SNOMED International

Leading healthcare terminology, worldwide

SNOMED CT Compositional Grammar Specification and Guide

Version 2.4

Publication date: 2020-06-17

Web version link: http://snomed.org/scg

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

Table of Contents

Leading healthcare terminology, worldwide

The Compositional Grammar Specification and Guide specifies a standard syntax for representing individual clinical meanings using either precoordinated or postcoordinated expressions.

Web browsable version: http://snomed.org/scg

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

1 Introduction

Background

SNOMED CT is a clinical terminology with global scope covering a wide range of clinical specialties and requirements. The use of SNOMED CT expressions in Electronic Health Records (EHRs) provides a standardized way to represent clinical meanings captured by clinicians and enables the automatic interpretation of this meaning. SNOMED CT expressions are a structured combination of one or more concept identifiers used to represent a clinical idea in a logical manner.

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

History

The SNOMED CT Composition Grammar was initially specified as part of the document "*SNOMED Clinical Terms Abstract Logical Models and Representational Forms, External Draft for Comment Version"*. This version (v1) was used extensively as both a human readable and machine parsable syntax.

A revised version (v2.0 to v2.2) of this specification was adopted as an IHTSDO standard in 2010. This version followed the prior version in most details, with the following enhancements:

- The specification of the grammar was defined in Augmented Backus-Naur Form (ABNF)[1] . This provided a formal standards-based reference for the grammar's structure.
- Unnecessary whitespace designators (i.e. <ws>) were removed from several places in the grammar.
- A maximum length constraint for SNOMED CT identifiers (SCTIDs) was added. SCTIDs consist of a sequence of digits, which must be between 6 and 18 digits in length.
- The hex code for carriage return (CR) was incorrectly given as '0C' in the previous version. It was corrected to '0D'.
- Detailed character encoding information for UTF-8 was added.
- The ABNF definition of term was amended to allow correct parsing by the ABNF parser generator.

In 2015, the original IHTSDO standard was revised (v2.3 and v2.3.1). The new specification is completely backwardly compatible with the prior standard, in that any expression written using the prior standard *will* necessarily conform to the new standard. However, two significant enhancements were made:

- Concrete values (e.g. integers, decimals and strings) are now permitted as attribute values.
- A definition status may now (optionally) be included at the start of an expression to indicate whether the clinical idea being expressed is 'equivalent to' or a 'subtype of' the expression.

In 2020, concrete attribute values were extended to permit booleans (v2.4).

These enhancements will be described and explained further within this guide.

Purpose

The purpose of this document is to define and describe a formal compositional grammar which is used to represent SNOMED CT expressions. SNOMED CT expressions are a structured combination of one or more concept identifiers used to express a clinical idea. SNOMED CT expressions may exist either independently or within the scope of an information model, message instance, EHR system or clinical repository. This document also provides examples and guidance to assist in the implementation of this syntax.

Scope

This document presents the specification of a compositional grammar, which can be used to represent SNOMED CT expressions. This document also provides a logical model for the grammar, discusses a set of example expressions and describes some implementation considerations.

The revised compositional grammar specified in this document is the first of a consistent set of computer processable languages designed to meet a broader set of requirements related to the use of SNOMED CT. Other SNOMED CT computable languages, which are either available or under development include:

- Expression Constraint Language: used to define *a bounded set of* clinical meanings represented by either precoordinated or postcoordinated expressions;
- **Query Language**: designed to express computable queries over SNOMED CT content; and
- Templates: which allow slots to be added to expressions, expression constraints or queries, which can be filled with specific values at a later time.

SNOMED CT Compositional Grammar is designed to provide a common foundation for the additional functionality added by these other languages. However, the compositional grammar itself does not provide this added functionality.

This document does not include a full description of how to implement a compositional grammar parser, classifier or interpreter. It also does not describe how to implement an EHR which uses compositional grammar to represent clinical ideas. Instead, it provides general guidance to assist in the implementation of compositional grammar in any of these applications.

Audience

The target audiences of this document include:

- SNOMED National Release Centers;
- SNOMED CT designers and developers, including 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;
- SNOMED CT terminology developers, including concept model designers, content authors, map developers and release process managers.

Document Overview

This document defines, describes and provides implementation guidance for the use of SNOMED CT Compositional Grammar. Chapter 2 begins by describing the use cases in which it is anticipated that SNOMED CT compositional grammar will be used. Chapter 3 then describes the requirements used to guide the definition of this language. In Chapter 4, the logical model of compositional grammar is presented, while in Chapter 5 the syntax is defined, in terms of an ABNF serialisation of the logical model. Chapter 6 then presents some examples of expressions that conform to SNOMED CT compositional grammar, and Chapter 7 discusses some implementation considerations. Finally, Appendix A explains how to represent concept definition and expression relationship statements.

Glossary

The following table contains the definition of terms used within this document. Please refer to the SNOMED Glossary for additional definitions.

Term	**Definition**
Augmented Backus-Naur Form	A language used to define the formal syntax of another language (as defined by Internet Standard 68, RFC 5234).
Compositional Grammar	The set of rules that govern the way in which SNOMED CT expressions are represented as a plain text string.
Concept Model	A set of rules that determines the permitted sets of relationships between particular types of concepts.

Term	Definition
Equivalent To (definition status)	Indicates that the given expression is both necessary and sufficient to fully define the clinical meaning being expressed.
Expression	A structured combination of one or more concept identifiers used to express a clinical idea.
Expression Constraint	A computable rule that can be used to define a set of clinical meanings.
Machine Readable Concept Model	A representation of the rules that comprise the SNOMED CT Concept Model in a form that can be processed by computer software and applied to validate content.
Postcoordinated Expression	Representation of a clinical meaning using a combination of two or more concept identifiers is referred to as a postcoordinated expression.
Precoordinated Expression	Representation of a clinical meaning using a single concept identifier is referred to as a precoordinated expression.
Subtype Of (definition status)	Indicates that the given expression is necessary but not necessarily sufficient to define the clinical meaning being expressed.

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

2 Use Cases

SNOMED CT Compositional Grammar provides a syntax that enables clinical ideas to be represented, even when a single SNOMED CT concept does not capture the required level of detail. This is important as 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. Application software that supports the use of SNOMED CT expressions enables detailed clinical information to be recorded, retrieved and analysed. The logic on which the SNOMED CT concept model is based allows alternative representations of the same or similar information to be recognised and compared.

Clinical expressions using SNOMED CT concepts can be of two types: precoordinated expressions, which use a single SNOMED CT concept identifier; and postcoordinated expressions, which contain more than one SNOMED CT identifier. Postcoordination greatly increases the depth of detail that SNOMED CT can represent, while avoiding a combinatorial explosion of precoordinated concepts.

In the following sections, we describe four of the main use cases for SNOMED CT Compositional Grammar, including:

- Expressions in health records;
- Expressions in messages;
- Precoordinated concept definitions; and
- Expression associations between SNOMED CT and LOINC.

2.1 Expressions In Health Records

Clinical information recorded in EHRs using SNOMED CT is commonly represented using identifiers that refer to precoordinated concepts. These concepts may either be defined in the SNOMED CT international release or in an appropriate SNOMED CT extension. There are, however, times when a clinician needs to record and share a clinical meaning, which has not been defined in any release of SNOMED CT. In these situations, SNOMED CT postcoordinated expressions can be used to represent a new clinical meaning using the compositional grammar syntax.

2.2 Expressions In Messages

The SNOMED CT Compositional Grammar allows SNOMED CT expressions to be represented as a text string that can be carried in messages. For example, in HL7 version 3 messages SNOMED CT expressions can be placed in the 'code' property of the ConceptDescriptor (CD) data type (Release 2). The compositional grammar described in this document is intended to replace the qualifier mechanism that formerly was in the CD data type (Release 1), and which was removed in Release 2.

In September 2009, the HL7 Version 3 Standard "Data Types - Abstract Specification, Release 2" was published for its fourth normative membership ballot. This revised standard defined what can be carried in the 'code' property of the CD data type as shown below in Table 1.

Table 1. Definition of the 'code' property of the Concept Descriptor (CD) data type from HL7 Data Types Release 2
Code (code) : ST.SIMPLE **Definition:** The plain code symbol defined by the code system, or an expression **in a syntax defined by the code system which describes the *concept*.** *(emphasis added)* If provided, the code SHALL be an exact match to a plain code symbol or expression defined by the codeSystem. If the codeSystem defines a code or expression that includes whitespace, the code SHALL include the whitespace. **An expression can only be used where the codeSystem either defines an expression syntax, or there is a generally accepted syntax for the codeSystem**. *(emphasis added)*

The syntax described in this document is intended to satisfy the need for a "syntax defined by the code system" as stated above, when the "code system" is SNOMED CT. For specific guidance on using SNOMED CT Compositional

Grammar in the CD data type (Release 2), please refer to Appendix B.4 of "HL7 Version 3 Implementation Guide: TermInfo – Using SNOMED CT in CDA R2 Models, Release 1".

2.3 Precoordinated Concept Definitions

SNOMED CT Compositional Grammar may also be used to state the definition of precoordinated concepts. The introduction of definition status to this version of the language makes this usage more feasible. In particular, the definition of a fully defined concept uses an 'equivalent to' definition status to indicate that the expression is necessary and sufficient to define the clinical meaning of the concept. This is the same as giving the concept a definition status of 900000000000073002 |Defined|. The definition of a primitive concept uses a 'subtype of' definition status to indicate that the expression is necessary but not necessarily sufficient to define the clinical meaning of the concept. This is the same as giving the concept a definition status of 900000000000074008 |Primitive |.

For example, 31978002 |Fracture of tibia|(a fully defined precoordinated concept) can be defined using a compositional grammar expression as being 'equivalent to' an 428881005 |Injury of tibia| and a 46866001 |Fracture of lower limb|, with an 116676008 |Associated morphology| of 72704001 |Fracture| and a 363698007 |Finding site| of 12611008 |Bone structure of tibia|. This definition is shown below in Figure 1 using the SNOMED CT Diagramming Guidelines.

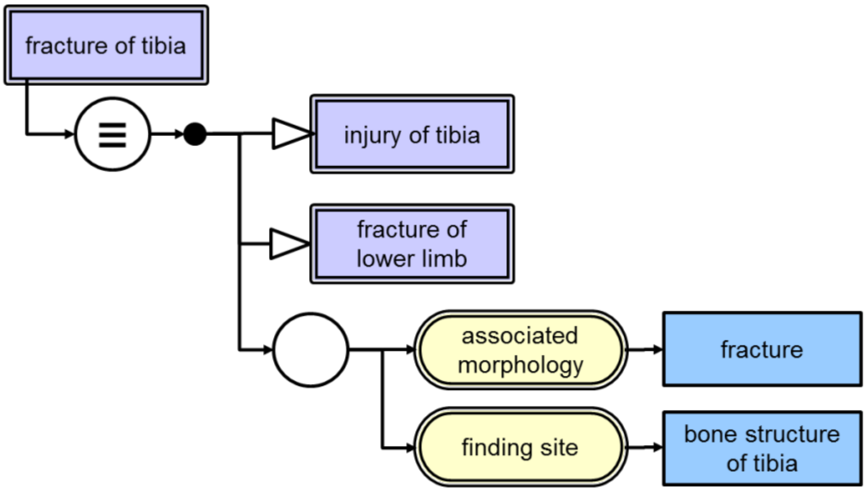


Figure 1: Definition of Precoordinated Concept

Please note that SNOMED CT expressions are used to represent a single clinical idea, rather than stating the equivalence between two clinical ideas. For this reason, a concept definition written as a SNOMED CT expression includes only the definition itself, without reference to the concept being defined. For an extension to the grammar which includes both the concept and its definition together, please refer to Appendix A.

2.4 Expression Associations With LOINC

LOINC® is a code system for identifying laboratory tests and other observations. It specifies universal codes, names and other attributes for laboratory results as well as clinical reports, physical exam findings, survey instruments and other observations. It was developed to enable the exchange and pooling of results from diverse sources in order to enhance clinical care, outcomes management and research. LOINC is owned, maintained and licensed by the Regenstrief Institute Incorporated (RII).

The Cooperation Agreement between IHTSDO and RII dated July 2013 states the cooperative work that is to be performed, including the creation of maps, expression associations and content alignments between SNOMED CT and LOINC.

Each LOINC Term that falls within the scope of the Cooperative Work will be associated with a SNOMED CT Expression using the SNOMED CT Concept Model. Each Expression Association links the meaning of a LOINC Term to SNOMED CT, without requiring the creation of a new SNOMED CT Concept. Expression associations will be defined using SNOMED CT Compositional Grammar. Each *Expression Association* will include an indication of whether it represents a 'sufficient and necessary' (i.e. equivalent to) definition of the LOINC Term, or a 'necessary but not sufficient' (i.e. subtype of) definition of the LOINC Term.

3 Requirements

In this chapter, we state the requirements of SNOMED CT Compositional Grammar. These requirements are grouped into *General SNOMED CT Language Requirements,* which are shared by all SNOMED computable languages, and *Representation of Clinical Meaning Requirements*, which enable the representation of clinical meanings in health records, messages or expression associations.

3.1 General SNOMED CT Language Requirements

The general SNOMED CT language requirements include:

Requirement G.1: Backward compatibility

> The language must be backwardly compatible with any version of the language that has previously been adopted as an SNOMED International standard. This means that any expression that is valid based on the previous SNOMED CT Compositional Grammar standard

[1] should also be valid based on this updated version. Please note that this requirement does not preclude extensions to the language which do not affect its backward compatibility.

Requirement G.2: Consistency

> Each logical feature of the language should have a single, consistent meaning across all SNOMED computable languages. Each logical feature should also have a consistent set of syntax representations.

Requirement G.3: Sufficient and necessary

> Each language must be sufficiently expressive to meet the requirements defined for the use cases it was designed for. However, functionality without a corresponding use case will not be included, as this increases the complexity of implementation unnecessarily.

Requirement G.4: Machine processability

> In order to facilitate the easy adoption by technical audiences, instances of each language must be able to be parsed into a logical representation using a machine processable syntax specification. This requirement will be met by defining the language syntax in ABNF.

Requirement G.5: Human readability

> Non-technical stakeholders require that the language is as human readable as possible, while still meeting the other requirements. This is essential for both the clinical validation of expressions, as well as for the education and training required to author expressions.

---

[1] Based on correct parsing using the ABNF specification

3.2 Representation Of Clinical Meanings

The requirements for representing postcoordinated clinical meanings in health records, messages and expression associations are:

Requirement M1: Unambiguous meaning

> An expression must be able to specify one unambiguous meaning, independently of the implementation environment.

Requirement M2: Conformance to concept model

> Expressions must conform to the rules defined by the SNOMED CT concept model, or be transformable into an expression that conforms to the SNOMED CT concept model. For example, expressions with a clinical finding that is refined using the attribute 272741003 |Laterality|, can be transformed into a conformant expression by applying the laterality to every lateralizable body structure within the definition of the clinical finding. Please

note that expressions may either conform to the international SNOMED CT concept model, or a local (e.g. national) extension to the concept model. For more information please refer to For more information please refer to section 12.3.17 and section 12.4.15 of the Terminology Services Guide.

Requirement M3: Subset of SNOMED CT's Description Logic profile

The logical features provided by SNOMED CT compositional grammar must be a subset of those supported by SNOMED CT's Description Logic profile. This profile is primarily based on OWL 2 EL, however some additional features (which are still able to be classified in polynomial time) may also be included. This ensures that all clinical meanings expressed using SNOMED CT compositional grammar can be translated into an equivalent representation in OWL and classified in polynomial time.

Requirement M4: Functional requirements

The logical model must support the following capabilities:

Function	Details
Concept reference	The ability to reference a precoordinated SNOMED CT concept using its identifier and optional human-readable term.
Intersection / Conjunction	The ability to connect two focus concepts via a logical AND operator.
Refinement	The ability to refine (or specialise) the meaning of an expression using one or more attributes values.
Attribute group	The ability to group a collection of attributes which operate together as part of a refinement.
Attribute	An attribute-value pair which further refines the meaning of the expression
Nesting	The ability to include an expression as the value of an attribute
Concrete values	The ability to use integers, decimals, strings and booleans as attribute values
Definition status	The ability to specify whether the clinical meaning is 'equivalent to' or a 'subtype of' the expression.

4 Logical Model

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[1] 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). Figure 2 below illustrates the overall structure of a compositional grammar expression using an abstract representation. Please note that no specific semantics should be attributed to each arrow in this abstract diagram.

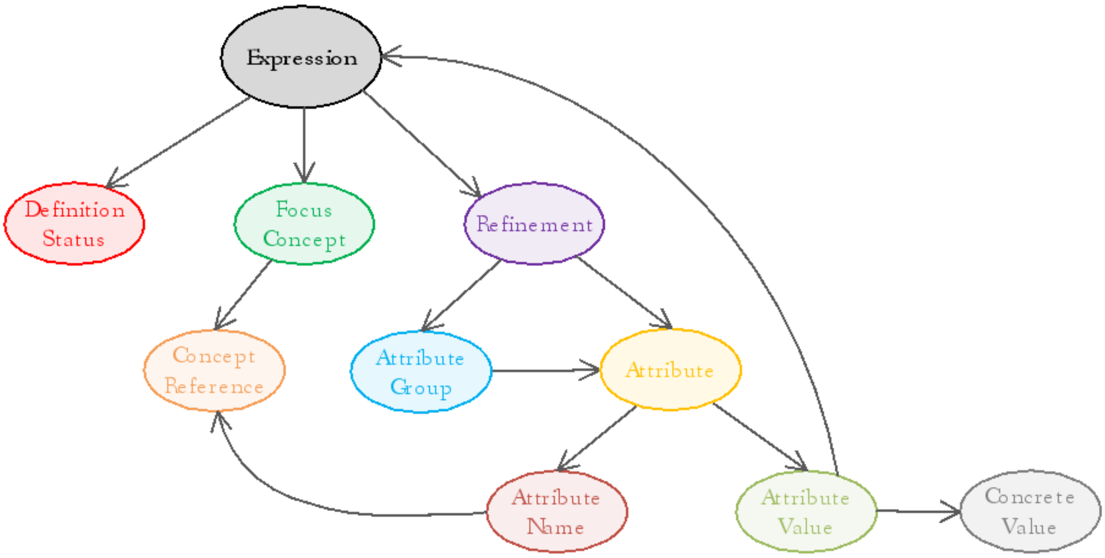


Figure 2: Abstract model of a compositional grammar expression

Figure 3 below shows an example of an expression with the main components marked. These components will be explained further in the subsequent sections of this document.

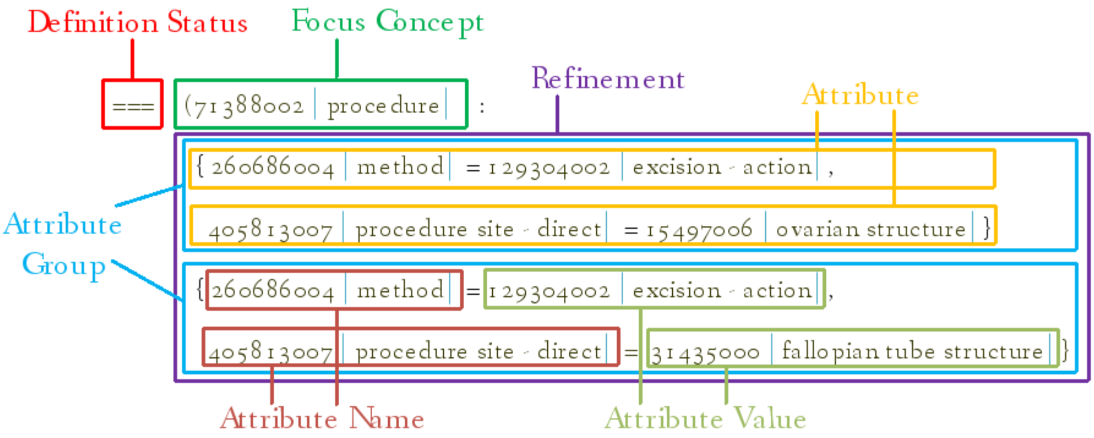

Definition Status
Focus Concept
Refinement
Attribute
=== (71388002 |procedure| :
{ 260686004 |method| = 129304002 |excision - action| ,
405813007 |procedure site - direct| = 15497006 |ovarian structure| }
{ 260686004 |method| = 129304002 |excision - action| ,
405813007 |procedure site - direct| = 31435000 |fallopian tube structure| }
Attribute Group
Attribute Name
Attribute Value

Figure 3: The main parts of an example expression

[1] 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).

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[2] 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 (string, integer or decimal). Figure 2 below illustrates the overall structure of a compositional grammar expression using an abstract representation. Please note that no specific semantics should be attributed to each arrow in this abstract diagram.

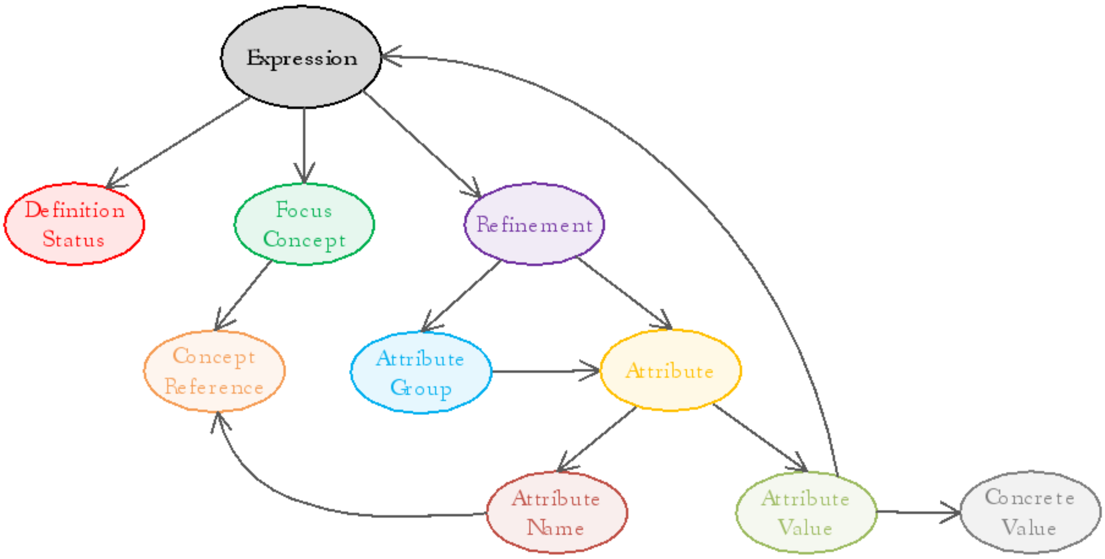


Figure 2: Abstract model of a compositional grammar expression

Figure 3 below shows an example of an expression with the main components marked. These components will be explained further in the subsequent sections of this document.

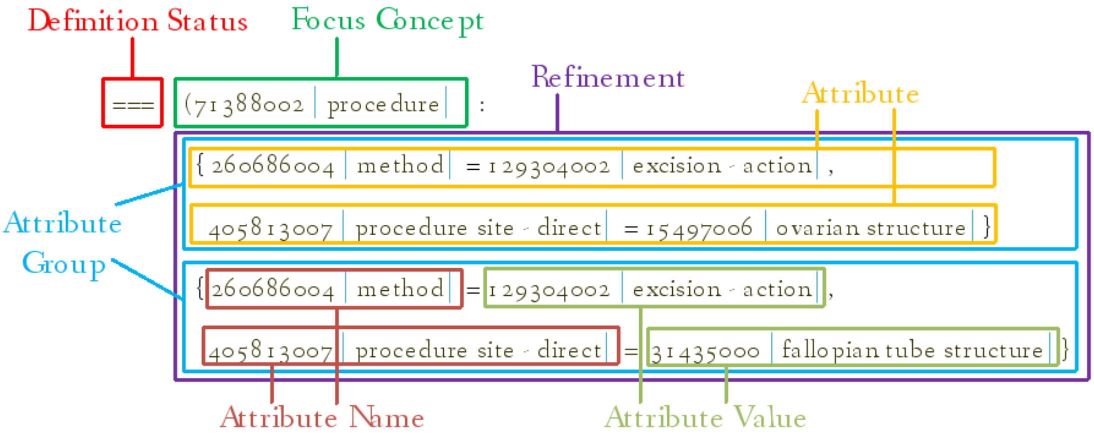

Definition Status
Focus Concept
Refinement
Attribute
Attribute Group
===
(71388002 | procedure | :
{ 260686004 | method | = 129304002 | excision - action | ,
405813007 | procedure site - direct | = 15497006 | ovarian structure | }
{ 260686004 | method | = 129304002 | excision - action | ,
405813007 | procedure site - direct | = 31435000 | fallopian tube structure | }
Attribute Name
Attribute Value

Figure 3: The main parts of an example expression

[2] 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).

4.1 Details

Figure 4 below provides a more formal representation of the logical model of SNOMED CT Compositional Grammar using a UML class diagram.

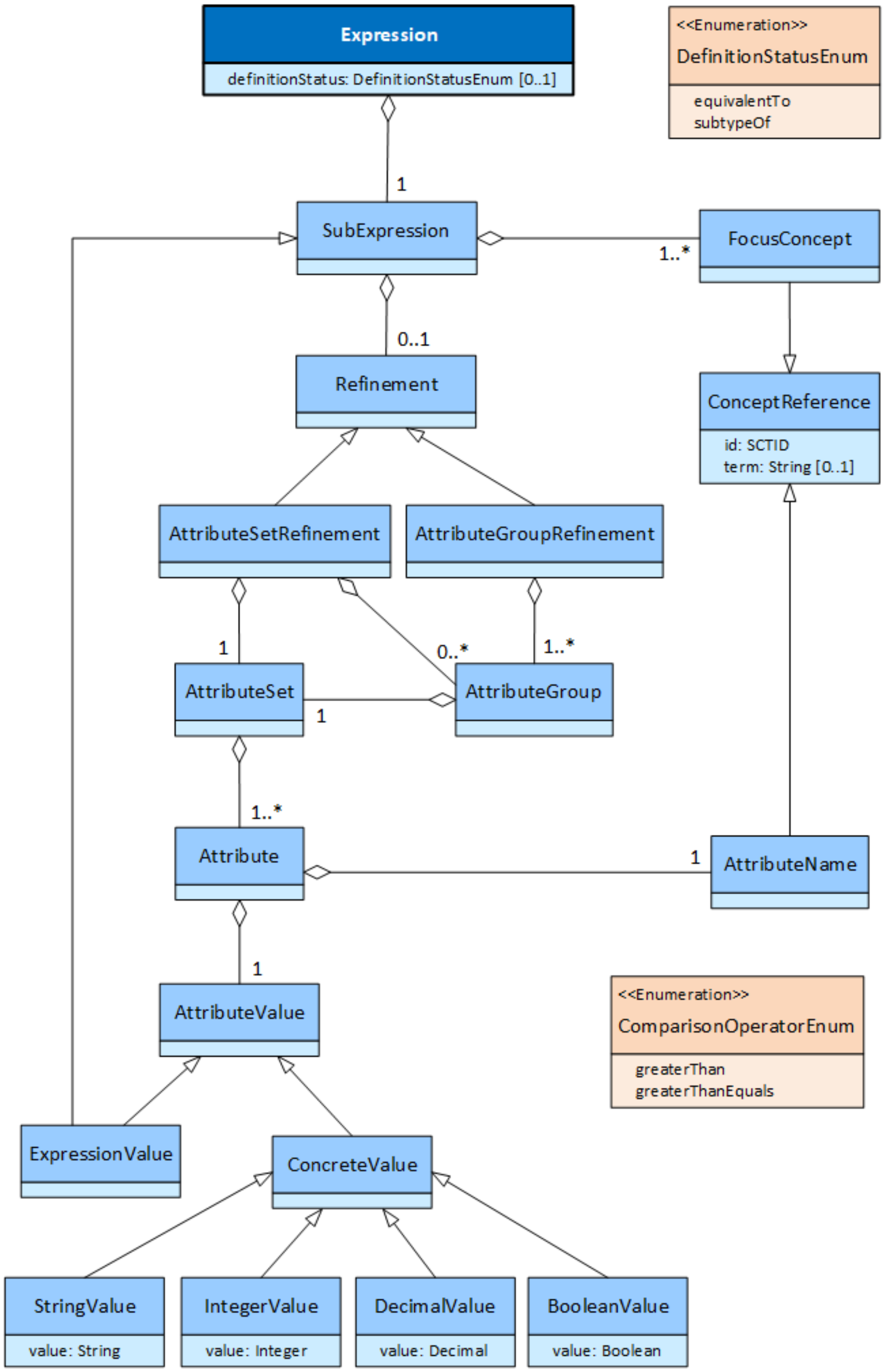


Figure 4: Logical Model of SNOMED CT Compositional Grammar

5 Syntax Specification

The following sections describe the syntax used by the SNOMED CT Compositional Grammar expression language. This syntax is a serialised representation of the logical model presented in the chapter 4, and is therefore logically equivalent. While alternative syntaxes are possible, the syntax presented here is considered to be the normative standard for interoperability purposes.

5.1 Normative Specification

The following ABNF definition specifies the syntax of SNOMED CT Compositional Grammar.

```
expression =  ws [definitionStatus ws] subExpression ws
subExpression = focusConcept [ws ":" ws refinement]
definitionStatus = equivalentTo / subtypeOf
equivalentTo = "==="
subtypeOf = "<<<"
focusConcept = conceptReference *(ws "+" ws conceptReference)
conceptReference = conceptId [ws "|" ws  term ws "|"]
conceptId = sctId
term = nonwsNonPipe *( *SP nonwsNonPipe )
refinement = (attributeSet / attributeGroup) *( ws ["," ws] attributeGroup )
attributeGroup = "{" ws attributeSet ws "}"
attributeSet = attribute *(ws "," ws attribute)
attribute = attributeName ws "=" ws attributeValue
attributeName = conceptReference
attributeValue = expressionValue / QM stringValue QM / "#" numericValue / booleanValue
expressionValue = conceptReference / "(" ws subExpression ws ")"
stringValue = 1*(anyNonEscapedChar / escapedChar)
numericValue = ["-"/"+"] (decimalValue / integerValue)
integerValue =  digitNonZero *digit / zero
decimalValue = integerValue "." 1*digit
booleanValue = true / false
true = ("t"/"T") ("r"/"R") ("u"/"U") ("e"/"E")
false = ("f"/"F") ("a"/"A") ("l"/"L") ("s"/"S") ("e"/"E")
sctId = digitNonZero 5*17( digit )
ws = *( SP / HTAB / CR / LF ) ; optional white space
SP = %x20 ; space
HTAB = %x09 ; tab
CR = %x0D ; carriage return
LF = %x0A ; line feed
QM = %x22 ; quotation mark
BS = %x5C ; back slash
digit = %x30-39
zero = %x30
digitNonZero = %x31-39
nonwsNonPipe = %x21-7B / %x7D-7E / UTF8-2 / UTF8-3 / UTF8-4
anyNonEscapedChar = HTAB / CR / LF / %x20-21 / %x23-5B / %x5D-7E / UTF8-2 / UTF8-3 / UTF8-4
escapedChar = BS QM / BS BS
UTF8-2 = %xC2-DF UTF8-tail
UTF8-3 = %xE0 %xA0-BF UTF8-tail / %xE1-EC 2( UTF8-tail ) / %xED %x80-9F UTF8-tail / %xEE-EF 2( UTF8-tail )
UTF8-4 = %xF0 %x90-BF 2( UTF8-tail ) / %xF1-F3 3( UTF8-tail ) / %xF4 %x80-8F 2( UTF8-tail )
UTF8-tail = %x80-BF
```

5.2 Informative Comments

This section provides a brief description of each rule listed in the normative specification.

<table>
<tr><td colspan="2"><b>expression</b> = ws [definitionStatus ws] subExpression ws</td></tr>
<tr><td></td><td>An expression often consists only of a subExpression. However, in some cases a definition status is needed to state whether the clinical meaning being represented is equivalent to or a subtype of the subExpression. If no definition status is included, the clinical meaning being represented is assumed to be ‘equivalent to’ the subExpression.</td></tr>
<tr><td colspan="2"><b>subExpression</b> = focusConcept [ws ":" ws refinement]</td></tr>
<tr><td></td><td>A subexpression consists of one or more focus concepts, optionally followed by a refinement. The meaning of the expression is a subtype of all the focus concepts constrained by the refinement. Note that where there is a requirement for multiple separately qualified concepts to be present these are expressed in attribute groups within a refinement of a general concept such as " situation with explicit context ".</td></tr>
<tr><td colspan="2"><b>definitionStatus</b> = equivalentTo / subtypeOf</td></tr>
<tr><td></td><td>The definition status states whether the clinical meaning being expressed is equivalent to or a subtype of the given expression representation.</td></tr>
<tr><td colspan="2"><b>equivalentTo</b> = "==="</td></tr>
<tr><td></td><td>A definition status of equivalentTo (i.e. “===”) indicates that the clinical meaning being represented is semantically equivalent to the given expression. If no definition status is included in the expression, this definition status is assumed.</td></tr>
<tr><td colspan="2"><b>subtypeOf</b> = "<<<"</td></tr>
<tr><td></td><td>A definition status of subtypeOf (i.e. “<<<”) indicates that the clinical meaning being represented is a semantic subtype (or subclass) of the given expression.</td></tr>
<tr><td colspan="2"><b>focusConcept</b> = conceptReference *(ws "+" ws conceptReference)</td></tr>
<tr><td></td><td>A focusConcept consists of one or more concept references separated by ‘plus’ signs.</td></tr>
<tr><td colspan="2"><b>conceptReference</b> = conceptId [ws " | " ws term ws " | "]</td></tr>
<tr><td></td><td>A conceptReference is represented by a ConceptId optionally followed by a term enclosed by a pair of "|" characters. Whitespace before or after the ConceptId is ignored as is any whitespace between the initial "|" characters and the first non-whitespace character in the term or between the last non-whitespace character and before second "|" character.</td></tr>
<tr><td colspan="2"><b>conceptId</b> = sctId</td></tr>
<tr><td></td><td>The ConceptId must be a valid SNOMED CT identifier for a concept. The initial digit may not be zero. The smallest number of digits is six, and the maximum is 18.</td></tr>
<tr><td colspan="2"><b>term</b> = nonwsNonPipe *(*SP nonwsNonPipe)</td></tr>
</table>

	The term must be the term from any SNOMED CT description that is associated with the concept identified by the preceding concept identifier. For example, the term could be the preferred synonym for a given dialect. The term may include valid UTF-8 characters except for the pipe "|" character. The term begins with the first non-whitespace character following the starting "|" character and ends with the last non-whitespace character preceding the next "|" character.
refinement = (attributeSet / attributeGroup) *(ws ["," ws] attributeGroup)	
	A refinement contains all the grouped and ungrouped attributes that refine the meaning of the containing expression.
attributeGroup = "**{**" ws attributeSet ws "**}**"	
	An attribute group contains a collection of attributes that operate together as part of the refinement of the containing expression.
attributeSet = attribute *(ws " **,** " ws attribute)	
	An attribute set contains one or more attribute name-value pairs, separated by commas.
attribute = attributeName ws "=" ws attributeValue	
	An attribute is a name-value pair expressing a single refinement of the containing expression.
attributeName = conceptReference	
	The attribute name is the name of an attribute (or relationship type) to which a value is applied to refine the meaning of a containing expression. The attribute name is represented in the same way as other concept references.
attributeValue = expressionValue / QM stringValue QM / “#” numericValue / booleanValue	
	An attribute value is either an expression, a string-based concrete value enclosed in quotation marks, a numeric concrete value (i.e. integer or decimal) preceded by a “#”, or a boolean.
expressionValue = conceptReference / "**(**" ws subExpression ws "**)**"	
	An expression value is either a single concept reference without brackets, or a more complex expression enclosed in round brackets.
stringValue = 1*(anyNonEscapedChar / escapedChar)	
	A string value includes one or more printable ASCII characters (these are also valid UTF8 characters encoded as one octet) and/or UTF8 characters encoded as 2- 3- or 4-octet sequences. Quotes and backslash characters must be preceded by the escape character (“\”).
numericValue = ["-"/"+"] (decimalValue / integerValue)	
	A numeric value is either an integer or a decimal. Positive numbers optionally start with a plus sign ("+"), while negative integers begin with a minus sign ("-").
integerValue = digitNonZero *digit / zero	

	An integer value is either starts with a non-zero digit followed by zero to many additional digits, or is the integer zero itself.
decimalValue = integerValue "." 1*digit	
	A decimal value starts with an integer. This is followed by a decimal point and one to many digits.
booleanValue = true / false	
	A boolean value is either true or false.
true = ("t"/"T") ("r"/"R") ("u"/"U") ("e"/"E")	
	The boolean value of true is represented as "TRUE" using any combination of upper or lower characters.
false = ("f"/"F") ("a"/"A") ("l"/"L") ("s"/"S") ("e"/"E")	
	The boolean value of false is represented as "FALSE" using any combination of upper and lower characters.
sctId = digitNonZero 5*17(digit)	
	A SNOMED CT id is used to represent an attribute id or a concept id. The initial digit may not be zero. The smallest number of digits is six, and the maximum is 18.
ws = *(SP | HTAB | CR | LF)	
	Optional whitespace characters (space, tab, carriage return and linefeed) are ignored everywhere in the expression except: 1. Whitespace within a conceptId is an error. (**Note:** Whitespace before or after the last digit of a valid Identifier is ignored.) 2. Non-consecutive spaces within a term are treated as a significant character of the term. (**Note:** Whitespace before the first or after the last non-whitespace character of a term is ignored.) 3. Whitespace within the quotation marks of a concrete value is treated as a significant character.
SP = %x20	
	Space character.
HTAB = %x09	
	Tab character.
CR = %x0D	
	Carriage return character.
LF = %x0A	
	Line feed character.
QM = %x22 ; quotation mark	
	Quotation mark character.

BS = %x5C	
	Back slash character.
digit = %x30-39	
	Any digit 0 through 9.
zero = %x30	
	The digit 0.
digitNonZero = %x31-39	
	Digits 1 through 9, but excluding 0. The first character of a concept identifier is constrained to a digit other than zero.
nonwsNonPipe= %x21-7B / %x7D-7E / UTF8-2 / UTF8-3 / UTF8-4	
	Non whitespace (and non pipe) includes printable ASCII characters (these are also valid UTF8 characters encoded as one octet) and also includes all UTF8 characters encoded as 2- 3- or 4-octet sequences. It excludes space (which is %x20) and the pipe character "|" (which is %x7C), and excludes CR, LF, HTAB and other ASCII control codes. See RFC 3629 (UTF-8, a transformation format of ISO 10646 authored by the Network Working Group).
anyNonEscapedChar = HTAB / CR / LF / %x20-21 / %x23-5B / %x5D-7E / UTF8-2 / UTF8-3 / UTF8-4	
	anyNonEscapedChar includes any printable ASCII characters which do not need to be preceded by an escape character (i.e. “\”). This includes valid UTF8 characters encoded as one octet and all UTF8 characters encoded as 2, 3 or 4 octet sequences. It does, however, exclude the quotation mark (“) and the backslash (\). See RFC 3629 (UTF-8, a transformation format of ISO 10646 authored by the Network Working Group).
escapedChar = BS QM / BS BS	
	The double quotation mark and the back slash character must both be escaped within a string-based concrete value by preceding them with a back slash.
UTF8-2 = %xC2-DF UTF8-tail	
	UTF8 characters encoded as 2-octet sequences.
UTF8-3 = %xE0 %xA0-BF UTF8-tail / %xE1-EC 2(UTF8-tail) / %xED %x80-9F UTF8-tail / %xEE-EF 2(UTF8-tail)	
	UTF8 characters encoded as 3-octet sequences.
UTF8-4 = %xF0 %x90-BF 2(UTF8-tail) / %xF1-F3 3(UTF8-tail) / %xF4 %x80-8F 2(UTF8-tail)	
	UTF8 characters encoded as 4-octet sequences.
UTF8-tail = %x80-BF	
	UTF8 characters encoded as 8-octet sequences.

6 Examples

The examples in this section illustrate the syntax presented in section 5, and help to explain how it can be used to represent clinical meanings.

Please note that an expression containing a single concept reference is referred to as a *precoordinated expression*. Precoordinated expressions are discussed in section 6.1. Expressions which contain two or more concept references are called *postcoordinated expressions*. Postcoordinated expressions are discussed in section 6.2 to section 6.7.

6.1 Simple Expression

Precoordinated expressions consist of a single concept identifier, followed optionally by any term that is associated with that concept. While the use of a term is not mandated, it is often preferable to include preferred term from an appropriate dialect within the expressions to improve the human-readability of the expression. For example: 73211009 |Diabetes mellitus|As mentioned, the syntax does not require a term to be used with each concept identifier, so the following is also a valid expression: 73211009 |diabetes mellitus|

6.2 Multiple Focus Concepts

Two or more concepts may be combined to form a new concept by joining them with the "+" symbol. This is particularly useful where both concepts are primitive. The resultant expression is the child of each of the concepts in the expression. The resultant expression below represents a drug dose form that is both a spray and a suspension.

```
421720008 |Spray dose form| + 7946007 |Drug suspension|
```

Although not stipulated by the syntax, the two concepts joined in this way must (in nearly every case) come from the same top level hierarchy. A rare exception to this rule is those expressions which refer to products that are both drugs and devices, and are therefore descendants of both 373873005 |pharmaceutical / biologic product| and 260787004 |physical object|.

The syntax does not mandate which concepts in the expression should have associated terms and which should not, so it is valid (although not advisable) to mix and match. For example, the following syntax is valid: 421720008 + 7946007 |Drug suspension|The syntax allows spaces, tabs and carriage returns in most places. The following example has an identical meaning to the one above: 421720008
    + 7946007 |Drug suspension|

6.3 Expression With Refinements

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

```
83152002 |Oophorectomy| : 405815000 |Procedure device| = 122456005 |Laser device|
```

The next example, shown below, describes the right hip joint.

```
182201002 |Hip joint| : 272741003 |Laterality| = 24028007 |Right|
```

Where more than one qualifying attribute is required, these can be separated using a comma. The example below describes the removal of an ovarian structure using a laser device.

```
71388002 |Procedure| :
  405815000 |Procedure device| = 122456005 |Laser device| ,
  260686004 |Method| = 129304002 |Excision - action| ,
  405813007 |Procedure site - direct| = 15497006 |Ovarian structure|
```

A further example, below, describes the emergency removal of the appendix:

```
65801008 |Excision| :
  405813007 |Procedure site - direct| = 66754008 |Appendix structure| ,
  260870009 |Priority| = 25876001 |Emergency|
```

Refinements may also be applied to a conjoined concept. For example, the following two expressions are equivalent (because an 'ulna' is a bone in an 'upper limb'):

```
313056006 |Epiphysis of ulna| : 272741003 |Laterality| = 7771000 |Left|
```

```
119189000 |Ulna part| + 312845000 |Epiphysis of upper limb| : 272741003 |Laterality| = 7771000 |Left|
```

Note that there are no brackets round "

```
119189000 |Ulna part| + 312845000 |Epiphysis of upper limb|
```

" in the above example, as the refinement automatically applies to the conjoined concept.

6.4 Expressions With Attribute Groups

Where a SNOMED CT concept definition comprises a number of qualifying attributes, it may be necessary to group these to avoid ambiguity as to how they apply. An example of a SNOMED CT concept whose definition comprises multiple attribute groups is: 116028008 |Salpingo-oophorectomy|This procedure comprises two sub-procedures: the excision of part or all of the ovarian structure and the excision of part or all of the fallopian tube structure. This is demonstrated by the SNOMED CT expression for salpingo-oophorectomy, as shown below:

```
71388002 |Procedure| :
{ 260686004 |Method| = 129304002 |Excision - action| ,
  405813007 |Procedure site - direct| = 15497006 |Ovarian structure| }
{ 260686004 |Method| = 129304002 |Excision - action| ,
  405813007 |Procedure site - direct| = 31435000 |Fallopian tube structure| }
```

Where it is necessary to unambiguously qualify individual attributes of a concept whose definition comprises a number of attribute groups, a postcoordinated expression with grouping may be used. The following example describes a salpingo-oophorectomy, with laser excision of the right ovary and diathermy excision of the left fallopian tube. Note that without the grouping, it would not be possible to tell on which structure the laser excision was used and on which structure the diathermy excision was used.

```
71388002 |procedure| :
{ 260686004 |method| = 129304002 |excision - action| ,
  405813007 |procedure site - direct| = 20837000 |structure of right ovary| ,
  424226004 |using device| = 122456005 |laser device| }
{ 260686004 |method| = 261519002 |diathermy excision - action| ,
  405813007 |procedure site - direct| = 113293009 |structure of left fallopian tube| }
```

A number of grouped qualifiers may thus be used to refine a concept. Note that the comma between adjacent groups is optional (and omitted in this example), while the comma between adjacent attributes is mandatory. Also note, the syntax does not limit the maximum number of attributes in a group or the maximum number of groups within an expression.

It is also worth noting that when all attributes in a refinement belong to the same attribute group the braces around the attribute group are optional. Therefore the following two expressions are considered to be equivalent:

```
71388002 |Procedure| :
  260686004 |Method| = 129304002 |Excision - action| ,
  405813007 |Procedure site - direct| = 15497006 |Ovarian structure|
```

```
71388002 |procedure| :
{ 260686004 |method| = 129304002 |excision - action| ,
  405813007 |procedure site - direct| = 15497006 |ovarian structure| }
```

6.5 Expression With Nested Refinements

It is also possible to nest expressions, one inside the other. Any valid expression may be enclosed in a pair of brackets, and included as the value of an attribute in another expression. For example, the following expression describes a medication product that has a single dose form, which is both a spray and a suspension:

```
373873005 |Pharmaceutical / biologic product| :
  411116001 |Has dose form| = ( 421720008 |Spray dose form| + 7946007 |Drug suspension| )
```

In the example above, note the use of round brackets (i.e. "( )") to identify a nested expression, as opposed to braces (i.e. "{ }"), which is used to identify attribute groups.

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

```
24136001 |Hip joint structure| :
  272741003 |Laterality| = 7771000 |Left|
```

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

```
397956004 |Prosthetic arthroplasty of the hip| :
  363704007 |Procedure site| = ( 24136001 |Hip joint structure| :
    272741003 |Laterality| = 7771000 |Left| )
```

Applying a further grouped refinement to the above describes a procedure to replace a left hip by inserting a prosthesis. Note that this example mixes an ungrouped qualification and a grouped qualification. Where this is done, the canonical representation of the expression includes all ungrouped attributes before the grouped ones (see Terminology Services Guide for more details). Please also note that in the previous version of this standard a comma was not permitted between the last ungrouped attribute and the first attribute group – however this comma is now optional to avoid unexpected syntax errors.

```
397956004 |Prosthetic arthroplasty of the hip| :
  405814001 |Procedure site - indirect| = ( 24136001 |Hip joint structure| : 272741003 |Laterality| = 7771000
|Left| ),
 { 363699004 |Direct device| = 304120007 |Total hip replacement prosthesis| ,
   260686004 |Method| = 425362007 |Surgical insertion - action| }
```

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

```
243796009 |Situation with explicit context| :
{ 408730004 |Procedure context| = 385658003 |Done| ,
  408731000 |Temporal context| = 410512000 |Current or specified| ,
  408732007 |Subject relationship context| = 410604004 |Subject of record| ,
  363589002 |Associated procedure| =
 ( 397956004 |Prosthetic arthroplasty of the hip| :
     405814001 |Procedure site - indirect| = ( 24136001 |Hip joint structure| :
       272741003 |Laterality| = 7771000 |Left| )
   { 363699004 |Direct device| = 304120007 |Total hip replacement prosthesis| ,
     260686004 |Method| = 425362007 |Surgical insertion - action| } ) }
```

6.6 Expressions With Concrete Values

New attributes that are added to the SNOMED CT concept model (or to a SNOMED CT extension) may require their value to be concrete. Concrete values include integers (e.g. 500), decimals (e.g. 2.75), strings (e.g. "PANADOL"), and booleans (e.g. TRUE).

The expression shown below uses both concept values and concrete values to represent a capsule that contains 500 mg of amoxicillin[1] . In this case, the dose form, the active ingredient, the basis of strength substance and the strength unit all use a concept as their value, while the strength magnitude is represented using a numeric value. The numeric value is indicated in the expression by the preceding '#' symbol (e.g. #500).

```
774586009 |Amoxicillin only product| :
  411116001 |Has manufactured dose form| = 420692007 |Oral capsule| ,
 { 127489000 |Has active ingredient| = 372687004 |Amoxicillin| ,
 179999999100 |Has basis of strength| = 372687004 |Amoxicillin| ,
 189999999103 |Has strength value| = #500, 199999999101 |Has strength unit| = 258684004 |mg| }
```

The expression below represents an oral solution that contains 0.083 % albuterol.

```
   91143003 |Albuterol| :
    411116001 |Has manufactured dose form| = 385023001 |oral solution| ,
   { 127489000 |Has active ingredient| = 372897005 |Albuterol| ,
     179999999100 |Has basis of strength| = 372897005 |Albuterol| ,
     189999999103 |Has strength value| = #0.083, 199999999101 |Has strength numerator unit| = 118582008 |
%| }
```

Boolean and string-based concrete values may also be used. For example, the expression below represents a tablet containing 500 mg of paracetamol with a trade name (or brand name) of "PANADOL"[2].

```
   322236009 |Paracetamol 500mg tablet| : 209999999104 |Has trade name| = "PANADOL"
```

And the expression below represents an Irbesartan 150 mg tablet that is eligible for reimbursement through the national benefit scheme (of the given country).

```
  318969005 |Irbesartan 150 mg oral tablet| : 859999999102 |Is in national benefit scheme| = TRUE
```

Please note that boolean and string-based concrete values should be used with caution. Care must be taken to only use these in situations in which the clinical meaning cannot be adequately defined using a concept value.

[1] Please note that these example expressions are based on a hypothetical drug concept model, and are not intended to reflect any specific drug model. In these expressions, the SNOMED CT identifiers created with the '99 99999' namespace are for example only, and should not be used in a production environment.

[2] Concrete values of type string are case sensitive and compared using the Unicode Collation Algorithm (http://www.unicode.org/reports/tr10/).

6.7 Expressions With a 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.

Equivalent To

When the combination of focus concepts and attribute refinements are both necessary and sufficient to define the clinical meaning being represented, the 'equivalent to' definition status is used. Expressions with a definition status of 'equivalent to' may be used to express a fully defined clinical meaning in a health record or a message, to represent the definition of a fully defined precoordinated concept, or to represent an association between LOINC and an equivalent SNOMED CT expression.

To indicate that an expression has a definition status of 'equivalent to', three consecutive equals signs (i.e. "===") are placed at the start of the expression. For example, the following expression represents the definition of the fully defined concept 31978002 |Fracture of tibia|:

```
=== 46866001 |Fracture of lower limb| + 428881005 |Injury of tibia| :
    116676008 |Associated morphology| = 72704001 |Fracture| ,
    363698007 |Finding site| = 12611008 |Bone structure of tibia|
```

Because expressions are most often used to represent the full definition of a clinical meaning, 'equivalent to' is the assumed definition status when it is not explicitly stated. For this reason, the expressions above has an identical meaning to the following expression:

```
46866001 |Fracture of lower limb| + 428881005 |Injury of tibia| :
116676008 |Associated morphology| = 72704001 |Fracture| ,
363698007 |Finding site| = 12611008 |Bone structure of tibia|
```

Using 'equivalent to' as the assumed definition status helps to ensure the backward compatibility of this version of compositional grammar with the previous one.

Subtype Of

When the combination of focus concepts and attribute refinements are necessary but not necessarily sufficient to define the clinical meaning being represented, the 'subtype of' definition status is used. Expressions with a definition status of 'subtype of' may be used to represent a primitive clinical meaning in a health record or a message, the definition of a primitive precoordinated concept, or an association between LOINC and a SNOMED CT expression that is necessary but not sufficient.

To indicate that an expression has a definition status of 'subtype of', three consecutive less than signs (i.e. "<<<") are placed at the start of the expression.

For example, the primitive precoordinated concept 46635009 |Type 1 diabetes mellitus|can be defined using the expression:

```
<<< 73211009 |Diabetes mellitus| : 363698007 |Finding site| = 113331007 |Endocrine system|
```

The definition status of 'subtype of' on this expression indicates that the definition is not necessarily sufficient to define the concept 46635009 |Type 1 diabetes mellitus|.

7 Implementation Guidance

When implementing SNOMED CT Compositional Grammar expressions, the factors that need to be taken into consideration depend on what tasks are being performed. For example, implementations may require expressions to be authored, parsed, validated, stored, displayed, exchanged, classified or queried.

The subsections below look at each of these tasks individually and provide a summary of the factors that should be considered prior to implementation. Please note that the guidance provided below is not a step-by-step how-to manual, but instead provides some general insights that we hope are helpful in implementing this language specification.

7.1 Authoring

Authoring SNOMED CT postcoordinated expressions can be performed using two main techniques:

1. *Language-based authoring*: This technique involves the author constructing a SNOMED CT expression using the Compositional Grammar syntax defined in Chapter 5.
2. *Form-based authoring*: This technique involves the author entering values into separate fields of a form, and the clinical system automatically composing the values together into a syntactically correct SNOMED CT expression.

Language-Based Authoring

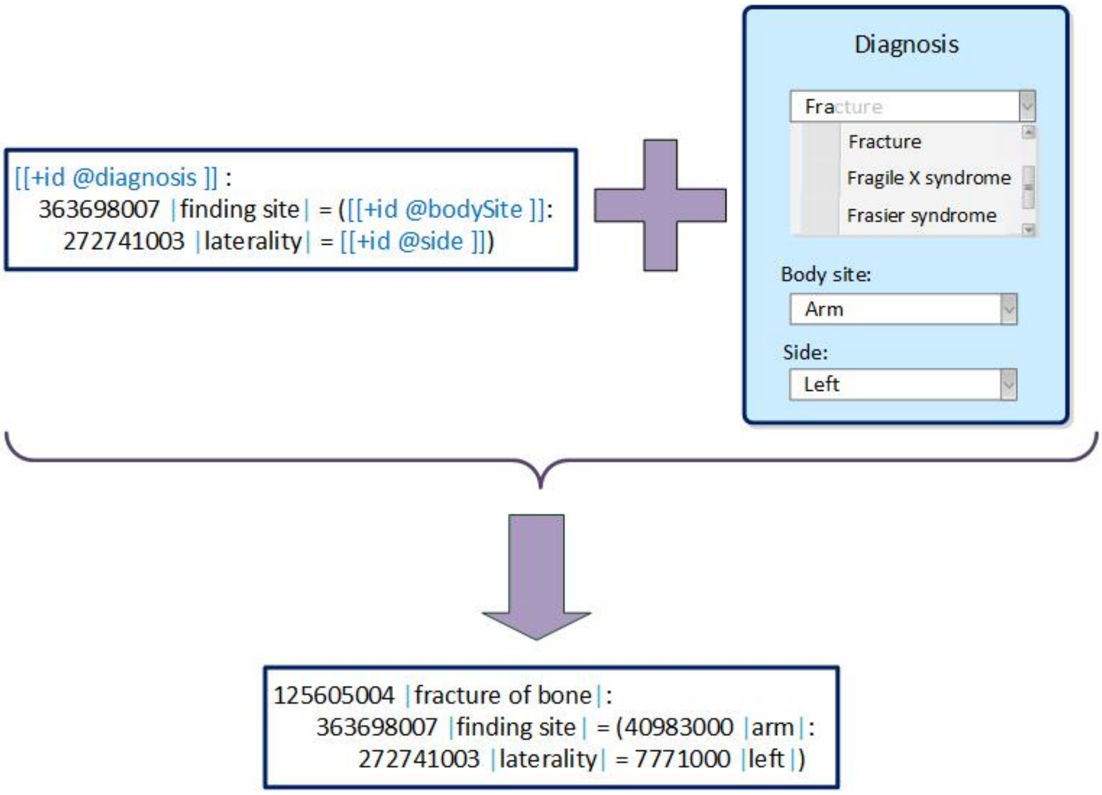
Language-based authoring is useful for situations in which expressions must be defined which don't necessarily conform to a consistent structure. To use language-based authoring, the user must be familiar with the basic features of the Compositional Grammar syntax. There are, however, a number of ways in which a tool can support the user while creating expressions, including:

- Validating the syntactical correctness of the expression as it is authored;
- Checking the expression for conformance against the concept model;
- Automatically populating or correcting the term associated with a concept reference;
- Providing integrated tools to search the SNOMED CT hierarchy for concept references to include in the expression;
- Filtering the concept search to those concepts which are valid to use at the given point in the expression (e.g. only showing attribute concepts, or those within the valid range of the given attribute); and
- Suggesting the set of valid operators or characters that may be used at a given point in the expression;

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.

Form-Based Authoring

Form-based authoring is particularly useful when non-technical users need to create postcoordinated expressions which have a consistent structure. In these situations, it may be useful to create an 'expression template' in which the attribute values are represented by slots. The slots in these expression templates can then be replaced by the values that the user enters into the associated fields of the form. It may also be useful to restrict the valid values that may be entered into each field (and therefore each slot) using an expression constraint. The diagram below illustrates this approach.



```
[[+id @diagnosis ]] :
   363698007 |finding site| = ([[+id @bodySite ]]:
     272741003 |laterality| = [[+id @side ]])
```

```
125605004 |fracture of bone|:
   363698007 |finding site| = (40983000 |arm|:
     272741003 |laterality| = 7771000 |left|)
```

Figure 5: Form-based expression authoring

For more information on adding refinements to expressions during data entry, please refer to section 2.7 of the Records Services Guide.

7.2 Parsing

Parsing is the process of analysing a string of characters according to the rules of a formal grammar. Parsing a Compositional Grammar expression involves processing the expression string, using the ABNF syntax defined in chapter 5, and breaking it into its constituent parts. This creates a representation of the expression that can be further processed. Parsing an expression is required to perform syntactic validation, concept model validation, translation, querying or classification.

A number of parser development tools are available which can generate a parser from a context-free grammar written in ABNF, such as the one defined in this document. These tools include:

- APG
- aParse
- abnfgen

Please note, the ABNF syntax defined in this specification was tested using the APG Parser Generator

[1].

Other non-ABNF parser generators are also available which can be used with an alternate syntax representation – for example:

- ANTLR
- XText
- ACE

Some of these tools (e.g. XText and ACE) can also be used to generate authoring environments with features such as syntax highlighting and autocompletion.

Alternatively, an expression parser can be created manually using a programming language such as Perl or C++.

[1] www.coasttocoastresearch.com/interactiveapg

7.3 Validating

Compositional Grammar expressions can be automatically validated to ensure that they conform to a variety of rules, including:

- Expressions must conform to the syntax defined in chapter 5. This validation can be performed using an expression parser, as described in section 7.2;
- Expressions must conform to the concept model. This validation can be performed by comparing the parsed expression against the rules defined in the SNOMED CT concept model;
- All concept references included in the expression must be valid. In most cases this means that the concept references must refer to active concepts in the given version and edition of SNOMED CT.

7.4 Storing

Storing SNOMED CT postcoordinated expressions in Electronic Health Records enables clinical systems to capture a wider range of clinical meanings than would be possible using precoordinated expressions alone. When storing postcoordinated expressions written using SNOMED CT Compositional Grammar in health records, a number of options are available including:

- Storing the entire expression as a string in the relevant field of the patient's record (as described in section 3.1 of the Record Services Guide). This requires the maximum length of the field to be large enough to support the required expressions;
- Storing a list of expressions in an *Expressions* table, and associating each expression with a local identifier. (Please refer to section 12.6.1 of the Terminology Services Guide and section 3.3 of the Record Services Guide). These local expression identifiers are then stored within the relevant field of the patient's record. The *Expressions* table enables a lookup service to match the local expression identifier stored in the patient's record with the associated postcoordinated expression. This allows shorter identifier strings to be recorded within the patient record, instead of the longer postcoordinated expressions. While this approach may have a slightly detrimental effect on the retrieval of a patient record, it can reduce the storage requirements and enable indexing over expressions for faster searching. These local identifiers, however, cannot be shared with other applications, which do not have access to the associated postcoordinated expressions;
- Using a shared expression repository to generate expression identifiers that can be stored in the relevant field of the patient's record. This approach allows shorter identifiers to be recorded in the patient records (as per the previous approach), while using identifiers which can be shared between applications that share access to the same expression repository.

Please note that while transforming an expression to other equivalent forms (e.g. a normal form) may be useful to support effective data retrieval, even minor corrections to the definition of a concept in future releases may significantly alter the resulting form of the same expression. Therefore it is recommended that:

- The primary or original record should be stored using the representation that is as close as possible to the form in which it was recorded;
- If transformations to alternative representations are used to enhance the efficiency of retrieval, these should be stored as secondary supporting tables or indices. This has the advantage that these alternative forms can be regenerated based on the most up to date set of definitions when a new release of SNOMED CT is installed, without affecting the integrity of the original records.

For alternative approaches for storing postcoordinated expressions, that do not use SNOMED CT Compositional Grammar, please refer to section 3 of the Record Services Guide.

7.5 Displaying

A number of options exist for displaying SNOMED CT expressions, including:

- Displaying the expression using compositional grammar in its originally authored and stored form;
- Displaying the human-readable term that was shown to the user when originally selecting the expression;
- Enhancing the compositional grammar by adding in terms that may have been omitted, or replacing the existing terms with either local-dialect Preferred Terms or Fully Specified Names;
- Enhancing the display by using different font colors for each different part of the expression (e.g. identifiers, terms, vertical bars, and operators), and by using whitespace in a way that improves the readability of the expression;
- Automatically transforming the expression into a human-readable term using a predefined algorithm. For example, a simple algorithm may just remove the concept identifiers and nesting – e.g. "fracture of bone: finding site = arm, laterality = left". More sophisticated algorithms may use pattern matching and predefined templates to construct a more natural term – e.g. "fracture of bone in left arm";
- Representing the attribute values of the expression by populating a structured form. This approach is primarily suited to expressions with a consistent template, where the form can be pre-designed. However populating dynamically-generated forms from an expression is possible.

Please note that while various options exist, care should be taken to ensure that the selected solution is medico-legally safe and acceptable.

7.6 Exchanging

There are a number of alternative techniques that may be used when exchanging SNOMED CT expressions in messages. These include:

- Placing the full SNOMED CT compositional grammar expression in the relevant *code* field of the message.
    - For example, as noted in section 2.3, the HL7 version 3 CD data type (Release 2) allows the 'code' property to include SNOMED CT expressions (see HL7 TermInfo guide).
- Using an alternative structure for representation of the logical structure of an expression.
    - For example an OWL serialization.
- Placing a unique expression identifier in the relevant *code* field of the message
    - 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.
- Decomposing the expression into separate codes that can be placed in different fields within the message structure.
    - This approach assumes alignment between the message structure and the relevant parts of the SNOMED CT concept model.

Please refer to section 5 of the Record Services Guide for more information on exchanging expressions.

7.7 Classifying

In order to automatically understand the inferred relationships between SNOMED CT expressions and/or precoordinated concepts, the expressions should be classified using a semantic reasoner. A number of reasoners exist, such as Snorocket, ELK and FACT++, which can classify and reason over OWL 2 EL.

Before a SNOMED CT compositional grammar expression can be classified using one of these reasoners, it must first be parsed (as described in section 7.2), normalised and then translated into OWL 2 EL. The specific normalisation transformations that are required prior to translation into OWL include:

- Grouping all ungrouped attributes with a relationship type that is allowed to be grouped. For more information please refer to section 9.2.2 of the Terminology Services Guide; and
- Transforming expressions to ensure that they conform to the concept model – in particular, where a laterality refinement has been applied to a focus concept that is not subsumed by 123037004 |Body

structure| , apply this laterality to all lateralisable finding sites within the definition of this focus concept. For more information please refer to section 12.3.17 and section 12.4.15 of the Terminology Services Guide.

The translation into OWL can then be performed in a variety of programming languages, including Perl.

Please note that an alternative way to classify a set of expressions is to test each pair of expressions for equivalence and subsumption by performing a string-based comparison on their normal forms. For more information on this approach please refer to section 12.5 of the Terminology Services Guide.

7.8 Querying

A number of approaches exist to enable querying over SNOMED CT compositional grammar expressions. The recommended approach, however, is to first parse, translate (into OWL 2 EL) and classify the expressions together with the precoordinated content of SNOMED CT. A query language (such as the SNOMED CT Expression Constraint language, SNOMED CT Query language, SPARQL or DL Query) can then be used to construct queries, which are similarly parsed, translated and executed using an appropriate query engine. For more information about the SNOMED CT expression constraint and query languages, please refer to the Expression Constraint Specification and Guide and SNOMED CT Glossary respectively.

An alternative approach to querying by testing two expressions for equivalence and subsumption is to transform each expression into its normal form and perform a comparison between the two normal form expressions. For more information on testing equivalence and subsumption between expressions using this approach please refer to section 12.5 of the Terminology Services Guide.

Appendix A – Concept Definition and Expression Relationship Statements

SNOMED CT Compositional Grammar is a syntax for representing SNOMED CT expressions. SNOMED CT expressions are a structured combination of one or more concept identifiers used to express a clinical idea. When documenting a full concept definition or an expression relationship statement, which asserts the relationship between two expressions, it is often useful to use a syntax that is consistent with SNOMED CT Compositional Grammar. For these purposes, the following ABNF syntax extension may be used:

statement = ws "(" ws subExpression ws ")" ws definitionStatus ws "(" ws subExpression ws ")" ws

where 'subExpression' and 'definitionStatus' are defined as per SNOMED CT Compositional Grammar.

For example, the following statement documents the definition of the fully defined concept 95617006 |Neonatal cyanosis|.

```
( 95617006 |Neonatal cyanosis| ) ===
 ( 3415004 |Cyanosis| + 363696006 |Neonatal cardiovascular disorder| :
    246454002 |Occurrence| = 255407002 |Neonatal| ,
    363698007 |Finding site| = 113257007 |Structure of cardiovascular system| )
```

Figure 6 illustrates this concept definition using the SNOMED CT Diagramming Guideline.

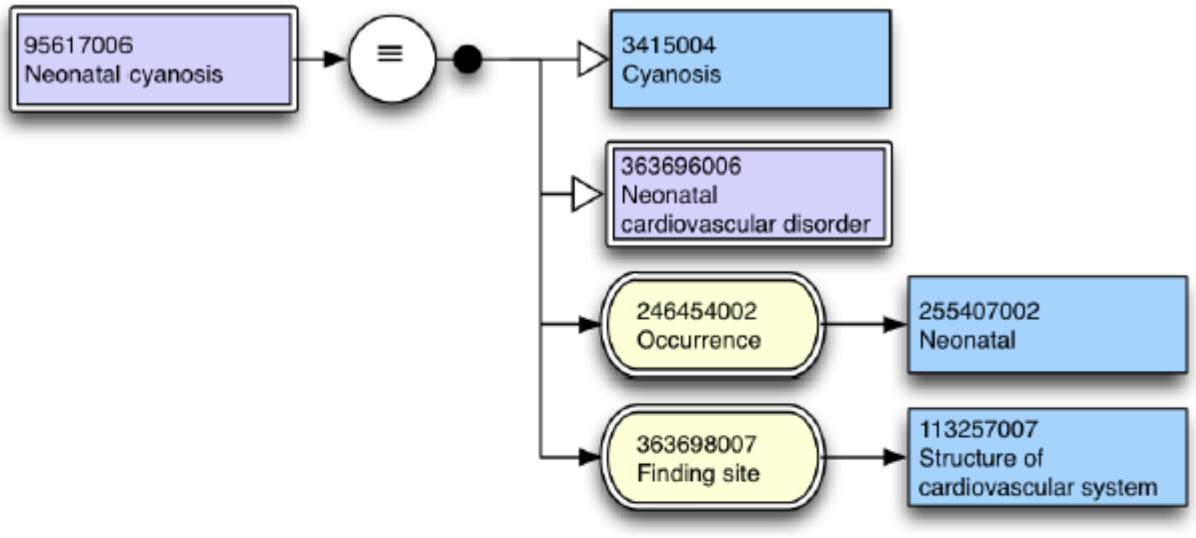


Figure 6: Concept definition diagram for Neonatal cyanosis

In this second example, the definition of the primitive concept 144008 |Normal peripheral vision| is shown:

```
( 144008 |Normal peripheral vision| ) <<< ( 301980006 |Finding of visual field| :
   363698007 |Finding site| = 49549006 |Structure of visual system| )
```

Figure 7 illustrates this concept definition using the SNOMED CT Diagramming Guideline.

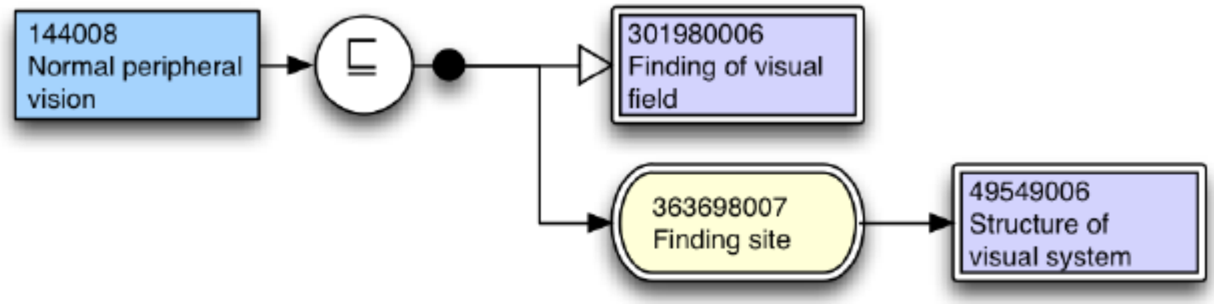


Figure 7: Concept definition diagram for Normal peripheral vision

In this last example, the equivalence of two expressions is defined (as illustrated in Figure 8):

```
( 49601007 |Disorder of cardiovascular system| : 246454002 |Occurrence| = 255407002 |Neonatal| )
=== ( 64572001 |Disease| : 246454002 |Occurrence| = 255407002 |Neonatal| ,
   363698007 |Finding site| = 113257007 |Structure of cardiovascular system| )
```

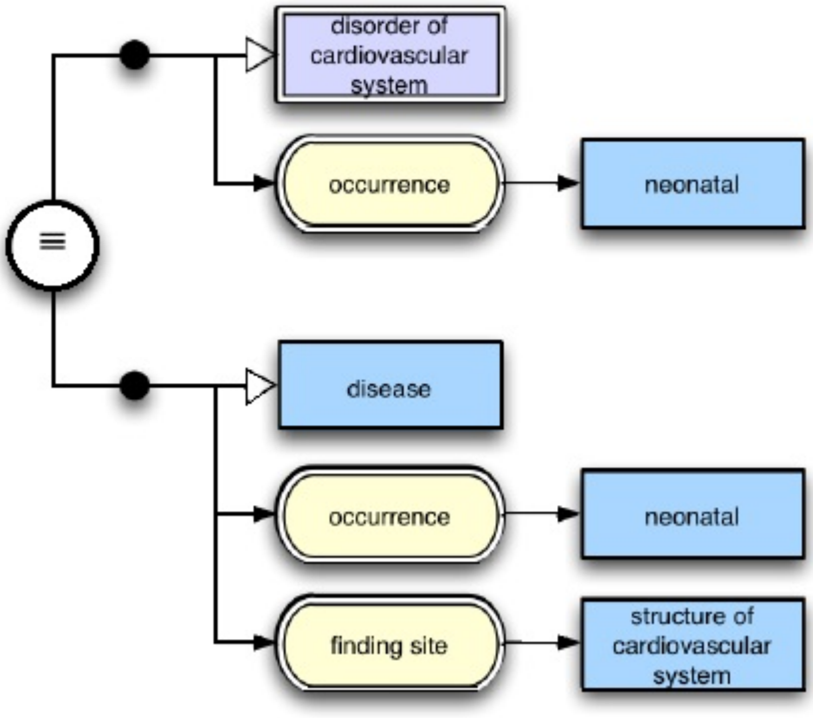

≡
disorder of cardiovascular system
occurrence
neonatal
disease
occurrence
neonatal
finding site
structure of cardiovascular system

Figure 8: Diagram showing the equivalence of two expressions

References

1. *Cooperation Agreement dated July 2013 between the International Health Terminology Standards Development Organisation (IHTSDO) and The Regenstrief Institute Incorporated (RII)*, http://snomed.org/loinc-agreement
2. *HL7 Version 3 Implementation Guide: TermInfo – Using SNOMED CT in CDA R2 Models, Release 1*, HL7 5th DSTU Ballot, January 2014, http://wiki.hl7.org/index.php?title=File:V3_IG_SNOMED_R1_D5_2014JAN.docx.
3. *SNOMED CT Diagramming Guideline (v1.0)*, IHTSDO, 22 November 2013, http://snomed.org/diaguide.
4. SNOMED CT SNOMED International Glossary, Draft version July 2014, http://snomed.org/gl.
5. *SNOMED CT Starter Guide*, IHTSDO, February 2014, http://snomed.org/sghttp://snomed.org/sg.
6. *SNOMED CT Technical Implementation Guide*, IHTSDO, July 2014, http://snomed.org/tig