SNOMED Clinical Terms® Abstract Logical Models and Representational Forms

EXTERNAL DRAFT FOR COMMENT

Version 6b (31-Jan-08)

STATUS

This document contains material useful to technical implementers working to incorporate SNOMED CT into software. However, this current document is a revision of an earlier document which was issued by the SNOMED International Editorial Board (SIEB) as an External Draft. This revision has been issued for review and is therefore subject to revision and correction without notice

This document is intended to assist rather than direct or constrain developers. Implementers who make development decisions based on the advice herein do so entirely at their own risk.

The International Health Terminology Standards Development Organisation and the authors have applied their best endeavors to this document but do not offer any warranty in respect of the accuracy or appropriateness of the information for implementation in any technical environment.

Table of Contents

Document History

Version	Date	Notes
1	07-Mar-05	Initial draft for discussion by Concept Model Design Team
2	19-Apr-05	Revised draft for discussion by Concept Model Design Team
3	31-May-05	Revised and restructured version for Concept Model Working Group
4	09-Jul-05	Corrections and minor revisions agreed by CMWG and approved by TSG and SIEB for release as an "External Draft"
5	20-Jan-06	Revisions based on input from CMWG members and others in review of "External Draft"
6	06-Nov-06	Revision to align with change in name from "context-dependent categories" to "situation with explicit context" and related changes.
6a	31-Jul-07	Document updated to reflect the transfer of SNOMED CT to the International Health Terminology Standards Development Organisation (IHTSDO)
6b	31-Jan-08	Copyright statements updated to 2008

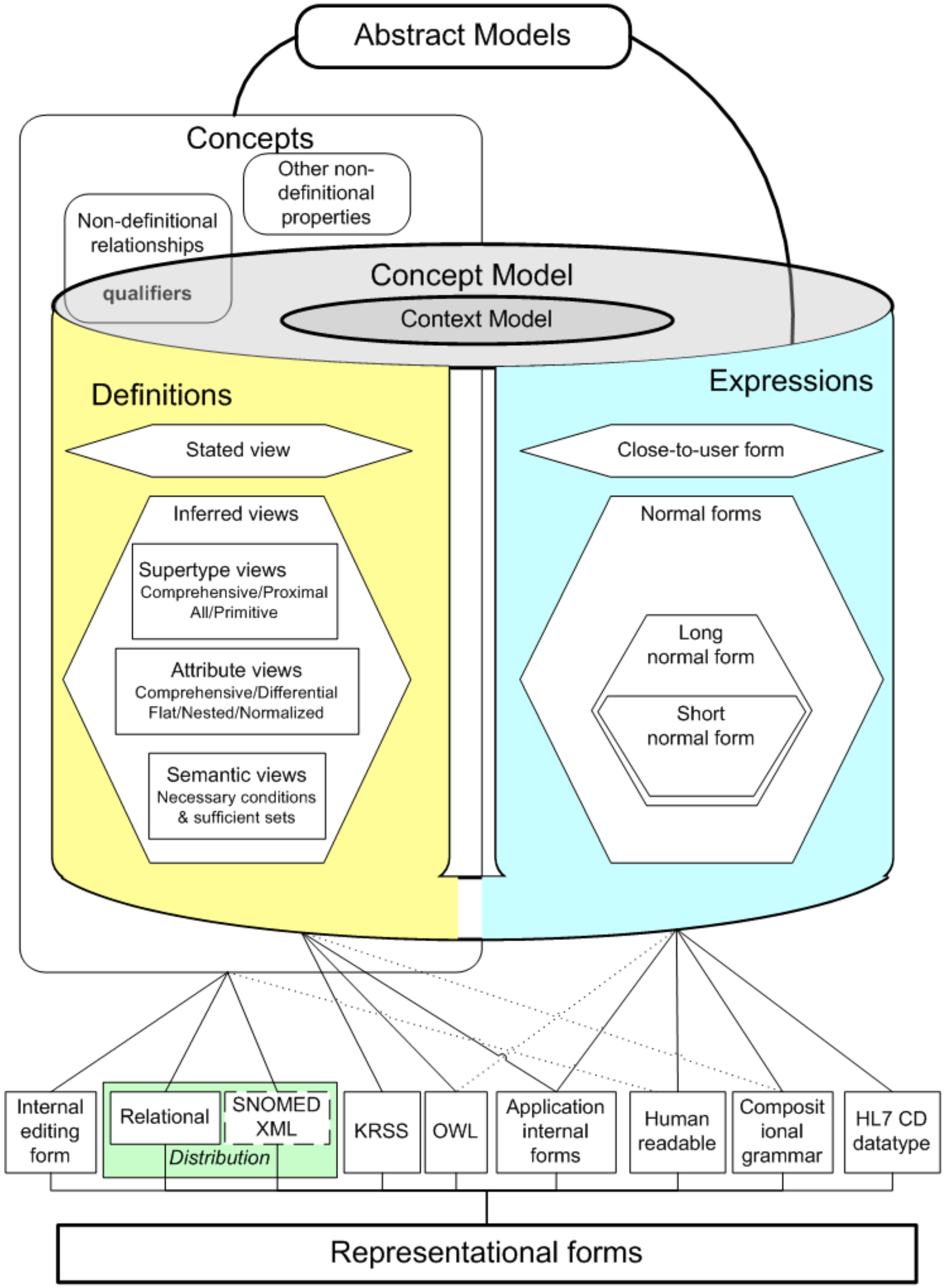


Figure 1. A diagrammatic overview of topics discussed in this document

1 Introduction

This paper considers the various *abstract logical models* and *representational forms* that apply to SNOMED Clinical Terms *concepts* and *expressions*. The intention of the paper is to aid understanding of the various forms frequently discussed by those developing and implementing SNOMED CT. It also provides a draft glossary that is intended to assist communication about these topics.

1.1 Abstract logical models and representational forms

For the purposes of this paper the distinction between an abstract logical model and representational forms is as follows:

Abstract logical model: a description of the manner in which items of information are, or may be, logically assembled independent of the form or syntax in which they are expressed.

Representational form: a physical representation of an abstract logical model used for storing, communicating or presenting information.

1.2 Concepts and expressions

This paper specifies abstract logical models and representational forms that apply to SNOMED CT *concepts* and *expressions*. For the purposes of this paper *concepts* and *expressions* are defined as follows:

Concept: a clinical idea to which a unique *conceptId* has been assigned.

SNOMED CT associates each *concept* with one or more *Descriptions* (which specify terms that may be applied to describe the clinical idea) and one or more *Relationships* to other *concepts* (which logically interrelate the clinical ideas)[1].

Concepts are classes that exist in SNOMED CT. They are referenced by instances within records, documents queries or other resources. Instances of use of a *concept* are referred to as *expressions*.

For example the *concept* "headache (finding)" in SNOMED CT includes:

- A *conceptId* (25064002),
- A set of *descriptions* ("headache", "pain in head", etc.)
- A set of *relationships* ("is a"="pain", "finding site"="head structure", etc.).

Expression: a collection of references to one or more *concepts* used to express an instance of a clinical idea.

This definition includes either a single *conceptId* or a post-coordinated collection of *conceptIds*. The abstract logical model specifies how the members of a post-coordinated collection are related to one another.

For example, a patient record recording that a patient has a headache may include the *conceptId* (25064002), possibly accompanied by one of the terms associated with that *concept* (e.g. "headache"). This is an *expression* because it expresses an instance of the real world phenomenon represented by the referenced *concept* occurring in a particular patient.

[1] A SNOMED CT *concept* may also be associated with or referenced by *CrossMaps* and/or *SubsetMembers* but these aspects are outside the scope of this paper.

1.3 Rendering and logical transformations[2]

Different representational forms can be applied to the information described by the same abstract logical model. This paper refers to the process of applying a representational form to the information in a logical abstract model as **rendering**.

> For example, a simple abstract model that includes an identifier and term can be rendered in various ways such as those shown in the following list:
>
> - 25064002|headache|
> - <code code="25064002" displayName="headache"/>
> - *25064002* headache

SNOMED CT *concepts* are related to one another by *defining relationships*. Each defining relationship specifies something that is necessarily true.

For example, it is necessarily true that the disorder "appendicitis" has "finding site" "appendix structure".

Description logic is applied to SNOMED CT *defining relationships* to ensure that they are logically consistent with one another. It is possible to use *defining relationships* to support logical transformations between different "abstract logical models" which express the same meaning. This paper refers to the process of using logical inference to transform the model as **logical transformation**.

> For example, an abstract logical model representing "headache" using a single *conceptId* (25064002) can be logically transformed to a post-coordinated logical abstract model in which the more general *concept* "aching pain" (7635008), qualified by the attribute "finding site" (363698007) and with the value "head structure" (69536005).

Rendering can be accomplished using only the information in a specific instance of the abstract logical model. In contrast, **logical transformation** also requires access to the *defining relationships* between *concepts* identified in that instance of the abstract logical model.

[2] A detailed guide to transformation of SNOMED CT expressions is also available. See "Transforming Expressions to Normal Forms".

2 Abstract logical models

2.1 Abstract logical model of a concept

The abstract logical model of a SNOMED CT *concept* contains the properties of the *concept* and of its associated relationships and descriptions specified in the distribution tables for SNOMED. Figure 2 provides an overview of the model while Table 1 provides a mapping from the most familiar representational form (the SNOMED CT distribution file structure) to the Abstract Logical Model[3].

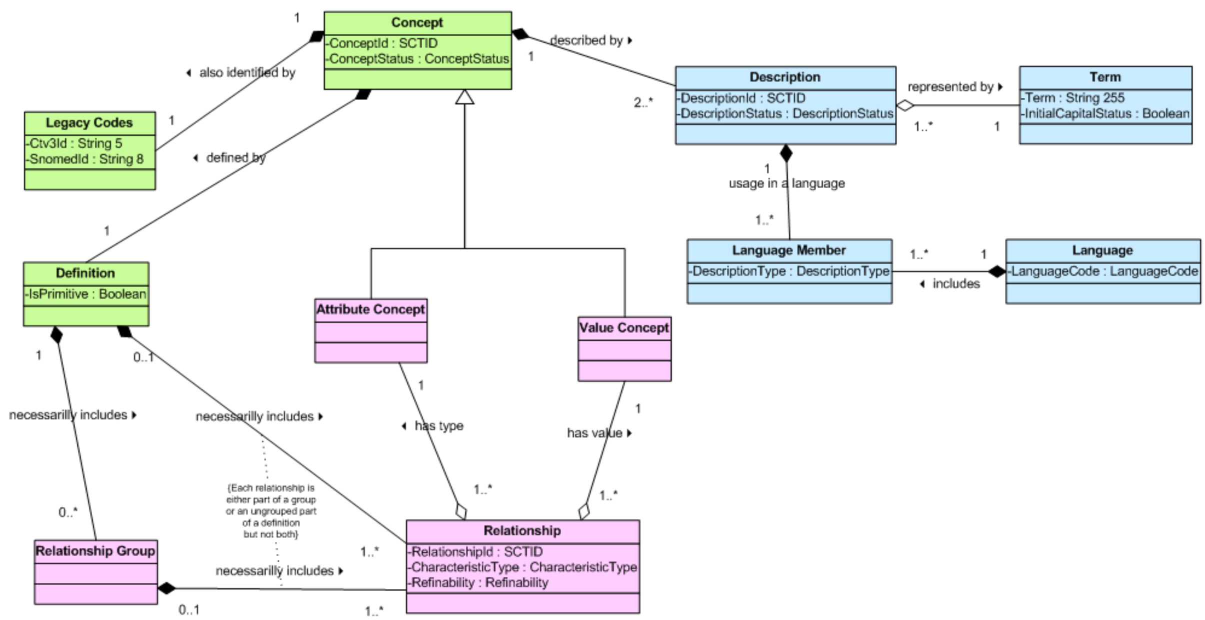


Figure 2. General abstract logical model of a SNOMED CT *concept*

2.1.1 Descriptions and other non-definitional properties of *concepts*

2.1.1.1 Descriptions

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

2.1.1.2 Legacy codes

Legacy identifiers present in the SNOMED CT logical Concept Model include the Ctv3Id (the Read Code from NHS Clinical Terms Version 3) and the SnomedId (the SNOMED code used in SNOMED 3).

2.1.1.3 Cross Maps

Cross Maps to other terminologies and classifications are indirectly a part of the logical SNOMED CT model for *concepts*. However, this aspect of the model is outside the scope of this paper.

[3] A larger scale version of part of Figure 2 is provided in Figure 3.

Table 1. Mapping from Distribution Tables to Abstract Logical Model

Distribution Table	Abstract Logical Model representation
Concepts	
ConceptId	Concept.ConceptId
Conceptstatus	Concept.Conceptstatus
FullySpecifiedName	Concept→Description→Term.Term Where Description→Language Member.DescriptionType=3 in a given language.
Ctv3Id	Concept→Legacy Codes.Ctv3Id
SnomedId	Concept→Legacy Codes.SnomedId
IsPrimitive	Concept→Definition.IsPrimitive
Descriptions	
DescriptionId	Description.DescriptionId
DescriptionStatus	Description.DescriptionStatus
ConceptId	Description→Concept.ConceptId
Term	Description→Term.Term
InitialCapitalStatus	Description→Term.InitialCapitalStatus
DescriptionType	Description→Language Member.DescriptionType
LanguageCode	Description→Language Member→Language.LanguageCode
Relationships	
RelationshipId	Relationship.RelationshipId
ConceptId1	Relationship→Definition→Concept.ConceptId
RelationshipType	Relationship→Attribute Concept.ConceptId
ConceptId2	Relationship→Value Concept.ConceptId
CharacteristicType	Relationship.CharacteristicType
Refinability	Relationship.Refinability

2.1.2 Abstract Logical model of *concept definitions*

The abstract logical model of a *concept definition* comprises the set of relationships which together define that *concept* plus an indication of whether this definition fully-defines the *concept* (i.e. whether the *concept* is primitive or fully-defined).

Figure 3 shows the part of the abstract logical model related to the definition of a *concept*.

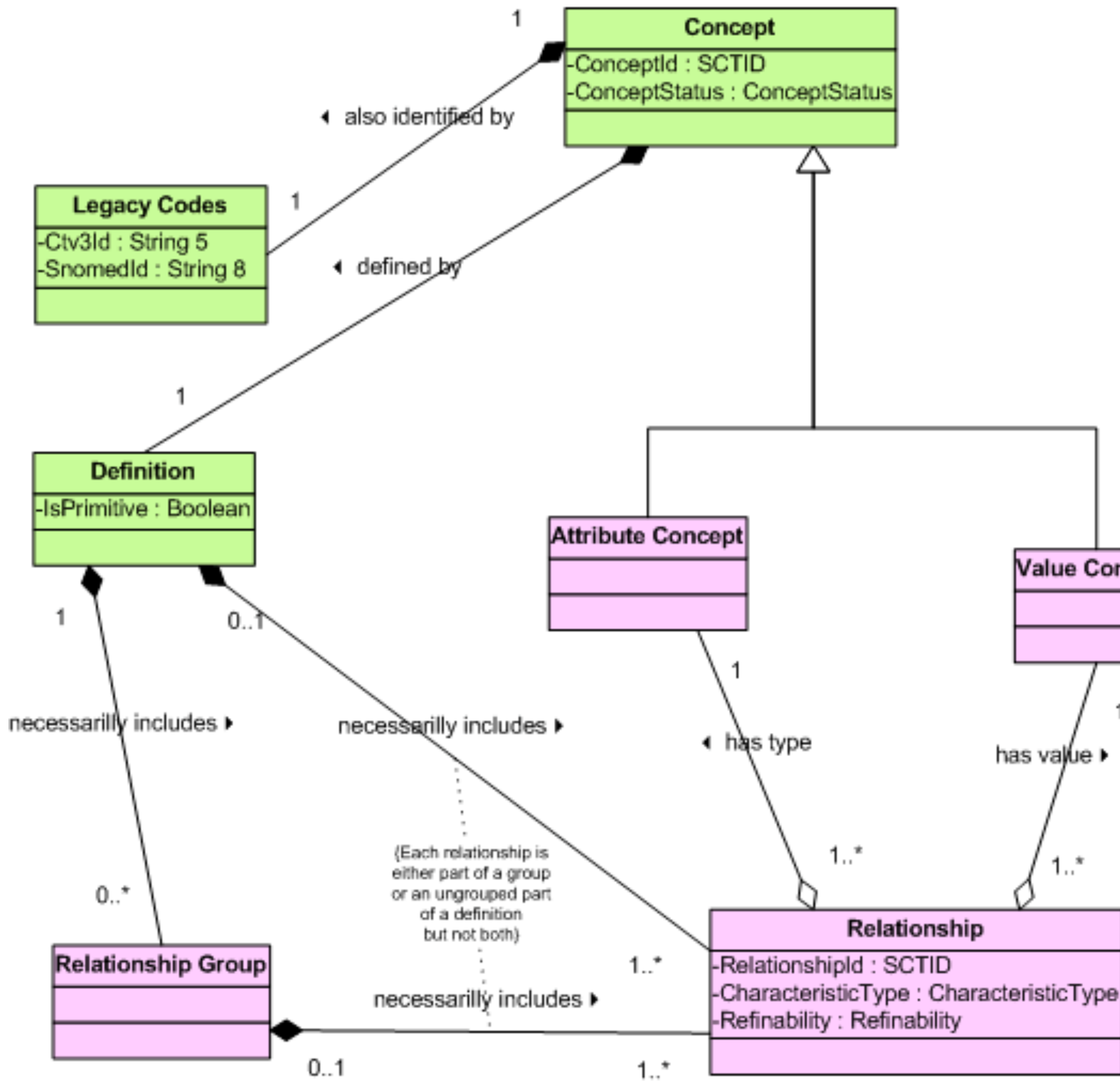


Figure 3. General Abstract Logical Model of a SNOMED CT *concept definition*

2.1.3 Non-definitional relationships

The model used to specify SNOMED CT defining relationships is also used to express other non-defining relationships between *concepts*. These are distinguished by the CharaceristicType attribute.

2.1.3.1 Qualifiers

Qualifiers specify a named relationship that may be used to refine the meaning of a *concept*. Qualifiers specify the allowable value *concept*(s) that can be applied to refine the *concept*.

Qualifiers are not part of the definition of a *concept*. However, qualifiers may be applied to a *concept* as part of an *expression* to refine the meaning of the *concept*.

2.1.3.2 Additional relationships

Additional relationships represent characteristics of a *concept* that are not part of the definition of the *concept*.

Additional relationships may be generally applicable (e.g. "foot" is part of "lower limb") or specific to a particular place or time (e.g. "amoxicillin 250 mg capsules" "have NHS prescribing status" "prescription only medication")

2.1.3.3 Historical relationships

Historical relationships are used to relate inactive *concepts* to active *concepts*.

Historical relationships provide a bridge between *concepts* that are now inactive (and thus not formally defined) and *concepts* that are active. This allows pre-existing records which use the now inactive *concept* to be appropriately interpreted.

Interpretation of the semantics of an inactive *concept* is possible by first traversing the historical relationships and then applying the definition of the appropriate active *concept* before the further processing of the semantics of a *concept definition* or *expression*.

2.2 Alternative logical abstract model views of concept definitions

2.2.1 Introduction

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

For example:

Consider the following set of defining relationships:

"pain in upper limb" "is a" "pain"

"pain in upper limb" has "finding site" "upper limb structure"

"hand structure" "is a" "upper limb structure"

"pain in hand" "is a" "pain"

"pain in hand" has "finding site" "hand structure"

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

"pain in hand" "is a" "pain in upper limb"

The definition of "pain in hand" can thus be viewed in three semantically identical forms:

1. As originally stated:
    - "pain in hand" "is a" "pain" and has "finding site" "hand structure"

or

2. With the additional inferred relationship:
    - "pain in hand" "is a" "pain" and "is a" "pain in upper limb" and has "finding site" "hand structure"

or

3. With the inferred relationship but without the redundant stated relationship "is a" "pain":
    - "pain in hand" "is a" "pain in upper limb" and has "finding site" "hand structure"

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

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

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

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

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

2.2.2 Flat and nested definition views

2.2.2.1 Flat definition views

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

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

For example:

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

"pain in left hand" has "is a" "pain".

"pain in left hand" has "finding site" "structure of left hand".

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

"structure of left hand" "is a" "hand structure"

"structure of left hand" has "laterality" "left".

2.2.2.2 Nested definition views

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

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

For example:

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

"pain in left hand" "is a" "pain"

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

2.2.2.3 SNOMED CT support for flat and nested definition views

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

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

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

For example:

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

2.2.3 Stated and inferred definition views

2.2.3.1 Stated definition view

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

Currently the stated view is not released as part of the standard SNOMED CT distribution.

2.2.3.2 Inferred definition views

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

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

Several semantically identical views may be inferred and these are discussed in section 2.2.4.

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

2.2.4 Alternative inferred definition views

2.2.4.1 Introduction

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

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

Each of the different logical inferred views of *concept definitions* is a combination of a specific *supertype view* (see 2.2.4.2) and an appropriate *attribute view* (see 2.2.4.3)

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

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

Variations in extent may apply independently to supertype ("is a") relationships and to other defining relationships.

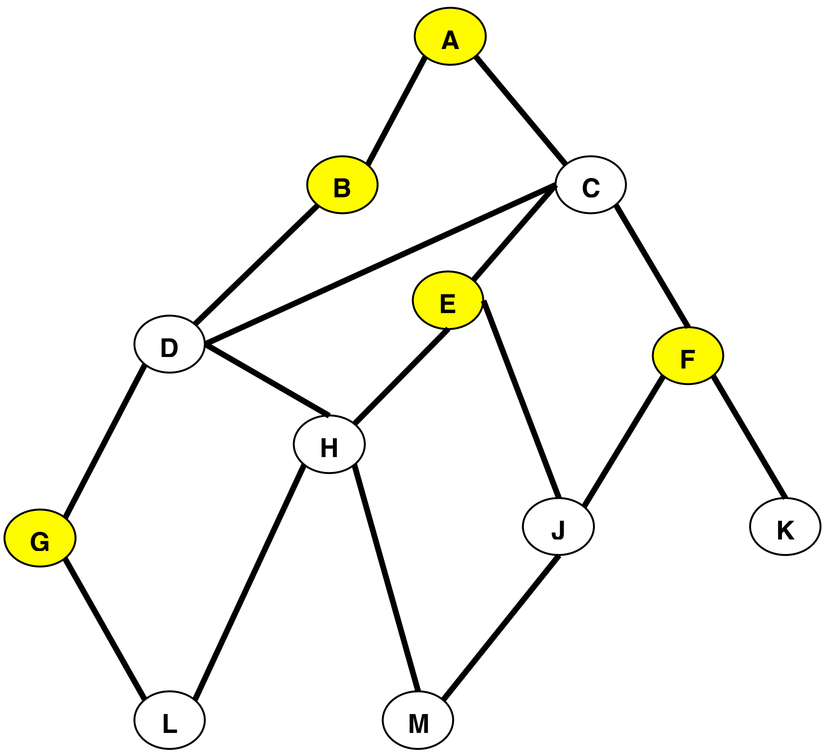
2.2.4.2 Supertype aspects of concept definition views

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

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

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

This is a highly redundant expression of the logical abstract model of a concept definition. Applied to the full content of SNOMED CT it results in tens of millions of relationships[4].



<u>Transitive closure of supertypes for each concept</u>

A → *(none)*

B → A

C → A

D → A, B, C

E → A, C

F → A, C

G → A, B, C, D

H → A, B, C, D, E

J → A, C, E, F

K → A, C, F

L → A, B, C, D, E, G, H

M→ A, B, C, D, E, F, H, J

Figure 4. Example hierarchy with list of supertypes in the transitive closure

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

Other strategies for rapid subsumption testing are discussed in the SNOMED Technical Implementation Guide. However, unless storage capacity is a significant constraint, a pre-computed transitive closure table appears to out-perform other options and is robust, flexible and easy to implement.

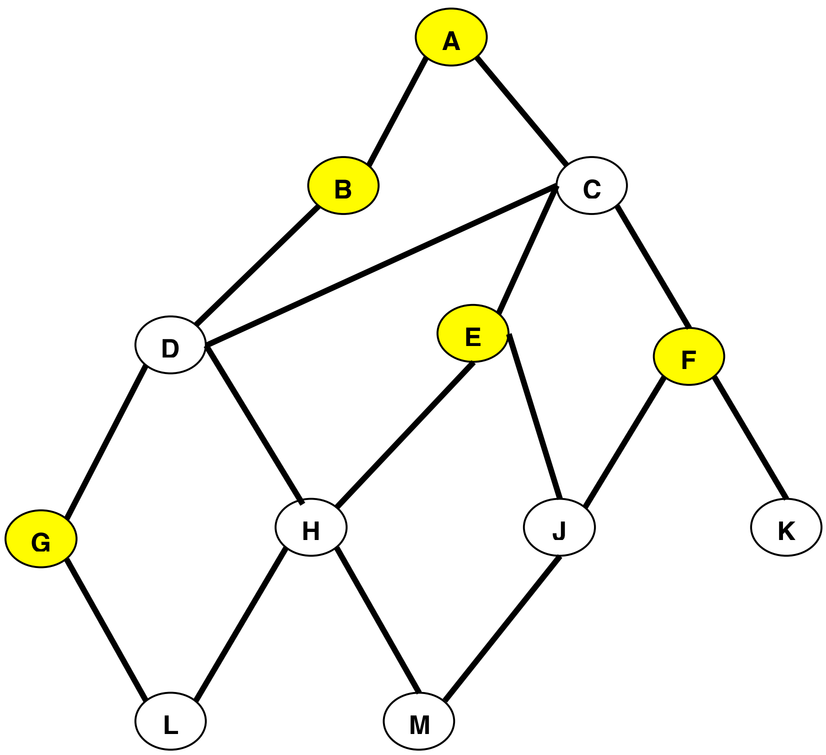
---

[4] At present (January 2006) the transitive closure is not released by SNOMED but can be computed from the set of released "is a" relationships. A proposal to include the pre-computed transitive closure in the SNOMED distribution is currently under consideration.

2.2.4.2.2 Proximal supertype view (standard distribution view)

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

This is the view used in the standard SNOMED CT distribution.



<u>Proximal supertypes for each concept</u>

A → *(none)*
B → A
C → A
D → B, C
E → C
F → C
G → D
H → D, E
J → E, F
K → F
L → G, H
M → H, J

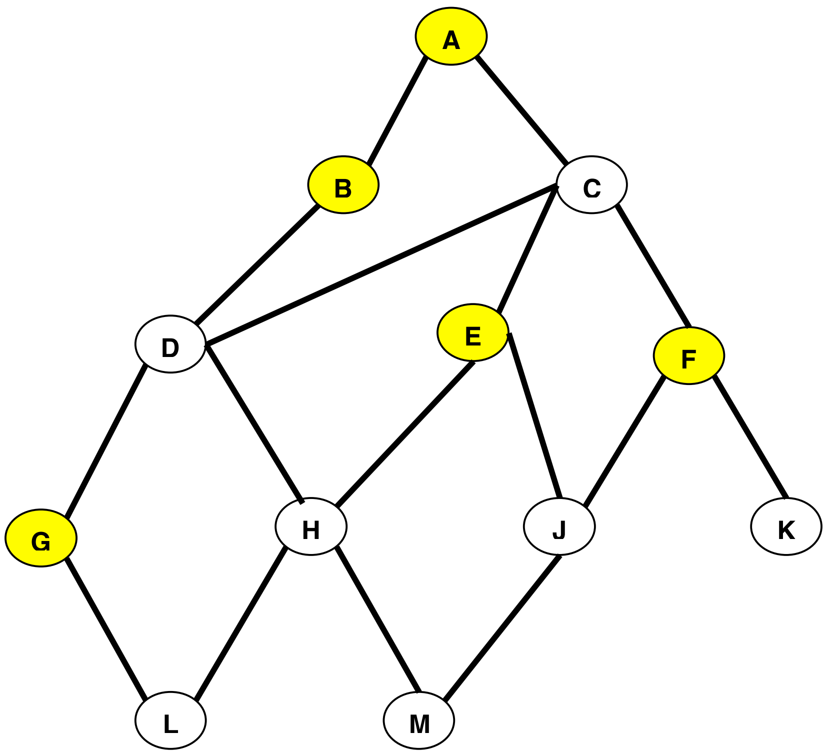
Figure 5. Example hierarchy with list of proximal supertypes

2.2.4.2.3 Comprehensive primitive supertype view

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

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

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



<u>Comprehensive set of primitive supertypes for each concept</u>

A → *(none)*

B → A,(B)[note 1]

C → A

D → A, B

E → A, (E)[note 1]

F → A,(F)[note 1]

G → A, B, (G)[note 1]

H → A, B, E

J → A, E, F

K → A, F

L → A, B, E, G

M→ A, B, E, F

Figure 6. Example hierarchy with comprehensive list of primitive supertypes

Notes

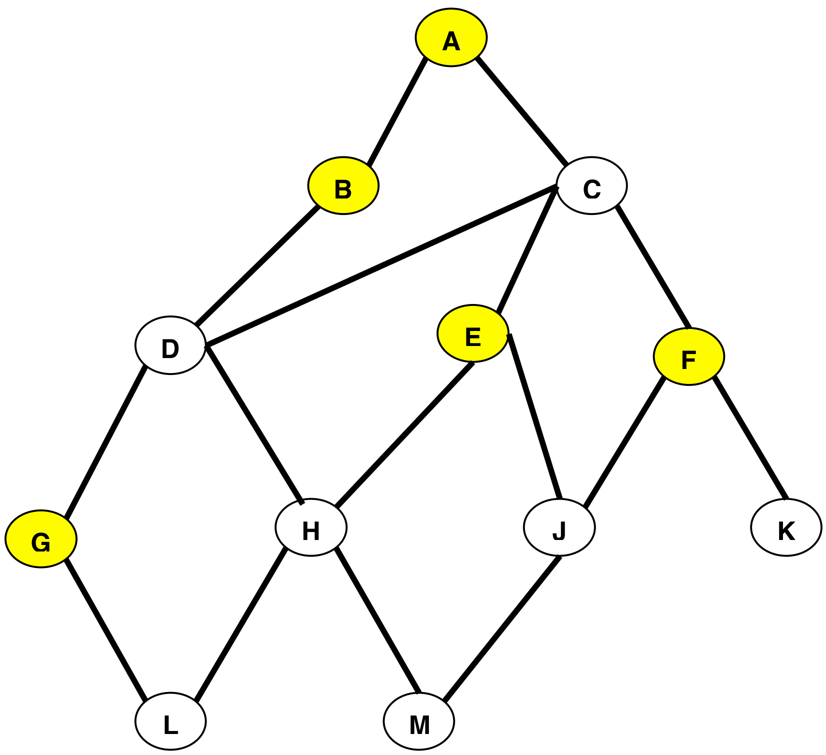
1. In this view the definitions of primitive concepts should implicitly or explicitly include a reference to the defined concept itself. This is because a primitive concept expresses some meaning that is not fully distinguished from its supertypes by other defining relationships. The reference to self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

2.2.4.2.4 Proximal primitive supertypes (short normal view)

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

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

The "is a" relationships in the SNOMED CT ‘canonical table’ represent this view,



<u>Set of proximal primitive supertypes for each concept</u>

A → *(none)*

B → (B or A) [note 1]

C → A

D → B

E → (E or A) [note 1]

F → (F or A) [note 1]

G→ (G or B) [note 1]

H → B, E

J → E, F

K → F

L → E, G

M → B, E, F

Figure 7. Example hierarchy with list of proximal primitive supertypes

Notes

1. For a primitive *concept* two distinct resolutions are possible.
   a. *Treat the concept as its own proximate primitive (recommended)*

      In effect the normal form of a primitive *concept* is itself. It subsumes those *concepts* that are explicitly stated as being its subtypes and any *concepts* or expressions subsumed by those subtype *concepts*. It is subsumed by its proximate supertypes and any *concept* or expression that subsumes one of its supertypes. (e.g. "B is a B", "… "G is a G", etc)

   b. Compute the proximate primitives in the usual way (deprecated)

      In this case any resulting definition cannot be used to compute which *concepts* or expressions are subsumed by this *concept*. This is because the missing characteristic(s) are unknown. (e.g. "B is a A with some unspecified defining characteristic")

2.2.4.3 Attribute aspects of concept definition views

2.2.4.3.1 Introduction

An inferred definition view includes one of several alternative views of the defining characteristics of a *concept*. The considerations in this section exclude the supertype "is a" relationships discussed in section 2.2.4.2.

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

2.2.4.3.2 Comprehensive view of defining relationships

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

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

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

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

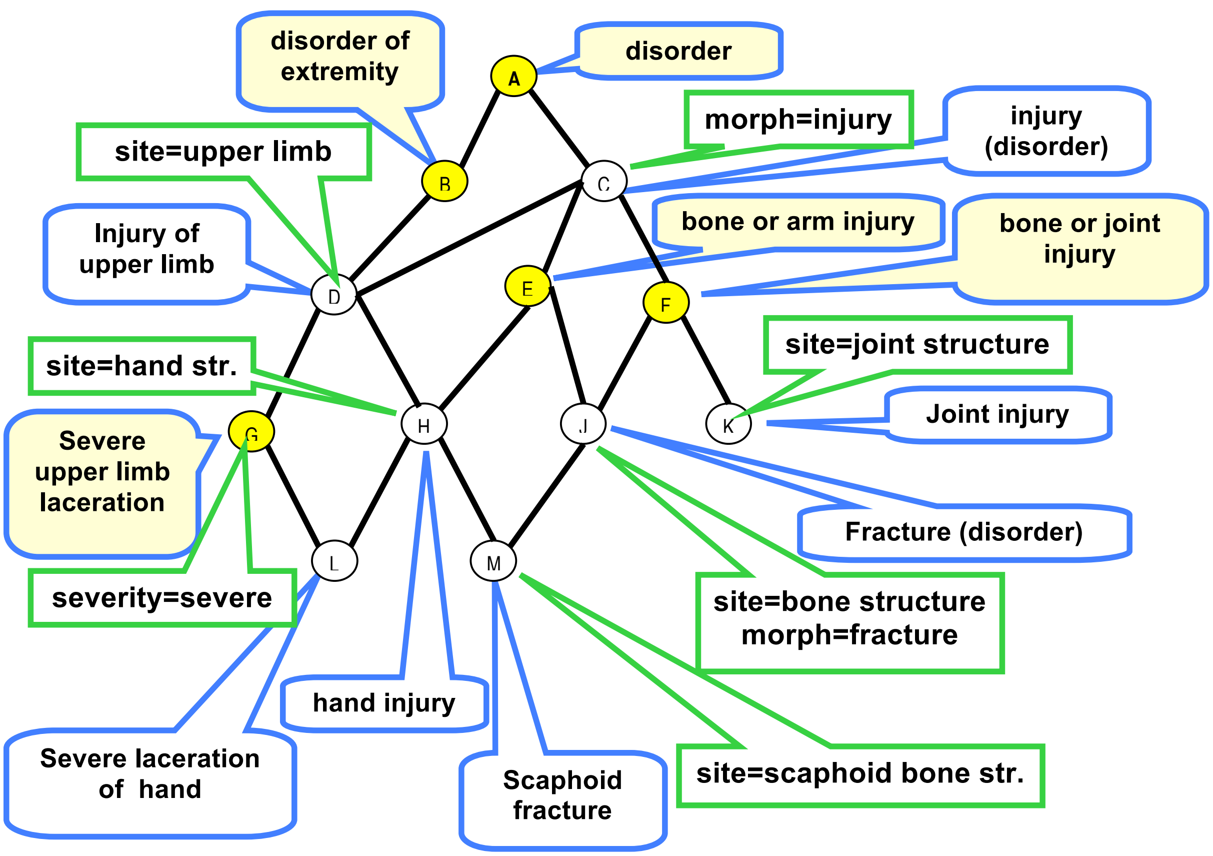


Figure 8. Example showing concepts with differentiating definitions

Table 2. Comprehensive attribute views for definitions illustrated in Figure 8

C. Injury disorder morphology=injury	**D. Injury of upper limb** site=upper limb structure morphology=injury
E. bone or arm injury *(primitive)* morphology=injury	**F. Bone or joint injury** *(primitive)* morphology=injury
G. Severe upper limb laceration *(primitive)* site-upper limb structure morphology=injury severity=severe	**H. Hand injury** site-upper limb structure site=hand structure morphology=injury
J. Fracture (disorder) site=bone structure morphology=injury morphology=fracture	**K. Joint injury** site=joint structure morphology=injury
L. Severe laceration of hand site-upper limb structure site=hand structure morphology=injury severity=severe <u>Note</u> Although the morphology "laceration" is not specified in the example "severe upper limb laceration" refined to the site hand fully defines this concept. In a complete view (including supertypes and attributes) this difference is clear.	**M. Scaphoid fracture** site-upper limb structure site=hand structure site=bone structure site=scaphoid bone structure morphology=injury morphology=fracture

2.2.4.3.3 Non-redundant defining relationships ("distribution view")

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

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

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

Table 3. Non-redundant attribute views for definitions illustrated in Figure 8

C. Injury disorder morphology=injury	**D. Injury of upper limb** site-upper limb structure morphology=injury
E. bone or arm injury *(primitive)* morphology=injury	**F. Bone or joint injury** *(primitive)* morphology=injury
G. Severe upper limb laceration *(primitive)* site-upper limb structure morphology=injury severity=severe	**H. Hand injury** site=hand structure morphology=injury
J. Fracture (disorder) site=bone structure morphology=fracture	**K. Joint injury** site=joint structure morphology=injury
L. Severe laceration of hand site=hand structure morphology=injury severity=severe Note Although the morphology "laceration" is not specified in the example "severe upper limb laceration" refined to the site hand fully defines this concept. In a complete view (including supertypes and attributes) this difference is clear.	**M. Scaphoid fracture** site=scaphoid bone structure morphology=fracture

2.2.4.3.4 Primitive differential attribute view of concept definitions

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

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

Table 4. Primitive differential attribute views for examples in Figure 8

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

Notes

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

2.2.4.3.5 *Supertype differential attribute view of concept definitions*

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

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

Table 5. Supertype differential attribute views for examples in Figure 8

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

2.2.5 Nature of the definition

2.2.5.1 Introducing the idea of necessary and sufficient definitions

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

a) fully-defined concepts
   - The definition is complete. It contains relationships that represent the full set of *necessary* and *sufficient* conditions.

b) primitive concepts
   - The definition is incomplete. It contains relationships that represent a set of *necessary* conditions but this set of conditions is not *sufficient* to fully define the concept.

Notes

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

- For example, “morphology” = “fracture” is a necessary condition of “fracture of femur”.

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

- For example, “morphology” = “fracture” and “finding site” = “bone structure of femur” form a *sufficient* set of conditions that define the concept “fracture of femur”.

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

For example

Consider the concept "gastric ulcer"

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

2.2.5.2 Limitations of the current SNOMED CT model

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

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

For example

Consider the two definitions shown below:

tuberculous arthritis

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

bacterial arthritis

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

The definition of "tuberculous arthritis" differs from that of "bacterial arthritis" in two respects. In practice the first of these (causative agent mycobacterium tuberculosis complex) is sufficient to define the *concept*. However, the nature of the inflammation that results is, necessarily, granulomatous. Thus an expression that specifies "bacterial arthritis" with "causative agent"="mycobacterium tuberculosis complex" would be equivalent to the *concept* "tuberculous arthritis" even though it does not explicitly refine the nature of the inflammation.

<u>Progress Note</u>

As the examples above show, this has semantic significance and this is discussed in more detail in the following sections. Options for distinguishing the sufficient set of defining relationships from those that are merely necessarily true are being investigated.

A complete solution to this issue would need to support the possibility of multiple separate sufficient sets but a more immediate practical solution may recognize only a singe sufficient set. In this case, one option for distribution is to use a different CharacteristicType value to indicate a relationship that is *necessarily* true but not required as part of the *sufficient* set of defining conditions. This could be introduced without other changes to the distribution format. However, before this can be done the implications for classification of SNOMED CT need to be explored. As a result while this is an important issue its resolution is not trivial. Therefore, interim advice on implementing effective logical retrieval within current limitations is provided in section 2.2.5.5.

2.2.5.3 Sufficient definition

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

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

For example:

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

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

Based on this definition:

Any post-coordinated expression that specified a disease involving an ulcer with finding site stomach would be equivalent to or a subtype of "gastric ulcer".

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

2.2.5.4 Necessary definition

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

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

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

For example:

Gastric ulcer could be defined as follows:

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

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

2.2.5.5 Impact on retrieval

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

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

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

Example

The normal form definition of "pulmonary tuberculosis" is as follows:

116680003 | is a | = 64572001 | disease |
,246075003 | causative agent | = 113858008 | mycobacterium tuberculosis complex |
{116676008 | associated morphology | = 6266001 | granulomatous inflammation |
,363698007 | finding site | = 39607008 | lung structure | }

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

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

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

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

116680003 | is a | = 64572001 | disease |
,246075003 | causative agent | = 113858008 | mycobacterium tuberculosis complex |
,363698007 | finding site | = 39607008 | lung structure |

- This correctly subsumes both the pre-coordinated concept "pulmonary tuberculosis" and the post-coordinated candidate expression above.

Advice on specifying retrieval criteria

To ensure complete retrieval

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

<u>Progress note</u>

As noted earlier (see 2.2.5.2), options are being reviewed for explicit support of both *necessary* and *sufficient* sets of definitions. When one of these options is available, the *sufficient* definition of a *concept* should be used as the query predicate. A candidate *expression* matches this predicate if it *necessarily* fulfills all the *sufficient* conditions specified in the query.

2.3 Modeling semantic context

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

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

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

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

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

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

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

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

The following sections explain the way in which context model attributes apply to findings and procedures.

2.3.1 Findings with explicit context

A *finding with explicit context* is a *concept* that applies a particular context to specified associated clinical finding.

For example:

- *Family history of* diabetes mellitus
- *History of* bronchitis
- *Target* weight
- *No* headache
- *No family history of* diabetes mellitus

Findings with explicit context are defined using four attributes.

- Subject Relationship Context
- Temporal Context
- Finding Context
- Associated Finding

The use of these attributes is explained in the following sections.

2.3.1.1 Associated Finding

The associated finding is the finding that is placed in context.

Values

The associated finding may be a subtype of one of the following SNOMED CT concepts

- Clinical finding (404684003)
- Observable entity (363787002) – when associated with an appropriate value
- Measurement procedure (122869004) – associated with an appropriate value
- Link assertion (416698001)

The use of an "observable entity" or "measurement procedure" to label an appropriate value implies a concept which is logically a clinical finding (e.g. "weight" = 70kg).

Examples

Concept	**Associated finding**
family history of diabetes mellitus	diabetes mellitus
past history of bronchitis	bronchitis
target weight	weight[5]
no headache	headache
no family history of diabetes mellitus	diabetes mellitus

5 The concept "target weight" is not currently modeled in this way in SNOMED CT but is shown in this way to illustrate the intended approach to targets and goals.

2.3.1.2 Finding Context

Indicates the primary context of the finding.

Values

known
- **known present** *(default)*
  - definitely present
    - confirmed present
  - probably present
- **known absent**
  - definitely not present
  - probably not present
- **known possible**
  - probably not present
  - probably present
  - suspected

at risk

expectation
- likely outcome
  - impending
  - prognosis context

goal

unknown

Examples

Concept	**Finding context**
family history of diabetes mellitus	known present
past history of bronchitis	known present
target weight	goal[5]
no headache	known absent
no family history of diabetes mellitus	known absent

2.3.1.3 Subject Relationship Context (finding)

Indicates the relationship between the subject of the record in which the *concept* is used and the subject of the associated finding (e.g. "Family history of diabetes mellitus").

Values

person
subject of record *(default)*
member of family
specific family relationships …
donor
fetus
contact
other subjects (e.g. animal contacts, etc)

Examples

Concept	**Subject relationship context**
family history of diabetes mellitus	member of family
past history of bronchitis	subject of record
target weight	subject of record[5]
no headache	subject of record
no family history of diabetes mellitus	member of family

2.3.1.4 Temporal Context (finding)

Indicates a time frame that applies to a *finding with explicit context* (e.g. "past history of bronchitis”).

Values

current or past
- **<u>current or specified</u>** *(default)*
  - current
    - current – specified
    - current - unspecified
  - specified time
    - current - specified
    - past - specified
      - all times past
- **past**
  - past - specified
    - all times past
  - past - unspecified
  - recent

Examples

Concept	Temporal context
family history of diabetes mellitus	current or past[6]
past history of bronchitis	past
target weight	current or specified[7]
no headache	current or specified
no family history of diabetes mellitus	all times past

[6] The current model for family history concepts shows "current of specified" although but should show the higher level value "current of past".

[7] It may seem that a "future" value should exist for goals, risks etc. However, the temporal context binds the goal to the subject at the current time or at a specified time when the goal was set or the risk recognised.

2.3.2 Procedures with explicit context

A *procedure with explicit* is a *concept* that applies a particular context to a specified associated procedure.

For example:

- Hip replacement *planned*
- *History of* vasectomy
- Nutritional assessment *completed*
- Kveim test *not done*

Procedures with explicit context are defined using four attributes.

- Subject Relationship Context
- Temporal Context
- Procedure Context
- Associated Procedure

The use of these attributes is explained in the following sections.

2.3.2.1.1 Associated Procedure

The procedure *concept* that is being placed in context.

Values

The associated procedure must be a subtype of one of the following SNOMED CT concepts

- Procedure (71388002)

Examples

Concept	**Associated procedure**
hip replacement planned	hip replacement
history of vasectomy	vasectomy
nutritional assessment completed	nutritional assessment
Kveim test not done	Kveim test

2.3.2.2 Procedure Context

Indicates the primary context of the procedure.

Values

- **post-starting action status**
  - ended
    - discontinued
    - **done** *(default)*
      - attended
      - performed
    - stopped before completion
      - abandoned
      - suspended
  - in progress
    - not to be stopped
    - started
    - suspended
    - to be stopped
- **pre-starting action status**
  - being organized
    - accepted
    - planned
    - requested
    - scheduled
      - approved and scheduled
  - not to be done
    - cancelled
    - not needed
    - not offered
    - not wanted
    - rejected by performer
    - rejected by recipient
  - organized
  - schedule rejected
  - to be done
  - under consideration
    - needed
    - not wanted yet
    - not yet offered
    - offered
    - wanted
  - was not started
    - cancelled
    - considered and not done
- **not done**
  - did not attend
  - stopped before completion
- **indicated**
- **not indicated**
- **contraindicated**
- **action status unknown**

Examples

Concept	**Procedure context**
hip replacement planned	planned
history of vasectomy	done
nutritional assessment completed	done
Kveim test not done	not done

2.3.2.3 Subject Relationship Context (procedure)

Indicates the relationship between the subject of the record in which the *concept* is used and the subject of the associated procedure. (E.g. "fetal hemoglobin determination" - although not defined using context in current release it probably should be in future).

Values

person
- **<u>subject of record</u>** *(default)*
- **member of family**
  - specific family relationships …
- **donor**
- **fetus**
- **contact**

other subjects (e.g. animal contacts, etc)

Examples

Concept	**Subject relationship context**
hip replacement planned	subject of record
history of vasectomy	subject of record
nutritional assessment completed	subject of record
Kveim test not done	subject of record

2.3.2.4 Temporal Context (procedure)

Indicates the time frame applied to a *procedures with explicit context* (e.g. "past history of cholecystectomy").

Values

current or past
- **current or specified** *(default)*
  - current
    - current – specified
    - current - unspecified
  - specified time
    - current - specified
    - past - specified
      - all times past
- **past**
  - past - specified
    - all times past
  - past - unspecified
  - recent

Examples

Concept	Temporal context
hip replacement planned	current of specified[8]
history of vasectomy	past
nutritional assessment completed	current of specified
Kveim test not done	current of specified

[8] It may seem that a "future" value should exist for plans, requests etc. However, the temporal context binds the plan to the subject at the current time or at a specified time when the plan applied not the time when the action is planned to take place.

2.4 Abstract logical model of an *expression*

Figure 9 shows the general abstract model for a SNOMED CT *expression*.

An *expression* consists of a collection of one or more *concepts* refined by any number of refinements. An expression represents a meaning that is defined by the sum of the definition(s) of the *concepts* as modified by the refinements.

- Each refinement is expressed by an attribute name-value pair.
    - The attribute name must be a *concept* that is a subtype of "attribute".
    - The refinement value may be:
        - A single *concept*.
        - A nested expression.
    - Each refinement acts as an additional defining relationship applied to the definition of the base *concept*.
- Refinements may be grouped to represent interdependencies between them in the same way as relationship groups.

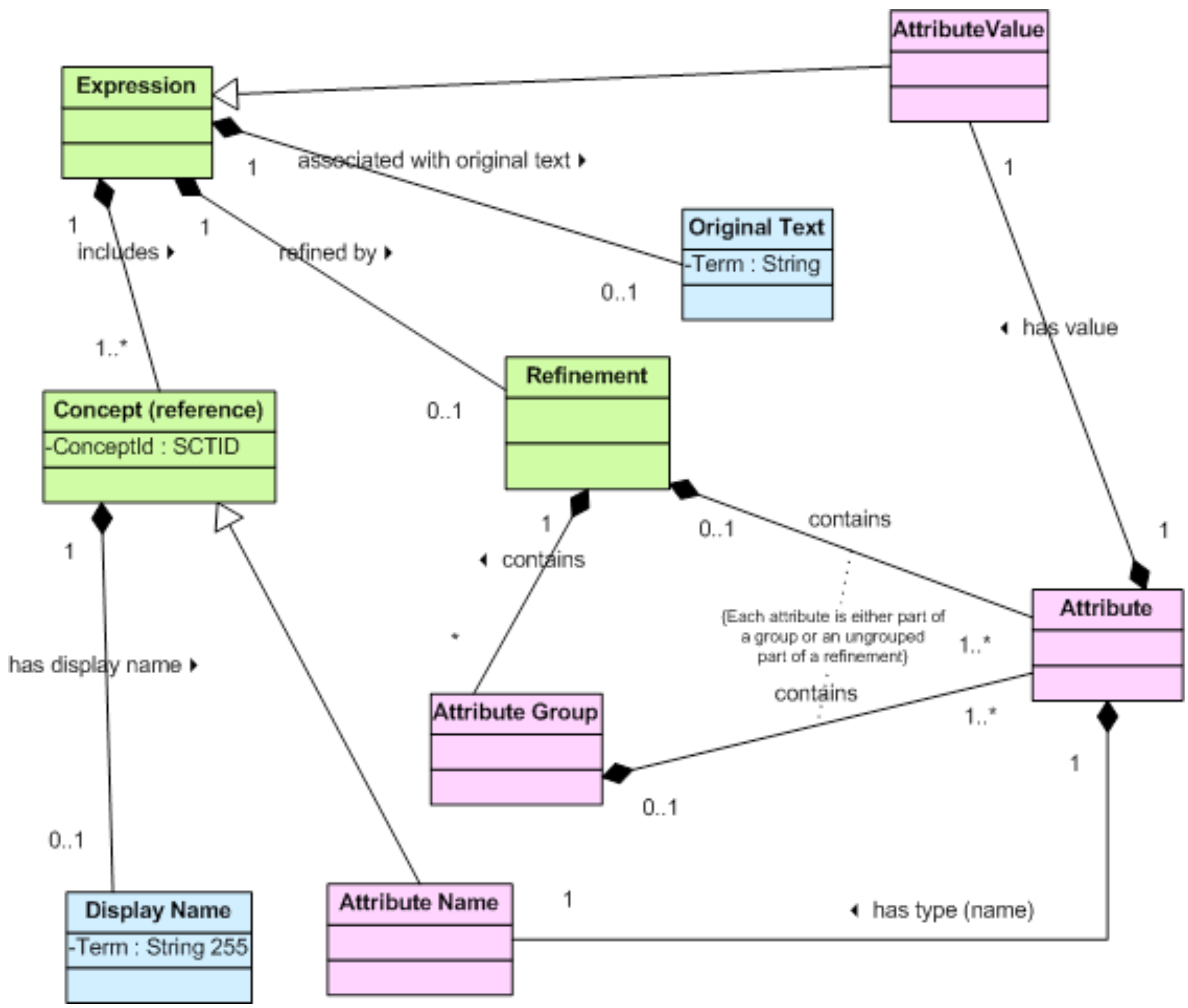


Figure 9. General Abstract Logical Model of a SNOMED CT expression

2.5 Alternative logical abstract model views of expressions

2.5.1 Introduction

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

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

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

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

Note

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

2.5.3 Inferred expression views

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

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

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

In general terms the types of transformation and resulting inferred views for *expression* are similar to those for *concept definitions*. Section 2.5.4 identifies some of the general options for expression views and the differences between *expression* and *concept definitions*.

2.5.4 Simple, nested and grouped expressions

A typical close-to-user entered expression may consist of a single *concept* modified by optional refinements as shown in Figure 10.

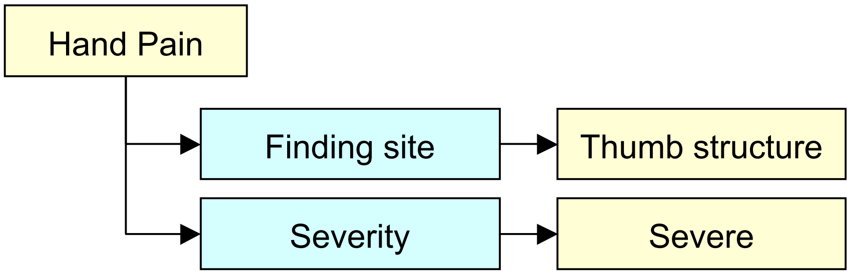


Figure 10. Model of a simple expression instance

This may look like a *concept definition* but it is not defining the concept "hand pain", it is specifying a meaning refined from the definition of "hand pain" by refining the "finding site" and applying a severity qualifier.

The target of a refinement may itself be refined producing a nested structure as shown in Figure 11.

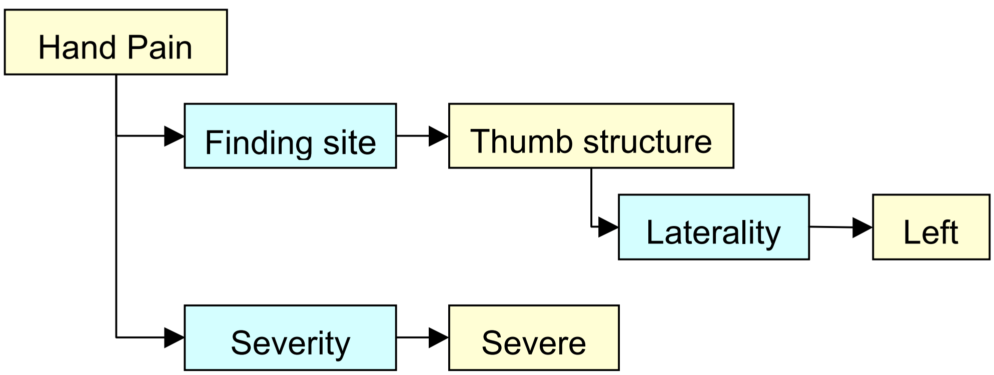


Figure 11. Model of a nested expression instance

In some cases refinements within an expression may be grouped to represent association between a two different refinements. For example, a method and a target site or device as shown in Figure 12.

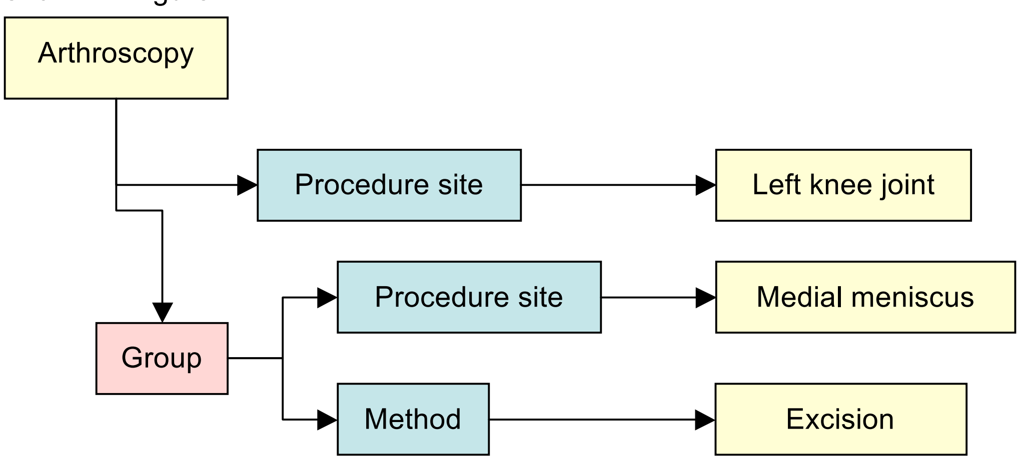


Figure 12. Model of an expression instance with grouping

2.5.5 Expressions with multiple focus concepts

Some expressions may have multiple concepts followed by optional refinements as shown in Figure 13

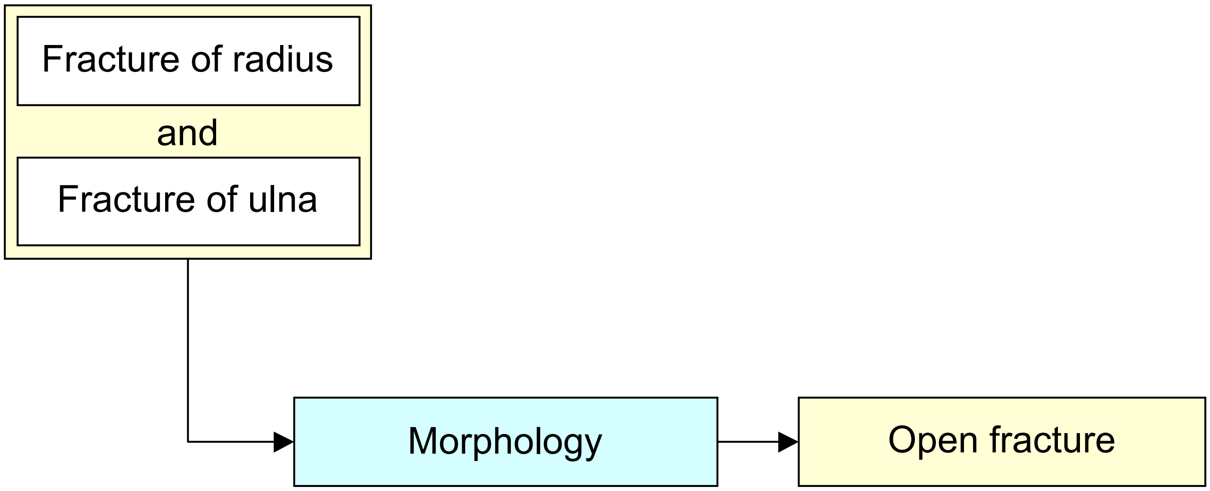


Figure 13. Model of an expression instance with multiple focus concepts

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

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

Some representational forms (e.g. HL7 version 3 Concept Descriptor data type) do not allow combinations to be expressed in this way. However, it is possible to apply a simple logical transformation to create a semantically identical view that can be conveyed in a syntax that supports a single focus concept with refinements (see Figure 14).

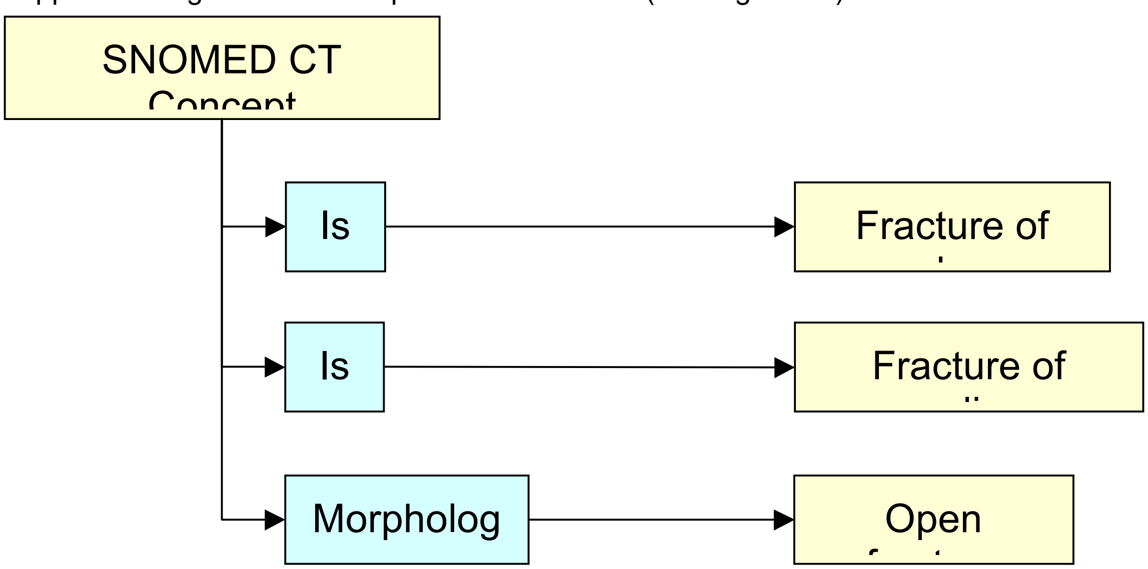


Figure 14. Alternative view of an expression with multiple focus concepts

2.5.6 Expressions that include context

Expressions may also explicitly represent the semantic context surrounding a finding or procedure (see 2.3). In these cases, the finding or procedure is nested inside the context component of the expression. The outer layer of the expression, which expresses the context, is sometime referred to as the "context wrapper". The nested expression representing the finding or procedure is sometimes referred to as the "clinical kernel expression".

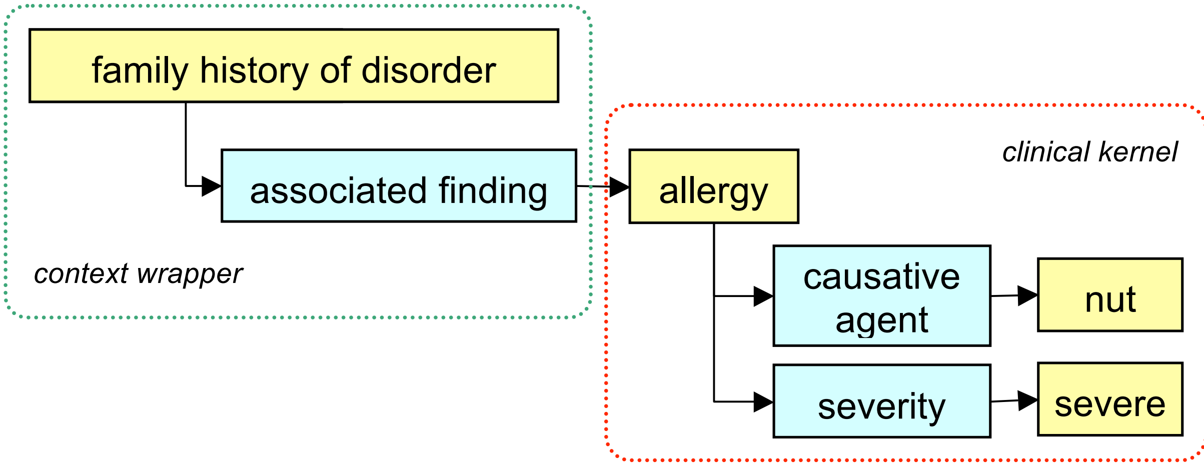


Figure 15. Nested close-to user form view of a context-rich expression

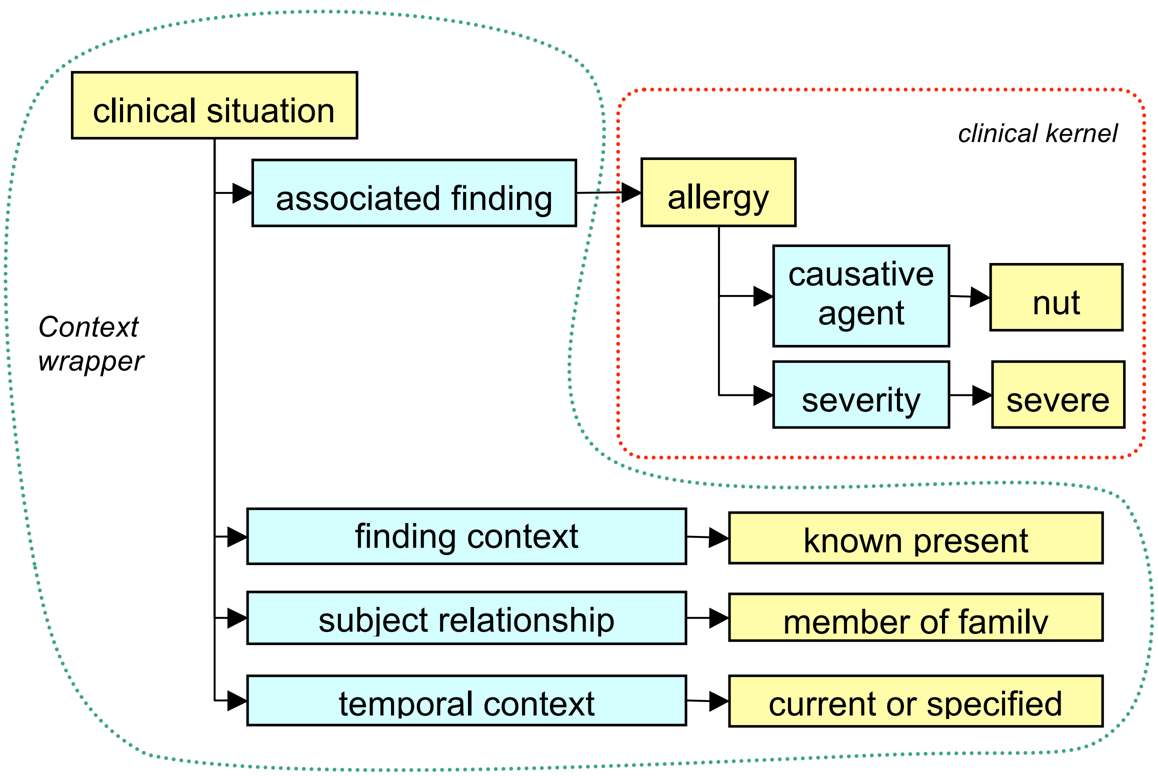


Figure 16. Nested normal form view of a context-rich expression

2.5.7 Transformation between expressions and definitions

A *definition* can be transformed into an expression by making its "is a" related concept(s) the focus of the *expression*.

An expression can also be converted to a definition by reversing this process because the semantic basis of concept *definitions* and *expressions* is identical. However, SNOMED CT *concept definitions* include some properties that are not available in an *expression* (e.g. IsPrimitive, RelationshipId and Refinability)

By applying these transformations *expressions* and *definitions* are directly comparable to determine subsumption and equivalence.

The decision to use an expression or definition model should be determined by the following factors:

- Stated definitions should always follow the definition model.
    - This is because definitions are not constructed based on an identified base *concept* but rather as a set of equally valid relationships.
- Stated (close-to-user) expressions should always follow the expression model.
    - This is because the author of the record is likely to identify a *concept* (or set of *concepts*) first and then apply refinements if these are needed to accurately express the required meaning.
- Inferred definitions or expressions may be represented in whichever model in most appropriate to the required function.
    - In cases where there is a choice between use of a *concept* or a post-coordinated expression, the expression model allows more consistent rendering. For example, nested expressions used as values of an attribute.
- For communication using HL7 messages
    - The *concept* Descriptor (CD) data type supports the *expression* model provided that there is only one focus concept
    - It does not directly support the full *concept definition* model or *expression*s with more than one focus concept. However, as noted earlier an expression with multiple focus concepts can be rendered as SNOMED CT *concepts* with the focus concepts expressed as "is a" refinements.

2.6 Refinement types

Various types of refinement are possible. Of these some are fully supported by the SNOMED CT Concept Model and released data while other possible methods of refinement step outside those boundaries.

2.6.1 Refinement of defining relationships

2.6.1.1 Refinement individual attribute values

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

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

Defining relationships that are marked with the refinability property value "not refinable" should not be refined.

2.6.1.2 Refinement attribute names

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

2.6.1.3 Refinement of defining relationship groups

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

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

2.6.1.4 Nested refinement of defining relationships

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

This occurs most frequently in the following situations:

Laterality qualification

> The laterality qualification literally applies to the value of the "procedure site" or "finding site" relationship.
>
> (Note lateralization is discussed separately)

Context dependent expression

> The "associated finding" or "associated finding" is a "finding" or "procedure" which may itself be refined (e.g. with severity).

Concepts that are defined with references to procedures or findings

> For example "post-therapy radiation sickness" which is defined as "after" "radiation therapy". The nature of the therapy may itself be refined.

2.6.2 Applying values to qualifiers

2.6.2.1 Applying values to individual qualifiers

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

<u>Not refinable</u>

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

<u>Refinable</u>

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

<u>Mandatory to refine</u>

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

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

2.6.2.2 Grouping qualifier refinements

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

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

2.6.2.3 Nested refinement of qualifiers

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

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

2.6.3 Applying laterality to a *concept*

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

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

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

Note

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

2.6.4 Sanctioned and unsanctioned refinement

2.6.4.1 Introduction to refinement sanctioning

As noted in the previous sections SNOMED CT contains some information that may be used to determine which kinds of refinements are likely to be processable to determine equivalence and subsumption. However, there is an open question about whether the absence of a defining or qualifying relationship indicates that refinement is deprecated.

Therefore in some situations it may be useful to consider the use of refinements that are not explicitly sanctioned in the Concept Model.

Progress Note
The question of how far SNOMED should go in providing advice on sanctioning, deprecating or prohibiting particular types of refinement is under discussion. Pending formal advice on this topic readers are referred to the discussion notes in section for 2.6.4.4.

2.6.4.2 Unsanctioned use of "Concept Model attributes"

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

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

Despite this limitation it may be appropriate to use a community agreed approach for a particular defined purposes. However, care should be taken to use attributes only in the manner described in SNOMED CT User Guide.

2.6.4.3 Use of "unapproved attributes"

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

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

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

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

2.6.4.4 Advantage and disadvantages of unsanctioned refinements

[THIS SECTION CONTAINS DISCUSSION NOTES ONLY]

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

Disadvantages of prohibition of all unsanctioned refinements

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

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

Disadvantage of allowing unconstrained refinement

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

- This is still nonsense from a SNOMED CT perspective or perhaps it could correctly mean is "aching pain in the pale colored head structure". However, if the author (or authoring application) assigned such an expression to represent two distinct findings "headache" and "head is pale in color" this meaning would not be apparent from a logical computational perspective.
- While prohibition of nonsense is not tractable it may be feasible to state rules that express which forms of expression are logical and computable. Furthermore the outcome of these rules needs to be deterministic so that the result of transforming do not differ according to implementation.

3. Alternative rational expressions of similar meanings

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

4. User interface design issues

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

Interim recommendations

1. Wherever refining an existing defining or qualifying relationship enables representation of the required meaning this approach should be preferred.
2. Where 1 does not meet the requirements any attribute which is used in the concept model for concepts of the same type may be applied. The value applied to the attribute must be one of the allowable values as specified for that attribute in the current SNOMED CT User Guide.
   - For example a "causative agent" attribute can be applied to a clinical finding concept. The value assigned to this attribute is a value assigned from "organisms", "physical force", "physical object" or "substance". However, "causative agent" cannot be applied to refine a procedure. Furthermore the value of the "causative agent" cannot be a procedure or disorder.
3. Where neither 1 nor 2 meet the requirement use of additional attributes or values may be considered to meet a specific requirement. However, in this case, the implementer and/or user community will need to:
   - Avoid a direct conflict with other uses of the same attribute
     - Ambiguity may arise if an existing attribute that is used in the SNOMED CT concept model is overloaded to fulfil a different use-case.
       - For example: The "laterality" attribute is used in the concept model to specify which of two functionally symmetrical paired structures is involved (e.g. "left wrist", "right kidney"). It should not be used for:
         - non-symmetrical structures (e.g. heart structures where the use of "left" and "right" refers to functionally different structures)
         - right or left side of a midline structure (e.g. "head" :"laterality"="left" does not mean the "left side of the head" it means "left head" – and is thus not a useful refinement)
         - relative laterality (e.g. "trachea" : "laterality" = "left" does not mean "to the left of the trachea" or "trachea deviated to the left" it means "left trachea" – and is not a useful refinement)
   - Agree the approach to be taken in advance
     - Ad-hoc refinement by end-users without any guidance on an agreed approach is liable to lead to multiple ways of representing the same required meaning and a loss of interoperability.
   - Document the approach taken in forms that:
     - Allow consistent use within the community
     - Identify any issues related to computation of equivalence and subsumption between these local variant expressions and the content of SNOMED CT.
     - Are communicated to an appropriate SNOMED Working Group to help establish a wider consensus
     - Make provision for future migration of data as a common SNOMED CT approach is developed in future.

<u>Progress Note</u>

Within the UK, NHS Connecting for Health has issued documentation on post coordination which specifies constraints on allowable refinements and adds some specific extensions to the refinements sanctioned by released relationships. This document is available to implementers in the UK.

2.6.5 Applying values to *concepts*

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

Currently the SNOMED CT Concept Model does not address issues of equivalence between a particular value applied to an observable or measurement procedure and a potentially similar finding (e.g. "creatinine measurement, serum" with a specified value and a finding such as "serum creatinine raised"). However, there is a loose approximation using the "interprets" and "has interpretation" for some concepts.

Progress Note
The relationships between observables and findings are currently under review.

2.6.6 Transformations from close-to-user to a normal or canonical view

The most important requirement for logical transformation of expressions is to enable information entered (in a close-to-user view) to be readily tested for subsumption against concept definitions or other expressions.

A proposed revised normalization algorithm which seeks to address this challenge is the subject of a separate document for discussion.

The idea of the revised algorithm is that it should take as input any valid SNOMED CT *expression* entered by user (or by an application in response to user input) and should return a normalized version of the expression than can be readily tested for subsumption.

Since the simplest case of a valid *expression* is a single *conceptId* with no *qualifications*, the algorithm should also be applicable to *concept definitions* to return logically equivalent normalized expressions.

It should also be applicable to various *expressions* ranging from "parsimonious", through "complete", whether or not they contain "redundant" *expressions* (where a qualification is repeated unnecessarily or is accompanied by another qualification which subsumes it).

The algorithm should always return an identical *expression* when applied to an expression that has already been normalized by one pass of the transformation.

The resulting normalized view should not contain any "fully defined" concepts because all "fully-defined" concepts present in the input expression (or in the definitions of referenced concepts) should also be normalized.

The resulting normalized view should contain any inferable context based either on explicit context in the input expression and any additional surrounding contextual information (e.g. HL7 moodCode).

The proposed algorithm extends earlier work on ‘canonical forms’ as follows:

- Normalizes fully-defined values within definitions or expressions producing nested expressions that are fully normalized.
- Manages close-to-user (stated) expressions that may not reflect the grouping structure of the current (or future) definition of the concept.
- Avoids adding spurious detail from current definitions by allowing the close-to-user (stated) expression to be stored and communicated and canonized based on the best available definitions at any point in time (or a specified historical set of definitions).
- Takes account of context rules including soft default context and a preliminary approach to moodCode mapping and handling of procedures with values (present in algorithm but not yet easily visible in test environment).

<u>Progress Note</u>

Algorithms for transformation from close-to-user expressions to normal form expressions is the subject of a separate guide under discussion.

See - "SNOMED CT Guide to Transforming Expressions to Normal Forms"

3 Representational Forms

3.1 *Concept*

The representation of a *concept* can be considered at two levels.

- Representation of the *concept* as a whole
- Representation of the *concept definition*

3.1.1 Complete *concept* representations

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

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

3.1.1.1 SCT Distribution Files

The distribution files as specified in the SNOMED CT Technical Reference Guide provide a form of representation for complete *concepts* (including associated components).

The distribution files are efficient for large scale batch distribution and facilitate easy import into relational databases.

3.1.1.2 SCT Distribution XML

The XML distribution schema specified by SNOMED provides a form of representation for complete *concepts* (including associated components).

The XML distribution files can be used as an alternative to the SCT Distribution Files. However, they are particularly efficient for communication of individual *concepts* or sets of *concept* (e.g. for update change-sets).

3.1.1.3 Application internal

SNOMED CT enabled applications will usually have their own internal optimized representation of the SNOMED distribution information. This may simply be a relational database with a specified set of indices or it may be a significantly different form.

Examples of proprietary representation include the forms used internally by CliniClue (ClueData), Health Language, Apelon TDE and other implementations.

3.1.1.4 Various human-readable renderings

Concept information may be rendered in various ways to allow human visualization and understanding. These forms may include plain text, mark-up and graphical trees diagramming relationships. All of these renderings can be regarded as representations of complete *concepts* or their definitions.

3.1.2 Concept definition representations

See also: Complete *concept* representations (section 3.1.1)

3.1.2.1 TDE internal

The Apelon TDE editor uses a particular representation of the definitions of a *concept* to optimize modeling and classification of SNOMED CT content.

This is used as an internal tool and is not required for practical use of SNOMED CT content.

This representation theoretically provides ways to convey complete SNOMED CT *concept* representations. However, this requires the use of generic XML tagged properties. In practical terms, some of the non-definitional data is stored separately for reasons of flexibility and performance.

3.1.2.2 KRSS

KSS is a general form for representing logical descriptions.

Transforms have been developed internally for producing KRSS representations of SNOMED CT definitions.

3.1.2.3 OWL

OWL is a web-technology based approach to representation of logical *concept definitions*.

There are several potential ways to use OWL to represent SNOMED CT definitions Transforms have been developed internally for producing OWL representations of SNOMED CT definitions. These are under evaluation and may need revisions to produce an agreed standard form.

The idea of specifying or releasing a standardized SNOMED CT distribution in OWL is under review.

3.1.2.4 Definitions in expression forms

Definitions can be transformed to expressions (see 2.5.7) and the representations applicable to this alternative form can then be applied (see 3.2)

3.1.2.5 Various human-readable renderings

Concept definitions may be rendered in various ways to allow human visualization and understanding. These forms may include plain text, mark-up and graphical trees diagramming relationships. All of these renderings can be regarded as representations of *concept definitions*.

3.2 Representational forms for expressions

3.2.1 SNOMED Compositional Grammar

The SNOMED Composition Grammar is a lightweight syntax proposed for representation SNOMED CT *expression*. It is currently used for demonstration purposes and has been proven to be both human readable and machine parsable.

Table 6 shows formal definition of the syntax using modified Bachus-Naur Form (BNF) in its simplest uncommented form. The same information is presented with detailed comments on each of the elements in Table 7 and Table 8. Examples of the grammar are shown in Table 9.

Table 6. BNF definition of Compositional Grammar (without comments)

```
expression::= concept [ws "+" concept ]* [":" ws refinements ws]
concept::= ws conceptId ws ["|" ws term ws "|" ws]
conceptId::= digitNonZero digit digit digit digit digit+
term::= nonws [nonpipe* nonws]
refinements::= attributeSet ws attributeGroup*
attributeGroup::= "{" ws attributeSet ws "}" ws
attributeSet::= attribute ws ("," ws attribute ws)*
attribute::= attributeName ws "=" attributeValue
attributeName::= ws attributeNameId ws ["|" ws term ws "|" ws]
attributeValue::= concept | (ws "(" expression ")" ws)
attributeNameId::= digitNonZero digit digit digit digit digit+
ws: ( space | tab | cr | lf )*
nonws::= <any character other than those defined in ws and not "|">
digit::= ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")
digitNonZero::= ("1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")
nonpipe::= <any character other than "|">
space::= &H20
tab::= &H09
cr::= &H0C
lf::= &H0A
```

Table 7. BNF representation of Compositional Grammar (detail)

expression::= concept [ws "+" concept]* [":" ws refinements ws]	
	An expression supports combinations of one or more *concepts* optionally refined by a set of refinements. The meaning of the expression is a subtype of all the *concepts* constrained by the set of refinements. Note that where there is a requirement for multiple separately qualified concepts to be present these are expressed in attribute groups within a refinement of a general concept such as "situation with explicit context".
concept::= ws conceptId ws ["|" ws term ws "|" ws]	
	A *concept* is represented by a *concept*Id optionally followed by a term enclosed by a pair of "|" characters. Whitespace before or after the *concept*Id 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.
conceptId::= digitNonZero digit digit digit digit digit+	
	The *concept* id must be a valid SNOMED CT identifier for a *concept*.
term::= nonws [nonpipe* nonws]	
	The term must be the term from a SNOMED CT description that is associated with the *concept* identified by the preceding *concept* identifier. The term may include any valid UTF-8 character except for the pipe "|" character. The term begins with the first non-whitespace character following the starting "|" character. The term ends with last non-whitespace character following preceding the next "|" character.
refinements::= attributeSet ws attributeGroup*	
	A refinement contains all the grouped and ungrouped attributes that refine the meaning of the containing expression.
attributeGroup::= "{" ws attributeSet ws "}" 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 ws)*	
	An attribute set contains one or more attribute name-value pairs expressing refinements.
attribute::= attributeName ws "=" attributeValue	
	An attribute name-value pair expressing a single refinement of the containing expression.

Table 8. BNF representation of Compositional Grammar (detail continued)

<table>
<tr><td colspan="2">attributeName::= ws attributeNameId ws ["|" ws term ws "|" ws]</td></tr>
<tr><td></td><td>The name (or relationship type) of an attribute to which a value is applied to refine the meaning of a containing expression. The attribute name is represented by an appropriate conceptId optionally followed by a term enclosed by a pair of "|" characters.
Whitespace before or after the conceptId is ignored as is any whitespace between the initial "|" characters and the first non-whitespace character in the term or between the last non-whitespace character and before second "|" character.</td></tr>
<tr><td colspan="2">attributeValue::= concept | (ws "(" expression ")" ws)</td></tr>
<tr><td></td><td>A concept or expression representing the value of a named attribute which refines the meaning of a containing expression. If an expression is used this must be enclosed in brackets.</td></tr>
<tr><td colspan="2">attributeNameId::= digitNonZero digit digit digit digit digit+</td></tr>
<tr><td></td><td>The attribute name id is must be the conceptId for a concept that is a subtype descendent of the SNOMED CT concept "attribute".</td></tr>
<tr><td colspan="2">ws: (space | tab | cr | lf)*</td></tr>
<tr><td></td><td>Whitespace characters (space, tab, linefeed and carriage return) are ignored everywhere in the expression except:
a. Whitespace within a conceptId or attributeNameId is an error.
o Note: whitespace before or after the last digit of a valid identifier is ignored.
b. Whitespace within a term is treated as a significant character of the term.
o Note whitespace before or after the last non-whitespace character of a term is ignored</td></tr>
<tr><td colspan="2">nonws::= <i><any character other than those defined in</i> ws <i>and not</i> "|"></td></tr>
<tr><td></td><td>Non whitespace includes all characters other than the pipe character "|" and those characters specified as whitespace (ws).</td></tr>
<tr><td colspan="2">digit::= ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")</td></tr>
<tr><td></td><td>The digit constraint applies to all characters in a concept identifier.</td></tr>
<tr><td colspan="2">digitNonZero::= ("1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")</td></tr>
<tr><td></td><td>The first character of a concept identifier is constrained to a digit other than zero.</td></tr>
<tr><td colspan="2">nonpipe::= <any character other than "|"></td></tr>
<tr><td></td><td>The characters of term are constrained to be any character other than a "|".</td></tr>
<tr><td colspan="2">space::= &H20</td></tr>
<tr><td colspan="2">tab::= &H09</td></tr>
<tr><td colspan="2">cr::= &H0C</td></tr>
<tr><td colspan="2">lf::= &H0A</td></tr>
<tr><td></td><td>Space, tab, carriage return and linefeed characters as defined by hexadecimal notation.</td></tr>
</table>

Table 9. Examples of SNOMED CT Compositional Grammar

```
195967001|asthma|:246112005|severity|=24484000|severe|
```

```
52734007|total replacement of hip|:272741003|laterality|=7771000|left|
```

```
243796009|situation with explicit context|:
{363589002|associated procedure|
      =(397956004|prosthetic arthroplasty of the hip|
         :363699004|direct device|=304120007|total hip replacement prosthesis|
          {363704007|procedure site|
             =(24136001|hip joint structure|:272741003|laterality|=7771000|left|)
          ,260686004|method|=257867005|insertion - action|})
,408730004|procedure context|=385658003|done|
,408731000|temporal context|=410512000|current or specified|
,408732007|subject relationship context|=410604004|subject of record|
}
```

3.2.2 HL7 Concept Descriptor data type

Figure 17 Schema representation of CD datatype *(pre-adopted <group> element)*

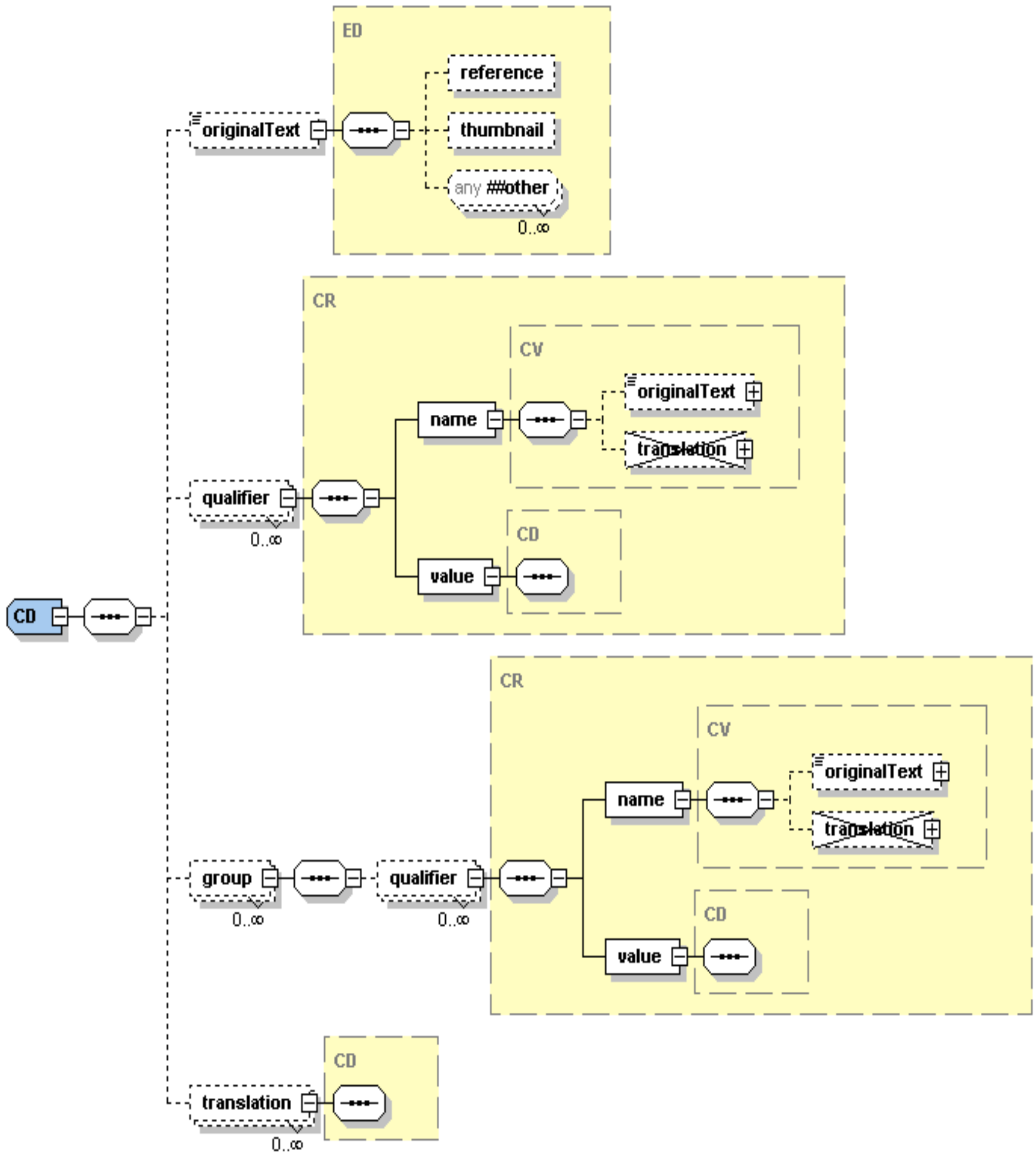


Figure 18. Examples of HL7 CD datatype with SNOMED CT encoding


```
<code codeSystem="2.16.840.1.113883.6.96" code="195967001" displayName="asthma">
<qualifier>
  <name code="246112005" displayName="severity"/>
  <value code="24484000" displayName="severe"/>
</qualifier>
</code>
```

```
<code codeSystem="2.16.840.1.113883.6.96" code="52734007" displayName="total replacement of hip">
<qualifier>
  <name code="272741003" displayName="laterality"/>
  <value code="7771000" displayName="left"/>
</qualifier>
</code>
```

```
<code codeSystem="2.16.840.1.113883.6.96" code="243796009" displayName="situation with explicit context">
 <originalText>total replacement of hip, left hip joint structure</originalText> <group>
  <qualifier>
   <name code="363589002" displayName="associated procedure"/>
   <value code="397956004" displayName="prosthetic arthroplasty of the hip">
    <qualifier>
     <name code="363699004" displayName="direct device"/>
     <value code="304120007" displayName="total hip replacement prosthesis"/>
    </qualifier>
   <group>
      <qualifier>
        <name code="363704007" displayName="procedure site"/>
        <value code="24136001" displayName="hip joint structure">
         <qualifier>
          <name code="272741003" displayName="laterality"/>
          <value code="7771000" displayName="left"/>
         </qualifier>
        </value>
      </qualifier>
      <qualifier>
        <name code="260686004" displayName="method"/>
        <value code="257867005" displayName="insertion - action"/>
     </qualifier>
    </group>
   </value>
  </qualifier>
  <qualifier>
   <name code=" 408730004" displayName="procedure context"/>
   <value code=" 385658003" displayName="done"/>
  </qualifier>
  <qualifier>
   <name code="408731000" displayName="temporal context"/>
   <value code="410512000" displayName="current or specified"/>
  </qualifier>
  <qualifier>
   <name code="408732007" displayName="subject relationship context"/>
   <value code="410604004" displayName="subject of record"/>
  </qualifier>
 </group>
</code>
```

3.2.3 Comparison of SNOMED Compositional Grammar and HL7 CD Datatype

The SNOMED CT compositional grammar and the HL7 Concept Descriptor data type are both useful as machine readable representations.

3.2.3.1 Advantages and disadvantages of SNOMED CT Compositional Grammar

The main advantages of the compositional grammar for representing SNOMED CT expression are as follows:

- Supports SNOMED CT pre and post-coordinated expressions
- Compact
  - Allows a complex expression to be represented as briefly as possible
    - With or without inclusion of terms.
  - A minimum of syntactic noise to distract human readers from the content when reviewing an expression.
  - Parsable using any industry standard BNF aware parser.
- Simplest expression is intuitive
  - A single conceptId without any surrounding syntax is a valid expression in SNOMED CT compositional grammar
- Specific and extensible
  - Although the compositional grammar has been stable for several years, the fact that the specification is owned by SNOMED means that experimental extensions are possible to meet emerging needs.

The main disadvantages of the compositional grammar for representing SNOMED CT expression are as follows:

- Not an industry standard like XML
  - This means that ubiquitous industry standard XML tools cannot be directly applied to parse, validate or transform expressions.
- Specificity
  - Communications that include codes from several different code systems may be more effective is a common syntax is used rather than a specific one for each of the code systems supported.

3.2.3.2 Advantages and disadvantages of HL7 Concept Descriptor

The main advantages of the HL7 CD data type for representing SNOMED CT expression are as follows:

- Supports SNOMED CT pre and post-coordinated expressions
- Based on XML
  - This means that ubiquitous industry standard XML tools can be directly applied to parse, validate or transform expressions.
- General purpose standard
  - Communications that include codes from several different code systems can be conveyed using the same syntax.

The main disadvantages of the compositional grammar for representing SNOMED CT expression are as follows:

- Verbose
  - Even the simplest SNOMED CT expression consisting of a single ConceptId requires an XML Element containing two attributes one of which identifies the code system and the other containing the ConceptId.
  - Expressions with multiple refinements, grouping and nesting can be very hard to read.
  - XML and HL7 data type schema add syntactic noise which is not essential and distracts from the underlying content.
- General purpose standard
  - Any extensions to SNOMED requirements for expression representation are subject to the processes of another standards body (i.e. HL7). While this is appropriate for some use-cases it is may limit flexibility as new requirements arise.

3.2.3.3 Conclusions on SNOMED Compositional Grammar and HL7 CD Datatype

The SNOMED compositional grammar and the HL7 CD datatype are both valuable and it is possible to transform from one form to the other without loss of information. The distinct features the two forms makes each of them suitable for a different profile of uses as indicated below.

The SNOMED CT compositional grammar is preferred for internal representations of SNOMED CT expressions. Its compact but human readable form make it ideal for:

- Documentation – of definitions, expressions and transformations
- Review of modeling issues
- Review of results of alternative transformations.
- Indexing expressions in reference databases

The HL7 Concept Description data type is preferred for standard message communication. The use of XML and the fact that this is a common form used for communication of coded data from many different code systems make it appropriate for:

- Communication of SNOMED CT encoded clinical information in healthcare messages.
- Communication with HL7 Common Terminology Service providers.

3.2.4 Expression in definition forms

An expression can be transformed to definition form (see (see 2.5.7) and the representations applicable to this alternative form can then be applied (see section 3.1.2). However, this approach is limited because several of the forms used to represent concept definitions do not support nesting.

3.2.5 Human-readable renderings

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

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

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