parameters	Response
			- Identifier for equivalent precoordinated concept **Failure:** Return an appropriate error message
	Get expression by close-to-user form expression	**- repId:** a unique reference to the expression repository **- CTU:** Canonical close-to-user form expression	**Success:** Return data associated with the expression - CTU expression - CF expression - NNF relationships - Identifier for equivalent precoordinated concept **Failure:** Return an appropriate error message

5.2.8 Search Expression

Objective

The objective of this service is to enable text-based searching for expressions in the expression repository.

Description

As specified in 5.1.1 Expression Repository Requirements, an expression repository includes the canonical form of the stored expressions. The canonical form excludes any terms associated with the referenced concepts.

Consider storing a term associated with the expression to aid searching.

Please note that this guide does not include guidance on the storage of a human-readable term for expressions, but it recommends implementing services to automatically generate a display term, see 5.2.9 Get Display Term. Some implementations may choose to store an autogenerated display term for each expression in the repository as part of adding the expression. This may be chosen to support easy lookup of the terms associated with the expressions to support services for searching.

Summary

Service	Description	Input Parameters	Response
Search for expression	Find expression by term	**- repId:** a unique reference to the expression repository **- Search string** - **Search technique** option (if required) - **Search filtering** or sorting options (if supported)	**Success:** A collection of matching terms each of which is linked to an expression within the expression repository. - Display term (generated) - Expression identifier **Failure:** Return an appropriate error message

{

5.2.9 Get Display Term

Objective

The objective of services to get a display term for an expression, is to support the automatic generation of a term that can be used for the human-readable representation of the term, e.g for display or exchange.

Description

Three different approaches to automatically creating display terms for postcoordinated expressions is presented, ranging from the simplest to the most complex.

1. Replace concept identifiers with terms
2. Apply simple term rules
3. Apply description templates and rules

Please refer to Appendix A: Techniques for Autogenerating Display Terms for details on these approaches. Please note that alternative approaches may be equally valid for this task.

Summary

Service	Description	Input Parameters	Response
Get term for expression	Gets an autogenerated term for the expression using the technique supported by the terminology server	- **substrate:** the base edition for this repository - **id:** a unique reference to the expression repository - **expression:** a unique reference to the expression - **technique:** if multiple techniques for generating terms exist, the technique to apply may be indicated	**Success:** Return the generated display term(s) **Failure:** Display an appropriate error message

5.2.10 Query Expression

Objective

A terminology server should support services that are required for patient- and population-oriented data retrieval and analytics tasks.
The objective of expression query services is to enable subsumption testing between expressions and enable selective retrieval of postcoordinated expressions satisfying stated criteria.

Description

Subsumption Testing

When expressions are used, it is necessary to determine whether the meaning of a particular expression is a subtype of a specified concept or more generally is subsumed by a particular expression constraint. The terminology services required are similar to those described for testing concepts in sections 4.5 Get and Test Concept Subtypes and Supertypes.

Selective Retrieval

As for precoordinated concepts, expressions may be queried based on the inferred properties derived from the classification process. Expression constraints may be formulated and executed against the expression repository and the associated substrate to return both precoordinated concepts and expressions that match the specified constraint, see 4.7 Validate and Apply Expression Constraints.

Summary

Service	Description	Input Parameters	Response
Query expression	Subsumption - check if there is a subsumption relationship between two expressions	**- repId:** a unique reference to the expression repository **- source:** CTU expression, expression id, or concept id **- target:** CTU expression, expression id, or concept id	**Success:** Return data to inform which of the following outcomes are true: • Source is a subtype of target • Target is a subtype of source • No subsumption relationship between source and target **Failure:** Display an appropriate error message
	Get concepts and expressions that conform to an expression constraint	**- repId:** a unique reference to the expression repository **- ecl:** an expression constraint complying to the Expression Constraint Language	**Success:** Return a set of conceptIds and CTU expressions that conform to the expression constraint Optionally additional information such as the fully specified name or preferred term of each concept included in the expansion plus generated terms for included expressions **Failure:** Return error message if expression constraint contains syntax errors Return error message if any concept identifiers in the expression constraint are not present in the specified edition

5.3 Implementation

This guide does not mandate a particular way to meet the requirements and the design of an expression repository as outlined in 4.1 Requirements and 4.2 Design. However, to ensure that terminology servers supporting postcoordination are implemented in a way that doesn't compromise terminological integrity and clinical safety, this section presents considerations important for the implementation of an expression repository. Also, different approaches to implementations are be presented.

Ensuring Safety

When implementing a terminology server supporting postcoordination with SNOMED CT, careful considerations must be taken to ensure that the approach taken doesn't compromise safety. As a combined effort between the terminology server and the client, the following criteria for expressions must, therefore, be ensured:

- All expressions should conform to the concept model when represented in their classifiable form
- No ambiguity is allowed, i.e. the transformation of the expression into its classifiable form must not result in a modification of the intended meaning of the expression

Meeting these criteria can be done in several ways and depends on the responsibility taken by the terminology server and client respectively.

Transformation Levels

Each terminology server designed to enable postcoordination with SNOMED CT should clearly specify the level of transformation supported, to enable requirements to be specified for the client side, and to support decisions about the use of postcoordination for the specific implementation. Figure 5.3-1 illustrates four levels of support for postcoordination. At each level, the sum of restrictions provided by the terminology server and the client must ensure that the criteria for 'safe' postcoordinations are met.

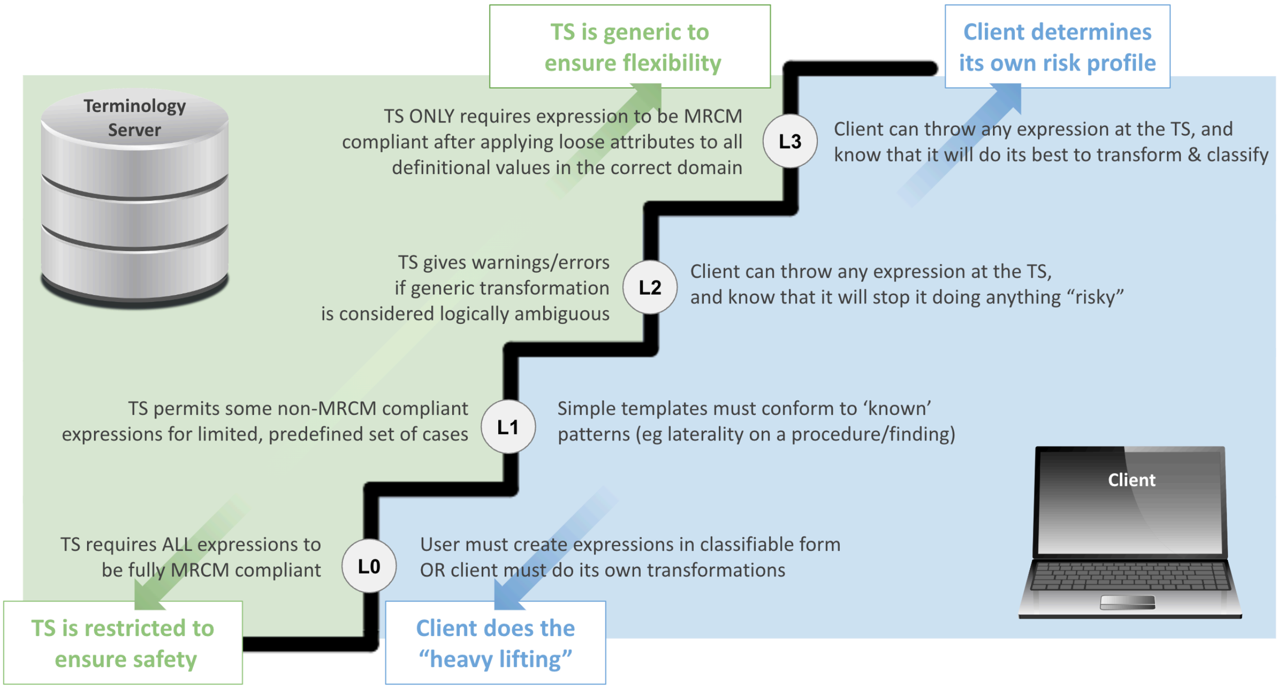

TS is generic to ensure flexibility
Client determines its own risk profile
Terminology Server
TS ONLY requires expression to be MRCM compliant after applying loose attributes to all definitional values in the correct domain
L3
Client can throw any expression at the TS, and know that it will do its best to transform & classify
TS gives warnings/errors if generic transformation is considered logically ambiguous
L2
Client can throw any expression at the TS, and know that it will stop it doing anything "risky"
TS permits some non-MRCM compliant expressions for limited, predefined set of cases
L1
Simple templates must conform to 'known' patterns (eg laterality on a procedure/finding)
Client
TS requires ALL expressions to be fully MRCM compliant
L0
User must create expressions in classifiable form OR client must do its own transformations
TS is restricted to ensure safety
Client does the "heavy lifting"

Figure 5.3-1: Overview of the four levels for support of postcoordination.

Terminology server transformations supported at the different levels

Transformation Level	Description	CTU	CF	Details
Level 0	The terminology server will classify the expression without transformation. The client is responsible for ensuring that expressions are fully MRCM-compliant.	CTU should be fully MRCM compliant	Same as CTU	5.3.1 Transformation Level 0

Level 1	The terminology server behaves as in level 0 if the expression is fully MRCM compliant. The terminology server will apply transformations based on known close-to-user form patterns to generate the classifiable form. The client is responsible for ensuring that expressions are fully MRCM-compliant or conform to allowed close-to-user form patterns.	CTU should be fully MRCM compliant or conform to close-to-user form patterns	Generated by transformation	5.3.2 Transformation Level 1
Level 2	*The transformation logic for these levels has not yet been agreed. At this stage, postcoordinations and transformation logic beyond the patterns described at level 1 are, therefore, not recommended.*			N/A
Level 3				N/A

Table 5.3-1: Overview of the four levels

5.3.1 Transformation Level 0

At this level, the terminology server requires all submitted expressions to already be in valid classifiable form. This form is a syntactically valid and concept model compliant representation expression that can be directly used as input to the classifier, enabling expressions to be classified together with other SNOMED CT content.

For more details on the Machine Readable Concept Model, please see http://snomed.org/mrcm.

Implementation Advice

To enable postcoordination at Level 0, it is important that the system used to create the expressions supports any transformation needed to represent the expressions in their classifiable form. This may be enabled using one of the following techniques:

- 4.1.3.2.2 Form-based Creation
- 4.1.3.2.3 Expression Templates

These techniques will support the creation of expressions that fully comply with the concept model rules and avoid some of the common mistakes created. For example using attributes that are incorrect for the domain or incorrectly grouped.

Incorrect Attributes

Consider the situation where the following expression is created to represent the meaning of a "Disorder of left lung":

```
=== 19829001 |Disorder of lung (disorder)| : 272741003 |Laterality (attribute)| = 7771000 |Left (qualifier value)|
```

At level 0, this expression would not be accepted by the terminology server, because the MRCM only permits those concepts that are both within the << 91723000 |Anatomical structure (body structure)| domain and also listed in the 723264001 |Lateralizable body structure reference set| to be qualified using 272741003 |Laterality (attribute)| . The focus concept of the expression is neither.

As a result of this rule, the action required to make the expression classifiable, is to add a nested expression where the laterality is applied to the value of the 363698007 | Finding site (attribute)| attribute that is part of the NNF for the same focus concept, 19829001 | Disorder of lung (disorder)| , as follows:

=== 19829001 |Disorder of lung (disorder)| : { 363698007 |Finding site (attribute)| = (39607008 |Lung structure (body structure)| : 272741003 |Laterality (attribute)| = 7771000 |Left (qualifier value)|)}

Incorrectly Grouped Attributes

Consider the situation where the following expression is created to express the meaning of a "Procedure on left lung":

=== 71388002 |Procedure (procedure)| : 363704007 |Procedure site| = 44029006 |Left lung structure (body structure)|

At level 0, this expression would not be accepted by the terminology server, because 363704007 | Procedure site (attribute)| is specified as being 'grouped', which requires a role group to be explicitly stated. As a result of this rule, the action required to make the expression classifiable, is to place the attribute in a group, as follows:

=== 71388002 |Procedure (procedure)| : { 363704007 |Procedure site| = 44029006 |Left lung structure (body structure)| }

Examples

Terminology servers at level 0 will reject any expressions not conforming to the classifiable form.

Table 5.3.1-1 provides examples of a set of expressions with an indication of their acceptability and potential issues at level 0.

Expression	Classifiable	Issue
=== 372244006 |Malignant melanoma| : 363698007 |Finding site| = 91775009 |Left shoulder|	**No**	Attribute 363698007 | Finding site| should be in a group
=== 372244006 |Malignant melanoma| :{ 116676008 |Associated morphology| = 1162635006 |Malignant melanoma| , 363698007 |Finding site| = 91775009 |Left shoulder | }	**Yes**	
372244006 |Malignant melanoma| :{ 363698007 |Finding site| = 91775009 |Left shoulder | }	**Yes**	Although this expression complies with the concept model rules, the expression will not classify as probably desired, as a subtype of 188060000 | Malignant melanoma of shoulder (disorder)| The reason for this is that the stated group does not include the required attribute: 116676008 | Associated morphology (attribute)| = 1162635006 |Malignant melanoma (morphologic abnormality)|
=== 372244006 |Malignant melanoma| : 363698007 |Finding site| = (16982005 |Shoulder| : 272741003 |Laterality| = 7771000 |Left|)	**No**	Attribute 363698007 | Finding site| should be in a group
	Yes	

Expression	Classifiable	Issue
=== 372244006 |Malignant melanoma| :{ 116676008 |Associated morphology| = 1162635006 |Malignant melanoma| , 363698007 |Finding site| = (16982005 |Shoulder| : 272741003 |Laterality| = 7771000 |Left|)}		
=== 188060000 |Malignant melanoma of shoulder| : 272741003 |Laterality| = 7771000 |Left|	**No**	Attribute 272741003 | Laterality| does not apply to the domain | Malignant melanoma of shoulder|

Table 5.3.1-1: Example expressions with an indication of their acceptability and potential issues at level 0

5.3.2 Transformation Level 1

Terminology servers at Level 1 will accept any expressions represented in valid classifiable form, as in Level 0. Additionally, some expressions that are not already in classifiable form will also be accepted. These will mainly be syntactically compliant expressions that are not MRCM compliant, but which the server can unambiguously transform to derived expressions that *are* MRCM compliant, using predefined automated transforms.

Loose Attributes

A loose attribute is any attribute within a CTU expression that is not in an attribute group AND the MRCM specifies:

- that it should be grouped, or
- that it is in the wrong domain

Scope of Transformations

Level 1 transformations of CTU expressions may apply to expressions meeting the following criteria.

- Transformations can only apply where there is no definition status
- There must be only one focus concept
- Level 1 transformations can only apply to loose attributes (as defined above)
- When the expression contains a mixture of loose attributes and attribute groups, the attribute groups are effectively treated as for Level 1, so no transformation of these are attempted

Transformation Process

Expression syntax, concepts being active and MRCM attribute-range validation should be applied before the transformation step. Implementations should fail fast if these are not met and not attempt transformation. This simplifies the preconditions for specific transformations.

The process of applying the transformations is iterative. This includes: collecting the set of remaining loose attributes, then applying the next transformation, and repeating until each transformation has been applied once. When each specific transformation is applied it should be greedy and transform all the loose attributes that meet the conditions of that transformation.

The list of transformations must be applied in a specific order; to ensure that the same classifiable expression is reached every time by all implementations.

Transformations

The allowed patterns can be summarized as follows:

The following pages elaborate on the preconditions and behaviour of each pattern.

5.3.2.1 Refining Existing Attribute

The terminology server will accept CTU expressions when the stated attribute(s) refines the meaning of the focus concept, without adding new attributes or groups.

In this case, the terminology server will accept attributes-value pairs with the following characteristics:

- The attribute is already present in the NNF of the focus concept
- The attribute value stated in the expression is a subtype of the attribute value of the focus concept

Important Notes

- If no groups are stated in the CTU expression (i.e. attributes are declared as loose attributes), the terminology server will accept attribute-value pairs correctly belonging to role groups of the focus concept as follows:
    - The terminology server will accept attribute-value pairs that refine one or more of the attribute-value pairs of the focus concept, i.e.
        - The attribute type should be the same as, or a subtype of, an attribute that already exists in the NNF of the focus concept, for which
            - The attribute value may be the same as, or a subtype of, the existing attribute value
    - The terminology server will reject attribute-values, if this is a supertype of the attribute value used in the definition of the focus concept
- If groups are stated in the CTU expression
    - The terminology server will accept groups that refine an existing role group of the concept, i.e. the group stated in the CTU should represent a subtype of the role group of the focus concept

Table 5.3.2.1-1: Examples of CTU expressions candidate to this pattern, and their evaluation at level 1.

CTU expression	Accept/ reject	Notes	Classifiable form
372130007 |Malignant neoplasm of skin| : 363698007 |Finding site| = 113179006 |Skin structure of nose|	Accept	The focus concept includes only one |Finding site| attribute with the value 39937001 |Skin structure|. The finding site value of the expression (i.e. 113179006 |Skin structure of nose|) is a subtype of 39937001 |Skin structure|	====== 372130007 |Malignant neoplasm of skin| : { 116676008 |Associated morphology| = 1240414004 |Malignant neoplasm| , 363698007 |Finding site| = 113179006 |Skin structure of nose| }

CTU expression	Accept/ reject	Notes	Classifiable form
372130007 |Malignant neoplasm of skin| : 363698007 |Finding site| = 176752008 | Breast structure|	Reject	The focus concept includes only one | Finding site| attribute with the value 39937001 | Skin structure|. The finding site value of the expression (i.e. 76752008 |Breast structure|) is not a subtype of 39937001 | Skin structure|	N/A
281444001 |Loosening of total shoulder replacement| : 255234002 |After| = 733429004 |Prosthetic total arthroplasty of right shoulder|	Accept	The focus concept includes only one |After| attribute with the value 31884000 |Implantation of joint prosthesis into shoulder joint|. The value of the |After| attribute in the expression (i.e. 733429004 |Prosthetic total arthroplasty of right shoulder|) is a subtype of |Implantation of joint prosthesis into shoulder joint|	=== 281444001 |Loosening of total shoulder replacement| : { 363698007 |Finding site| = 85537004 |Glenohumeral joint structure| } { 255234002 |After| = 31884000 |Implantation of joint prosthesis into shoulder joint| } { 47429007 |Associated with| = 304125002 |Total shoulder replacement prosthesis| } { 255234002 |After| = 733429004 |Prosthetic total arthroplasty of right shoulder| }

118473000 |Treatment of complex fracture | : 260686004 |Method| = 410814006 | Surgical reduction - action|	Accept	The focus concept of the CTU expression is defined with two method attributes
		The attribute of the CTU is a subtype of one of the attribute value pairs of the focus concept.
			=== 118473000 |Treatment of complex fracture| :
			{ 260686004 |Method| = 257903006 |Repair - action| , 405813007 |Procedure site - Direct| = 272673000 |Bone structure| }
			{ 260686004 |Method| = 129284003 |Surgical action| , 363704007 |Procedure site| = 272673000 |Bone structure| , 405816004 |Procedure morphology| = 72704001 | Fracture| }
			{ 260686004 |Method| = 410814006 |Surgical reduction - action| , 363704007 |Procedure site| = 272673000 |Bone structure| , 405816004 |Procedure morphology| = 72704001 | Fracture| }

29477005 |Repair of fracture with sequestrectomy| : 405813007 |Procedure site - Direct| = 41111004 |Bone structure of shaft of femur|	Accept	The focus concept of the CTU expression is defined with three groups, each containing a |procedure site- direct| method attribute The attribute of the CTU is a subtype of one of the attribute value pairs of the focus concept.	=== 29477005 |Repair of fracture with sequestrectomy| : { 260686004 |Method| = 129284003 |Surgical action| , 363704007 |Procedure site| = 272673000 |Bone structure| , 405816004 |Procedure morphology| = 72704001 | Fracture| } { 260686004 |Method| = 257903006 |Repair - action| , 405813007 |Procedure site - Direct| = 26107004 | Structure of musculoskeletal system| } { 260686004 |Method| = 129304002 |Excision - action| , 405813007 |Procedure site - Direct| = 272673000 | Bone structure| , 363700003 |Direct morphology| = 4857006 |

CTU expression	Accept/ reject	Notes	Classifiable form
			Sequestrum| } { 260686004 |Method| = 129284003 |Surgical action| , 363704007 |Procedure site| = 41111004 |Bone structure of shaft of femur| , 405816004 |Procedure morphology| = 72704001 |Fracture| } { 260686004 |Method| = 257903006 |Repair - action| , 405813007 |Procedure site - Direct| = 41111004 |Bone structure of shaft of femur| } { 260686004 |Method| = 129304002 |Excision - action| , 405813007 |Procedure site - Direct| = 41111004 |Bone structure of shaft of femur| , 363700003 |Direct morphology| = 4857006 |Sequestrum| }

6471000179103 |Transplantation of kidney and pancreas| : 405813007 |Procedure site - Direct| = 9846003 |Right kidney structure|	Accept	The loose attribute represents a subtype of one of the two |Procedure site - Direct| attributes of the focus concept	=== 6471000179103 |Transplantation of kidney and pancreas| :
			{ 260686004 |Method| = 410820007 |Surgical transplantation - action| , 405813007 |Procedure site - Direct| = 64033007 |Kidney structure| , 363701004 |Direct substance| = 420852008 |Kidney graft - material| }
			{ 260686004 |Method| = 410820007 |Surgical transplantation - action| , 405813007 |Procedure site - Direct| = 15776009 |Pancreatic structure (body| , 363701004 |Direct substance| = 421263007 |Pancreas graft - material| }
			{260686004 |Method| = 410820007 |Surgical transplantation - action| , 405813007 |Procedure site - Direct| = 9846003 |Right kidney structure| , 363701004 |Direct substance| = 420852008 |Kidney graft - material| }

5.3.2.2 Adding a Self-grouped Attribute

A self-grouped attribute is an attribute that, by editorial guidance, must be placed in an attribute group on its own. These are listed in Table 5.3.2.2-1.

Transformation Preconditions

This transformation will be applied to all loose attributes that meet the following conditions:

- The attribute applies to the MRCM domain of the focus concept
  - See Table 5.3.2.2-1
- Only one instance of the attribute is stated in the CTU expression
  - Although the MRCM allows multiple instances, this transformation condition aims to avoid misinterpretation of the intended meaning
- Either:
  - The attribute type is not already used in the necessary normal form of the focus concept
    - For example, if the focus concept already has a |Due to| attribute in the NNF then an expression including |Due to| would not be accepted
  - The attribute type is already used in the necessary normal form of the focus concept and the value is equal to or a subtype of the existing value

- I.e. it represents a refinement

Table 5.3.2.2-1: Self-grouped attributes and the domain they apply to.

Domain	Attribute
Clinical finding	before
	during
	after
	due to
	clinical course
	temporally related to
	associated with
Procedure	priority
	has focus

Please refer to the Editorial Guide for instructions on the use and the implied meaning of these attributes:

- Clinical Finding Defining Attributes
- Procedure Defining Attributes

As part of the expression transformation process, the terminology server will place the attributes in groups on their own, as shown in the examples below.

Table 5.3.2.2-2: Examples of CTU expressions candidate to this pattern, and their evaluation at level 1.

CTU expression	Accept/ reject	Notes	Classifiable form
84229001 |Fatigue| : 255234002 |After| = 840539006 |COVID-19|	Accept	The focus concept |Fatigue| is not already defined with an |After| attribute and the MRCM permits it to be added as an additional qualifier	=== 84229001 |Fatigue| : { 363714003 |Interprets| = 359755007 |Energy / stamina| } { 255234002 |After| = 840539006 |COVID-19| }
397181002 |Open fracture| : 42752001 |Due to| = 57168000 |Operation on bone|	Reject	The focus concept |Open fracture| is already defined with a |Due to| attribute, and the new value of 57168000 |Operation on bone| is not a subtype of 773760007 |Traumatic event| that is already used in the definition of that focus concept	N/A

CTU expression	Accept/ reject	Notes	Classifiable form
80146002 |Appendicectomy| : 260870009 |Priority| = 394849002 |High priority|	Accept	The focus concept |Appendicectomy| is not defined with a |Priority| attribute, and the MRCM permits it to be added as a new qualifier	=== 80146002 |Appendicectomy| : { 260686004 |Method| = 129304002 |Excision - action| , 405813007 |Procedure site - Direct| = 66754008 |Appendix structure| }, { 260870009 |Priority| = 394849002 |High priority| }

5.3.2.3 Adding Severity to Clinical Findings

At level 1, the terminology server will accept CTU expressions matching the following pattern:

```
[[+ (< 404684003 |Clinical finding| MINUS << 162465004 |Symptom severity| ) @finding]]:
   246112005 |Severity| = [[+ (<< 272141005 |Severities (qualifier value)| ) @severity]]
```

This means that CTU expressions that includes the severity attribute will be accepted if the following criteria are met:

- The focus concept is a subtype of |Clinical finding| (but not a subtype of 162465004 | Symptom severity (finding)|)
- The attribute value must be a subtype of 272141005 | Severities (qualifier value)|
- When applied as a loose attribute, only one instance of the attribute may be stated in the CTU expression
- The severity attribute is not already present in the definition of the focus concept

As part of the transformation process, the terminology server will place the attribute in a group on its own, similar to the self-grouped attributes.

Important Notes

Please be aware that caution should be taken when adding |severity| to a focus concept, because severities are difficult to generalize and may be subject to individual interpretation, as stated in the Editorial Guide.

Table 5.3.2.3-1: Examples of CTU expressions candidates to this pattern, and their evaluation at level 1.

CTU expression	Accept/ reject	Notes	Classifiable form

16331000 |Heartburn| : 246112005 |Severity| = 24484000 |Severe|	Accept	Focus concept is within the domain of the attribute and the attribute value is within the allowed range of the attribute.	=== 16331000 |Heartburn| : { 363698007 |Finding site (attribute)| = 32849002 |Esophageal structure (body structure)| }
			{ |Severity| = 24484000 |Severe| }

25064002 |Headache| : 246112005 |Severity| = 162471005 |Symptom very severe|	Reject	Attribute value is not within the allowed range for the |severity| attribute.	N/A

CTU expression	Accept/ reject	Notes	Classifiable form
717933005 |Severe thinness in adulthood| : 246112005 |Severity| = 442452003 |Life threatening severity|	Reject	Focus concept is already defined with a |severity| attribute, and the attribute value stated in the CTU is not a subtype of the value of the focus concept.	N/A

5.3.2.4 Lateralizing Clinical Findings

The terminology server will accept CTU expressions matching the following pattern:

```
[[+ (< 404684003 |Clinical finding| ) @finding]]:
   272741003 |Laterality| = [[+ (< 182353008 |Side| ) @side]]
```

The transformation of expressions following this pattern will include the following steps:

1. Copy the defining properties and role groups of the focus concept (as present in the NNF) to the classifiable form expression
2. Replace the value of the |Finding site| attribute with a nested expression:

From: @finding. 363698007 |Finding site| = @bodySite

To: 363698007 |Finding site| = (@bodySite: |Laterality| = [[+ @side]])

Important Notes

The terminology server will only accept expressions complying to this pattern if the focus concept of the CTU expression has the following characteristics:

- The focus concept
  - includes only one role group with a |Finding site| attribute **OR**
  - includes two or more role groups with a |Finding site| attribute **WHERE**
    - the value of the finding site attribute in each role group is the same concept
  - includes no attributes with values that are already lateralized anatomical structures (ie values whose own definitions already include a value for the 272741003|Laterality| attribute)
- The value of the finding site attribute is a member of the 723264001 | Lateralizable body structure reference set|
  - *being a member of this refset means that the associated body structure is lateralizable, and does not already state a laterality other than 182353008 |Side|*

Transformation of bilateral findings

When applying the concept 51440002 | Right and left| as the value for the 272741003 | Laterality| attribute, the transformation should be done in accordance with the modeling applied to the content in the International Edition of SNOMED CT.

This means that the transformation steps described above should be applied twice; one for each lateral half, as illustrated in the example below.

CTU	Classifiable form
301354004 |Pain of ear| : 272741003 |Laterality| = 51440002 |Right and left|	=== 301354004 |Pain of ear| : { 363698007 |Finding site| = (117590005 |Ear structure| : |Laterality| = 7771000 |Left|)}, { 363698007 |Finding site| = (117590005 |Ear structure| : |Laterality| = 24028007 |Right|)}

Examples

Table 5.3.2.4-1: Examples of CTU expressions and their evaluation at level 1

CTU expression	Accept/reject	Notes	Classifiable form
274663001 |Acute pain| : 272741003 |Laterality| = 7771000 |Left|	Reject	The modeling of |Acute pain| does not contain a |Finding site| attribute	N/A
21522001 |Abdominal pain| : 272741003 |Laterality| = 7771000 |Left|	Reject	The value of the |Finding site| attribute (i.e. 818983003 |Abdomen|) is not lateralizable	N/A
301354004 |Pain of ear| : 272741003 |Laterality| = 7771000 |Left|	Accept	The value of the |Finding site| attribute (i.e. 117590005 |Ear structure|) is lateralizable	=== 301354004 |Pain of ear| :{ 363698007 |Finding site| = (117590005 |Ear structure| : |Laterality| = 7771000 |Left|)}
274279008 |Renal pain| : 272741003 |Laterality| = 7771000 |Left|	Reject	The focus concept includes two groups with a |Finding site| attribute in each. The values of the |Finding site| attribute are not the same concept not do they subsume each other	N/A
16018431000119109 |Paresis of right lower limb| : 272741003 |Laterality| = 7771000 |Left|	Reject	The focus concept includes two groups with a |Finding site| attribute in each. The values of the |Finding site| attribute are not the same concept and one of the |Finding site| values is already lateralized while the other is not lateralizable.	N/A
288228002 |Myalgia/myositis - forearm| : 272741003 |Laterality| = 7771000 |Left| 288228002 |Myalgia/myositis - forearm (finding)|	Reject	The focus concept includes two groups with a |Finding site| attribute in each. Both of the two |Finding site| attribute values of the focus concept are lateralizable The attribute values are not the same concept.	=== 288228002 |Myalgia/myositis - forearm| : { 363698007 |Finding site| = (14975008 |Forearm structure| : 272741003 |Laterality| = 7771000 |Left|)} { 363698007 |Finding site| = (30608006 |Skeletal muscle structure of upper limb| : 272741003 |Laterality| = 7771000 |Left|)}
449702005 |Cellulitis and abscess of lower limb| : 272741003 |Laterality| = 7771000 |Left|	Accept	The values of the two |Finding site| attributes of the focus concept is the same concept and this is lateralizable	=== 449702005 |Cellulitis and abscess of lower limb| : { 363698007 |Finding site| = (61685007 |Lower limb structure| : 272741003 |Laterality| = 7771000 |

CTU expression	Accept/reject	Notes	Classifiable form

			Left|),
			116676008 |Associated morphology| = 385627004 | Cellulitis| }
			{ 363698007 |Finding site| = (61685007 |Lower limb structure| : 272741003 |Laterality| = 7771000 | Left|),
			116676008 |Associated morphology| = 44132006 | Abscess| }

5.3.2.5 Lateralizing Procedures

The terminology server will accept CTU expressions matching the following pattern:

```
[[+ (< 71388002 |Procedure| ) @procedure]]:
   272741003 |Laterality| = [[+ (< 182353008 |Side| ) @side]]
```

The transformation of expressions following this pattern will include the following steps:

1. Copy the defining properties and role groups of the focus concept (as present in the NNF) to the classifiable form expression
2. Replace the value of the |Procedure site| attribute (or any of its subtypes / @siteAttribute) with a nested expression:
   From: @siteAttribute = @bodySite
   To: @siteAttribute = (@bodySite: |Laterality| = [[+ @side]])

Important Notes

The terminology server will only accept expressions complying to this pattern where the focus concept has the following characteristics:

- The focus concept
  - includes only one role group with a @siteAttribute 363704007 |Procedure site| attribute (or any subtype of 363704007 |Procedure site|) **OR**
  - includes two or more role groups with a 363704007 |Procedure site| attribute (<< 363704007 | Procedure site|) **WHERE**
    - the @bodySite value is the same for all procedure site attributes
  - includes no attributes with values that are lateralized anatomical structures
- The value of the procedure site attribute is a member of the 723264001 | Lateralizable body structure reference set|
  - *i.e. the associated body structure is lateralizable, and*
  - *the associated body structure does not already state a laterality*

Transformation of bilateral procedures

When applying the concept 51440002 | Right and left| as the value for the 272741003 | Laterality| attribute, the transformation should be done in accordance with the modeling applied to the content in the International Edition of SNOMED CT.

This means that the transformation steps described above should be applied twice; one for each lateral half, as illustrated in the example below.

CTU	Classifiable form
52734007 |Total replacement of hip| : 272741003 |Laterality| = 51440002 |Right and left|	=== 52734007 |Total replacement of hip| : { 363698007 |Finding site| = (117590005 |Ear structure| : |Laterality| = 7771000 |Left|)}, { 363698007 |Finding site| = (117590005 |Ear structure| : |Laterality| = 24028007 |Right|)}

Examples

Table 5.3.2.5-1: Examples of CTU expressions and their evaluation at level 1.

CTU expression	Accept/ reject	Notes	Classifiable form
14600001000004107 |Closure of wound of ankle with flap| : 272741003 |Laterality| = 7771000 |Left|	Accept	The value of the | Procedure site - direct| attribute (i.e. 344001 | Ankle region structure|) is lateralizable	=== 14600001000004107 |Closure of wound of ankle with flap| : { 260686004 |Method| = 129357001 |Closure - action| , 405813007 |Procedure site - Direct| = (344001 |Ankle region structure| : 272741003 |Laterality| = 7771000 |Left|), 363700003 |Direct morphology| = 13924000 |Wound| , 424361007 |Using substance| = 256683004 |Flap| }
449647000 |Reduction of open fracture of lower leg with internal fixation| : 272741003 |Laterality| = 7771000 |Left|	Accept	The values of the two | Procedure site - direct| attributes of the focus concept is the same concept and this is lateralizable	=== 449647000 |Reduction of open fracture of lower leg with internal fixation| : { 260686004 |Method| = 129371009 |Fixation - action| , 405813007 |Procedure site - Direct| = (702468001 |Bone structure of lower leg| : 272741003 |Laterality| = 7771000 |Left|), 363700003 |Direct morphology| = 52329006 |Fracture, open| , 363699004 |Direct device| = 31031000 |Orthopedic internal fixation system, device| } { 260686004 |Method| = 129427006 |Reduction - action| , 405813007 |Procedure site - Direct| = (702468001 |Bone structure of lower leg| : 272741003 |Laterality| = 7771000 |Left|), 363700003 |Direct morphology| = 52329006 |Fracture, open| }
11971000224104 |Dissection of lymph node| : 272741003 |Laterality| = 7771000 |Left|	Reject	The value of the | Procedure site - direct| attribute (59441001 | Structure of lymph node|) is not lateralizable	N/A N A / N

CTU expression	Accept/ reject	Notes	Classifiable form
6471000179103 |Transplantation of kidney and pancreas| : 272741003 |Laterality| = 7771000 |Left|	Reject	The focus concept includes two groups with a |Procedure site - direct| attribute in each. The attribute values are not the same concept.	N/A N A / N
52734007 |Total replacement of hip| : 272741003 |Laterality| = 7771000 |Left| 288228002 |Myalgia/myositis - forearm (finding)|	Accept	The focus concept includes one group with a |Procedure site - Direct | attribute, and another group with a |Procedure site - Indirect| attribute. The value of the two different procedure sIte attributes are the same (i.e. 182201002 |Entire hip joint|) and this is lateralizable	=== 52734007 |Total replacement of hip| : { 260686004 |Method| = 425362007 |Surgical insertion - action| , 405814001 |Procedure site - Indirect| = (182201002 | Entire hip joint| : 272741003 |Laterality| = 7771000 |Left |), 363699004 |Direct device| = 304120007 |Total hip replacement prosthesis| } { 260686004 |Method| = 257903006 |Repair - action| , 405813007 |Procedure site - Direct| = (182201002 | Entire hip joint| : 272741003 |Laterality| = 7771000 |Left |)}
443682009 |Total replacement of right knee joint| : 272741003 |Laterality| = 7771000 |Left|	Reject	The value of the | Procedure site - direct| and the |Procedure site - indirect| attribute (i.e. 735264009 | Entire right knee joint|) is defined with a laterality	N/A

5.3.2.6 Adding Context to Clinical Findings

246090004 |Associated finding (attribute)| = 363358000 |Malignant tumor of lung (disorder)|,
408729009 |Finding context (attribute)| = 415684004 |Suspected (qualifier value)|,
408732007 |Subject relationship context (attribute)| = 410604004 |Subject of record (person)|,
408731000 |Temporal context (attribute)| = 410512000 |Current or specified time (qualifier value)| At level 2, the terminology server will accept expressions matching the following pattern:

At level 1, the terminology server will accept CTU expressions matching the following pattern:

```
[[<< 404684003 |Clinical finding| @finding]]:
  [[0..1]] 408729009 |Finding context| = [[+id(<< 410514004 |Finding context value| )@context]],
  [[0..1]] 408731000 |Temporal context| = [[+id(<< 410510008 |Temporal context value| )@time]],
  [[0..1]] 408732007 |Subject relationship context| = [[+id(<< 125676002 |Person| )@person]]
```

This pattern allows attributes applicable to the 413350009 | Finding with explicit context (situation)| domain to be used to add contextual information to clinical finding concepts.

Important Notes

The terminology server will only accept expressions complying to this pattern if the focus concept of the CTU expression has the following characteristics:

- The focus concept
  - is the concept 404684003 |Clinical finding| or a subtype hereof
- For each expression, only one instance of each attribute is accepted
- The value of the attribute must be within the allowed range of the attribute as specified by the MRCM

Examples of CTU expressions and their evaluation at level 2 Examples of CTU expressions and their evaluation at level 2

The generated classifiable form must be based on the following expression with 413350009 | Finding with explicit context (situation)| as the focus concept and the following default attributes and values unless otherwise specified by the CTU. The value of the Associated finding attribute will be replaced by the focus concept of the CF expression.

```
=== 413350009 |Finding with explicit context| :
  { 246090004 |Associated finding| = 404684003 |Clinical finding| ,
    408729009 |Finding context| = 410515003 |Known present| ,
    408732007 |Subject relationship context| = 410604004 |Subject of record| ,
    408731000 |Temporal context| = 410512000 |Current or specified time| }
```

Table 5.3.2.6-1: Examples of CTU expressions and their evaluation at level 1.

Input expression	Accept/ reject	Notes	Classifiable form
266987004 |History of cancer| : 408732007 |Subject relationship context| = 72705000 |Mother|	Reject	The focus concept is not a subtype of 404684003 |Clinical finding|, and the expression, therefore, does not match this pattern. Furthermore, the |Subject relationship context| of the focus concept is 410604004 |Subject of record|, and the concept 72705000 |Mother| can, therefore, not be used to refine the meaning of the focus concept.	N/A
363358000 |Lung cancer| : 408729009 |Finding context| = 415684004 |Suspected|	Accept		=== 413350009 |Finding with explicit context| : { 246090004 |Associated finding| = 363358000 |Lung cancer| , 408729009 |Finding context| = 415684004 |Suspected| , 408732007 |Subject relationship context| = 410604004 |Subject of record| , 408731000 |Temporal context| = 410512000 |Current or specified time| }
254837009 |Breast cancer| : 408731000 |Temporal context| = 410513005 |Past| , 408732007 |Subject relationship context| = 72705000 |Mother|	Accept		=== 413350009 |Finding with explicit context| : { 246090004 |Associated finding| = 254837009 |Breast cancer| , 408729009 |Finding context| = 410515003 |Known present| , 408732007 |Subject relationship context| =

Input expression	Accept/ reject	Notes	Classifiable form
			72705000 |Mother| , 408731000 |Temporal context| = 410513005 |Past| }

5.3.2.7 Adding Context to Procedures

At level 1, the terminology server will accept CTU expressions matching the following pattern:

```
[[<< 71388002 |Procedure| @procedure]]:
  [[0..1]] 408730004 |Procedure context| = [[+id(<< 288532009 |Context values for actions| ))@context]],
  [[0..1]] 408731000 |Temporal context| = [[+id(<< 410510008 |Temporal context value| )@time]],
  [[0..1]] 408732007 |Subject relationship context| = [[+id(<< 125676002 |Person| )@person]]
```

This pattern allows attributes applicable to the 29125009 | Procedure with explicit context (situation)| domain to be used to add contextual information to procedures.

Important Notes

The terminology server will only accept expressions complying to this pattern if the focus concept of the CTU expression has the following characteristics:

- The focus concept
  - is the concept 71388002 |Procedure| or a subtype hereof
- For each expression, only one instance of each attribute is accepted
- The value of the attribute must be within the allowed range of the attribute as specified by the MRCM

The generated classifiable form must be based on the following expression with 129125009 | Procedure with explicit context (situation)| as the focus concept and the following default attributes and values, unless otherwise specified by the CTU. The value of the Associated procedure attribute will be replaced by the focus concept of the CF expression.

```
=== 129125009 |Procedure with explicit context| :
  { 363589002 |Associated procedure| = 71388002 |Procedure| ,
    408730004 |Procedure context| = 385658003 |Done| ,
    408732007 |Subject relationship context| = 410604004 |Subject of record|
    408731000 |Temporal context| = 410512000 |Current or specified time| }
```

Table 5.3.2.7-1: Examples of CTU expressions and their evaluation at level 1

Input expression	Accept/ reject	Notes	Classifiable form
42125001 |Excisional biopsy of breast mass| : 408730004 |Procedure context| = 410525008 |Needed|	Accept		=== 129125009 |Procedure with explicit context| : { 363589002 |Associated procedure| = 42125001 |Excisional biopsy of breast mass| , 408730004 |Procedure context| = 410525008 |Needed| , 408732007 |Subject relationship context| = 410604004 |Subject of record| 408731000 |Temporal context| = 410512000 |Current or specified time| }
74400008 |Appendicitis| : 408730004 |Procedure context| = 443390004 |Refused|	Reject	Focus concept is not a subtype of << 71388002 |Procedure|	N/A

5.4 Implementation Examples

The following pages provides examples of how specific terminology servers support the implementation of expression repositories.

5.4.1 Expression Repository in RF2

Scope

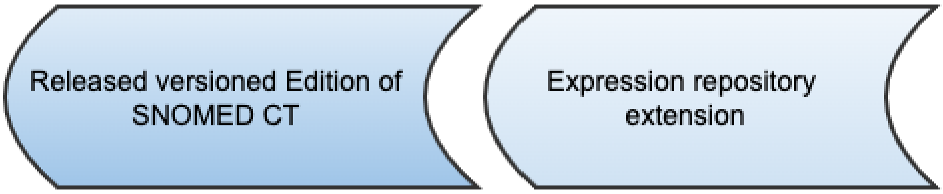
This part of the document introduces the RF2 approach to implementation of an expression repository, and this approach is supported by SNOMED international's terminology server, Snowstorm.

Under development

This section is still under active development, and the documentation may be updated as reference implementations and alignment with the specification mature.

Expression Repository Extension

Implementing an expression repository using the standard format specified by SNOMED International is comparable to the implementation of a SNOMED CT extension. As for SNOMED CT extensions, the content contained in the expression repository depends on a specific versioned Edition of SNOMED CT, and it requires ongoing maintenance to align with new versions of SNOMED CT.


Released versioned Edition of SNOMED CT
Expression repository extension

The difference between a common SNOMED CT Extension and an Expression Repository Extension can be summarized as follows:

Difference in the representation of clinical meanings

- Clinical meanings expressed in a common extension are represented as **SNOMED CT concept components** and require creation and management by SNOMED CT authors

- Clinical meanings expressed in an expression repository extension are represented as **SNOMED CT expression components**

Difference in the content included

- A common SNOMED CT Extension may contain various components and derivatives (see 4.3 Extensions)
- An expression repository extension contains only the components and derivatives required to represent postcoordinated expressions
  Consequently, the following components and derivatives may be contained to meet the requirements outlined in 5.1 Requirements:
  - Expression components to uniquely identify the expressions
  - A reference set to hold the CTU expressions
  - A reference set to hold the CF expressions
  - Expression relationships to represent the NNF of the expressions
  - A component annotation reference set to hold the display terms associated with the expression
- Release process
  - Extensions are released
  - Expressions are not released

Logical Model

Figure 5.4.1-1 illustrates the logical model of an expression repository extension.

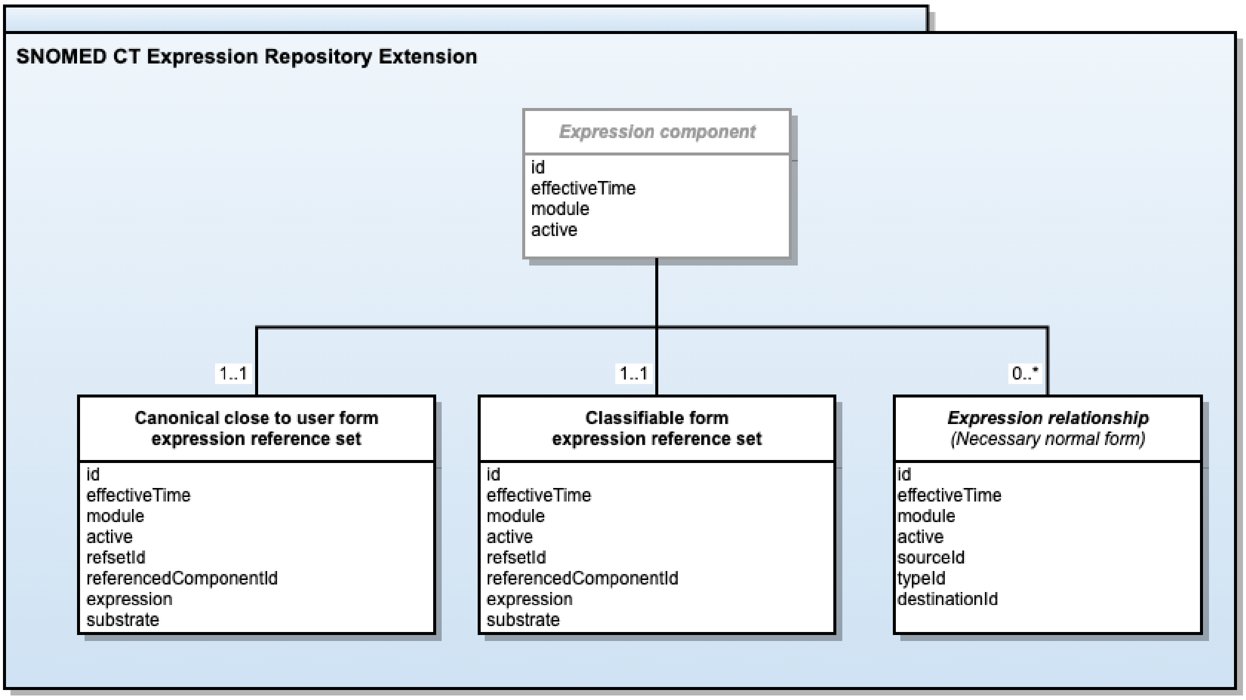


Figure 5.4.1-1: Overview of the logical model of a SNOMED CT Expression Repository

Expression Component

Implementing an expression repository involves the creation of expression components, where the expression identifier uses the partition id of '16'. For more information about SNOMED CT identifiers, please refer to the release file specification chapter 6 SNOMED CT Identifiers.

This identifier enables the unique identification of an expression and is used to establish a link between all required forms of the expression.

- The expression identifier is:
  - Used as the id for the expression component
  - Referenced in the 1119435002 | Canonical close to user form expression reference set (foundation metadata concept)| as the value of the referencedComponentId
  - Referenced in the 1119468009 | Classifiable form expression reference set (foundation metadata concept)| as the value of the referencedComponentId
  - Used as the source of the expression relationships to represent the necessary properties of the expression

The moduleId attribute of the expression component refers to a concept that represents and names the module in which the expressions are maintained. This concept is a subtype of 900000000000443000 | Module| , and the terms associated with this concept should clearly indicate that this module is an expression repository, e.g. 1234567890 | national extension expression repository module| .

Modules

For more information about SNOMED CT Modules, please refer to the Extensions Practical Guide, chapter 4.2 Modules.

Canonical Close to User Form Expression Reference Set

The CTU is stored in a separate reference set 1119435002 | Canonical close to user form expression reference set (foundation metadata concept)| which is of the type Postcoordinated Expression Type Reference Set .

Expressions in the close-to-user form will get an effectiveTime as soon as they are saved, this will match that of the current dependent code system version.

Classifiable Form Expression Reference Set

The CF is stored in a separate reference set 1119468009 | Classifiable form expression reference set (foundation metadata concept)| which is of the type Postcoordinated Expression Type Reference Set .

Members in the classifiable form reference set are given an effectiveTime. This will happen before the code system is upgraded to a new dependent version. Once these components are versioned they will keep the effectiveTime until the values of their fields are changed, via the transformation or classification processes. At this point changed components will have their effectiveTime cleared and will be versioned again when the code system is next versioned.

Expression Relationship

Following the classification of the CF expression, the NNF of the expression is represented as relationships. In the same way as SNOMED CT concepts, expression components include subtype and attribute relationships to represent the properties of the expression.

Components of expressions in the classifiable and necessary normal forms will have an effectiveTime set when the expression code system is versioned. This will happen before the code system is upgraded to a new dependent version. Once these components are versioned they will keep the effectiveTime until the values of their fields are changed, via the transformation or classification processes. At this point changed components will have their effectiveTime cleared and will be versioned again when the code system is next versioned.

5.4.2 Expression Repository with HL7 FHIR

Scope

This page summarizes the implementation of an Expression Extension using the HL7 FHIR Terminology Services API.

This approach is in the process of being implemented and supported by the Snowstorm FHIR Terminology Services API.

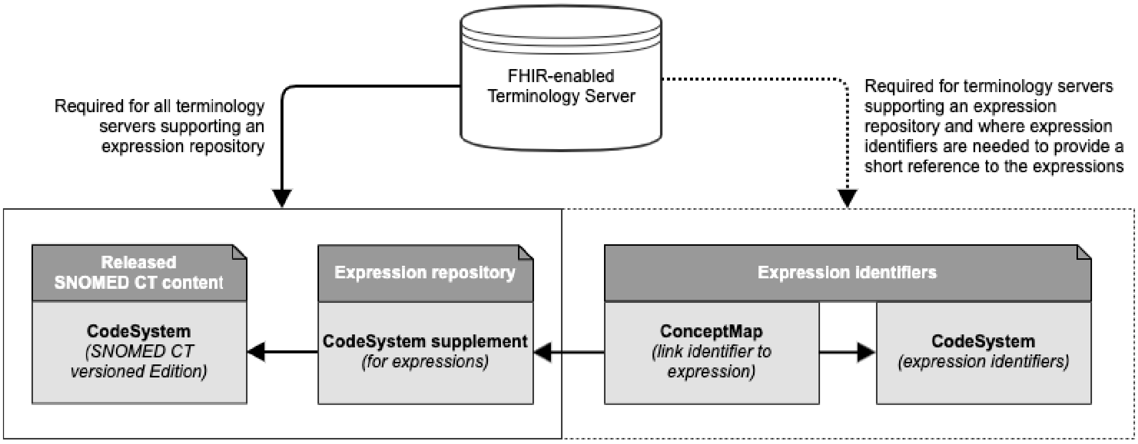
Under development

This section is still under active development, and the API examples will change as reference implementations mature and this is discussed with the community of practice.

Solution Overview

To represent an expression repository using FHIR terminology services, the required solution will depend on what approach is taken to uniquely refer to the expressions:

- The expressions themselves may be used as an identifier
  - With this approach, a **CodeSystem Supplement** will be created and function as the expression repository
    - This supplements the applied SNOMED CT CodeSystem
- A separate identifier is created for each expression
  - With this approach, the following FHIR terminology resources are required
    - a CodeSystem Supplement to function as the expression repository
      - This supplements the applied SNOMED CT CodeSystem
    - a CodeSystem to hold the expression identifiers
    - a ConceptMap to hold the association between the expression identifiers and the expressions


FHIR-enabled Terminology Server
Required for all terminology servers supporting an expression repository
Required for terminology servers supporting an expression repository and where expression identifiers are needed to provide a short reference to the expressions
Released SNOMED CT content
CodeSystem (SNOMED CT versioned Edition)
Expression repository
CodeSystem supplement (for expressions)
Expression identifiers
ConceptMap (link identifier to expression)
CodeSystem (expression identifiers)

Services

With this implementation, the defined operations for CodeSystems and ConceptMaps can be utilized to support the services required for an expression repository. Please refer to the HL7 FHIR documentation for further details:

- **CodeSystem**
  - $lookup
  - $validate-code
  - $subsumes
- **ConceptMap**
  - $translate
  - $closure

Create expression repository

Creating an expression repository using a FHIR Terminology Server involves the creation of the following structures:

- A CodeSystem supplement (*expRep*): This functions as the actual expression repository and is used to hold the expressions
- A CodeSystem (*expId*): This code system is used to hold the unique expression identifiers
- A ConceptMap: This is used to map between the expressions in the expRep CodeSystem supplement and the expId CodeSystem. The ConceptMap enables the translation of expressions to (and from) shorter codes for systems with restricted code lengths

Click to show/hide overview of the workflow...

The workflow for populating an expression repository using a FHIR Terminology Server is illustrated in

Error rendering macro 'caption-reference'

Error occurred rendering template content

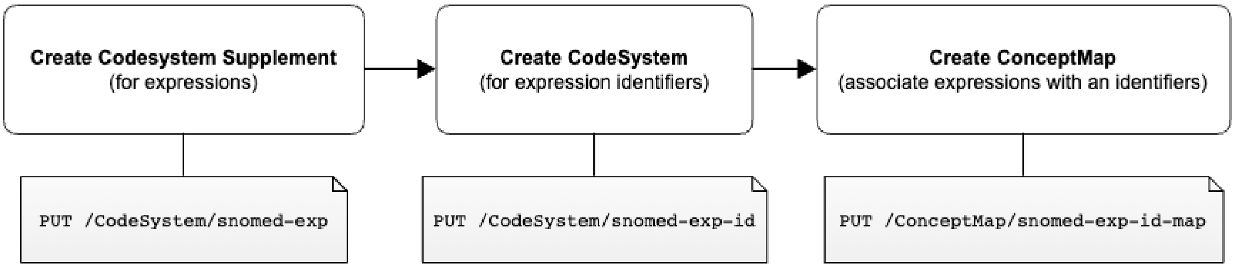

Create Codesystem Supplement
(for expressions)
Create CodeSystem
(for expression identifiers)
Create ConceptMap
(associate expressions with an identifiers)
PUT /CodeSystem/snomed-exp
PUT /CodeSystem/snomed-exp-id
PUT /ConceptMap/snomed-exp-id-map

Figure 5.4.2-1: -

Service	Description	Input

Create Code System Supplement
 	
POST / CodeSystem
	Create the CodeSystem supplement.
	This operation must specify a version URI that uses the snomed.info xsct URI with a specific module but no version.
	This resource must supplement a specific version of a SNOMED CT edition.
	The example supplements the International Edition of SNOMED CT (900000000000207008), version Jan 2023 (20230131).
		`{`
		`    "resourceType": "CodeSystem",`
		`    "url": "http://snomed.info/info",`
		`    "version": "http://snomed.info/xsct/11000003104"`
		`,`
		`    "content": "supplement",`
		`    "supplements" : "http://snomed.info/sct|http://snomed.info/sct/900000000000207008/version/20230131"`
		`}`

Service	Description	Input

Create SNOMED Exp IDs Code System
POST / CodeSystem
(Only needed if EHR cannot store expressions)
	If needed, create a CodeSystem for expression identifiers.	`{`
		`    "resourceType": "CodeSystem",`
		`    "url": "http://snomed.info/snomed/exp-id/1000003",`
		`    "status": "active",`
		`    "content": "complete"`
		`}`

Create Concept Map for Exp Ids
POST / ConceptMap
(Only needed if EHR cannot store expressions)
	Create the ConceptMap to map expressions in the code system supplement to expression identifiers in the Exp. Ids code system. This will be populated manually.	`{`
		`    "resourceType": "ConceptMap",`
		`    "url": "http://snomed.info/snomed/exp-id-map",`
		`    "status": "active",`
		`    "source": {`
		`        "sourceUri": "http://snomed.info/xsct/1234007?fhir_vs"},`
		`    "target": {`
		`        "targetUri": "http://snomed.info/snomed/exp-id/1101234?fhir_vs"}`
		`}`

Get repository details

Service	Description	Input
GET /CodeSystem/{id}	Get the details of the Code System using the provided ID; it is the same operation for the supplement or the Exp. IDs code system.	`Code System ID, provided in the response of the code system creation`

Add expression

Service	Description	Input

PATCH /CodeSystem/{id}	Adds an expression in the Code System Supplement using a patch operation.
	The display value can be generated from the expression, or the expression itself could be used in this field.
	This operation should run validation on the expression being inserted into the supplement.
		`[`
		`  {`
		`    "op": "add",`
		`    "path": "/concept",`
		`    "value": {`
		`      "code": "359817006 |Closed fracture of hip (disorder)| :`
		`       272741003 |Laterality (attribute)| = 182353008 |Side (qualifier value)|",`
		`      "display": "Right closed fracture of hip"`
		`    }`
		`  }`
		`]`

(Source: https://fhirblog.com/2019/08/13/updating-a-resource-using-patch/)

Lookup expression

Service	Description	Input	Output
GET /CodeSystem/{supplementId}/$lookup?code={ `CTU` }** **GET /CodeSystem//$lookup? `system={versionUri} &code={CTU}`	Lookup Expression in the supplement	• versionUri • CTU: close-to-user expression	
GET /ConceptMap/{ `snomed-exp-id-map}/ $translate? system={versionUri} &code={CTU}` `Example` *GET /ConceptMap/* `snomed-exp-id-map/ $translate?system=http://snomed.info/xsct/1234007 &code=87971000:272741003=7771000`	Lookup Expression Identifier from Expression	• versionUri • CTU: close-to-user expression	
GET /ConceptMap/{ `snomed-exp-id-map}/ $translate? reverse=true &system={versionUri} &code={expId}` `Example` *GET /ConceptMap/* `snomed-exp-id-map/ $translate? reverse=true &system= http://snomed.info/snomed/exp-id/1101234 &code=101101234165`	Lookup Expression from Expression Identifier	• versionUri • expId: expression identifier as defined pr. implementation	

Search for expression

Use an implicit value set with ECL.

Get display term

A display term should be returned in the response of the lookup operation. The display term may be generated in a very simple way by just concatenating.

Create Classifiable Form

This should happen when the expression is inserted into the code system.

Create Necessary Normal Form

This should happen when the expression is inserted into the code system.

Subsumption

Service	Description	Input
Expression subsumption test **GET /CodeSystem/$subsumes**	Test the subsumption between a pair of expressions or an expression and a code.	`{ "resourceType": "CodeSystem", "url": "http://snomed.info/snomed/exp-id/1234007", "status": "active", "content": "supplement", "supplements": "http://snomed.info/snomed/exp-id/9000000000000207008", "valueSet": "http://snomed.info/snomed/exp-id/1234007?fhir_vs" }`

ECL

Service	Input	Output
• **GET /ValueSet/$expand?url={url}** `?fhir_vs=ecl/{ecl}`	• url: The URI of the expression repository • ecl: A valid expression constraint defining the constraints of the query	

Example

A hospital department would like to report on the number of closed reduction procedures performed. All codes representing this type of procedure must be fetched so that they can be matched against the EHR records.

- Expand an intentional ValueSet using the expression library (and the Code System it supplements) with ECL "< 86052008 |Closed reduction of fracture (procedure)|"
  - **GET /ValueSet/$expand?url=** `http://snomed.info/xsct/1234007?fhir_vs=ecl/<86052008`
- Expand an intentional ValueSet using the expression library, the Code System it supplements and a derivative (module 98763002)
  - **GET /ValueSet/$expand?url=** `http://snomed.info/xsct/1234007;module/98763002?fhir_vs=ecl/<86052008`

Appendixes

Appendix A: Techniques for Autogenerating Display Terms

Three different approaches to automatically creating display terms for postcoordinated expressions are presented, ranging from the simplest to the more complex.

1. Replace concept identifiers with terms
2. Apply simple term rules
3. Apply description templates and rules

Please note that alternative, and more advanced approaches are possible.

When deciding what approach to apply, the following questions need to be considered:

- How close does the term need to match the spoken clinical language?
- Are resources available to develop and maintain the transformation rules and templates?

Replace Concept Identifiers with Terms

The simplest approach to generating display terms for a postcoordinated expression is to simply replace the concept identifiers with a term or description associated with the concept (preferably the preferred term of the applied language reference set). The result of this transformation leads to an expression that doesn't match the clinical language, but it may still be sufficient for the end-users to correctly interpret the semantics of the expression. The table below shows examples of this approach.

Expression	Approach	Display term
397181002:363698007=23416004	• Concept identifiers are replaced with the preferred term of each concept *(in the US language reference set)*	"|Open fracture|:|Finding site|=|Bone structure of ulna|"
336863008:272741003=7771000		"|Excision of cyst of lung|:|Laterality|=|Left|"

Simple Term Rules

To increase the human readability of the expressions, simple term rules can be applied in conjunction with the replacement of concept identifiers. Part of this involves avoiding the use of compositional grammar symbols in the terms, which can be achieved in different ways, e.g.

- Replace symbols with predefined terms
- Remove symbols and replace attributes with a ","

The table below exemplifies the transformation rules that can be used to replace compositional grammar symbols.

Symbol	Term
:	with a / with an
=	of
,	and
+	and

Examples of this approach are illustrated in the table below:

Expression	Approach	Display term
397181002:363698007=23416004	• Concept identifiers are replaced with the preferred term of each concept *(in the US language reference set)* • Compositional grammar symbols are replaced with a predefined term	"open fracture with a finding site of bone structure of ulna"
336863008:272741003=7771000		"excision of cyst of lung with a laterality of left"
397181002:363698007=23416004	• Concept identifiers are replaced with the preferred term of each concept *(in the US language reference set)* • Compositional grammar symbols are removed and attributes are replaced with a ","	"open fracture, ulna"
336863008:272741003=7771000		"excision of cyst of lung, left"

Description Templates

In some situations, description templates may be applied to ensure that the display terms of the expressions match the clinically spoken language more closely. This approach may be particularly appropriate to consider in situations where the postcoordinated expressions were originally generated using an expression template. In these cases, a corresponding description template and associated term rules may be developed more easily than when the expression was built by another method.

Table Appendix A:-1 illustrates the expression template, description template and term rules for creating expressions that represent a disorder with a fracture morphology and a finding site of some bone structure. The description template specifies that the term associated with the morphology concept is always followed by the term associated with the bone structure concept, and these two terms should be separated by the term ‘of’. Associated with this description template is a set of term rules, but please note that in a real implementation, additional rules may be included to refine the transformation and ensure a proper transformation in all cases (for more information, see general rules for generating descriptions for templates).

<table>
<tr><th>Expression template</th><th>Description template</th><th>Example term rules</th></tr>
<tr><td rowspan="2">[[+id(<< 64572001 |Disease| @disorder]]:
[[1..*]] {
[[1..1]] 116676008 |Associated morphology| = [[+id(<< 72704001 |Fracture| @fractureMorphology]],
[[1..1]] 363698007 |Finding site| = [[+scg (<< 272673000 |Bone structure|) @boneStructure]]
}</td><td rowspan="2">[[$fractureMorphology]] of [[$boneStructure]]</td><td>$boneStructure
• Use the FSN
• Remove the semantic tag (body structure)
• Remove prefix, such as:
▪ 'Bone structure of'
▪ 'Structure of'
• Remove postfix, such as:
▪ 'structure'</td></tr>
<tr><td>$fractureMorphology
• Use the FSN
• Remove the semantic tag (morphologic abnormality)
• Remove the "," separating the morphologies, if two appear, and swap the terms</td></tr>
</table>

Table Appendix A:-1: Example of Expression template, description template and transformation rules for expressions representing fractures.

Table Appendix A:-2 provides two examples of how a description template and a few term transformation rules can lead to human-readable display terms.

#	Expression slots		Description template	Term transformation	Display term
1	**@fractureMorphology**	|fracture, open (morphologic abnormality)|	[[**$fractureMorphology**]] of [[**$boneStructure**]]	fracture~~,~~ open ~~(morphologic abnormality)~~ → **open fracture**	"open fracture of ulna"
	@boneStructure	|bone structure of ulna (body structure)|		~~bone structure of~~ ulna ~~(body structure)~~ → **ulna**	
2	**@fractureMorphology**	|fracture, closed (morphologic abnormality)|		fracture~~,~~ closed ~~(morphologic abnormality)~~ → **closed fracture**	“closed fracture of distal ulnar epiphysis“
	@boneStructure	|structure of distal ulnar epiphysis (body structure)|		~~structure of~~ distal ulnar epiphysis ~~(body structure)~~ → **distal ulnar epiphysis**	

Table Appendix A:-2: Examples of description templates applied in conjunction with term transformation rules