

SNOMED Clinical Terms® Reference Sets

July 2006

Technical specification

IMPORTANT NOTE

THIS DOCUMENT IS AN *EXTERNAL DRAFT FOR DISCUSSION*. IT HAS BEEN APPROVED FOR CIRCULATION FOR COMMENT ONLY.

THIS DOCUMENT WILL BE AMENDED FOLLOWING CONSULTATION AND SHOULD NOT BE USED AFTER THE CLOSE OF THE CONSULTATION PERIOD.

SNOMED®, SNOMED® International, SNOMED RT® and SNOMED CT® are registered trademarks of the College of American Pathologists. All rights reserved. ©2002-2005 College of American Pathologists. SNOMED CT has been created by combining SNOMED RT and Clinical Terms Version 3, formerly known as Read Codes, which was created on behalf of the UK Department of Health and is a Crown copyright.

SNOMED RT® is a copyrighted work of the College of American Pathologists. © 2000 College of American Pathologists

Clinical Terms Version 3 was created on behalf of the United Kingdom's Department of Health and is a Crown Copyright.

LOINC® is a registered trademark of the Regenstrief Institute. Copyright 1995-2005, Regenstrief Institute and the Logical Observation Identifier Names and Codes (LOINC®) Committee. All rights reserved.

Windows®, Microsoft® Word, and Microsoft Visual Basic® are registered trademarks of the Microsoft® Corporation in the United States and/or other countries.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Table of Contents

Table of Contents	3
Document History	5
1 Introduction	6
1.1 Purpose	6
1.2 Who should read this guide?	6
1.3 Status	6
1.4 Feedback.....	7
2 Introducing Reference Sets.....	8
2.1 Overview.....	8
2.2 Basic principles.....	8
2.2.1 Improve the internationalization of the SNOMED CT design	8
2.2.2 Increase the support for SNOMED CT localization	8
2.2.3 Enhance change management support	8
2.2.4 Facilitate the use of Refset groups.....	9
2.3 Background	9
2.4 Proposed Reference set types	10
2.5 Relational representation of RefSets – summary	10
3 RefSets.....	11
3.1 Overview.....	11
3.2 The RefSet table structure.....	11
3.3 The RefSet members table structure	14
4 Simple Reference Sets.....	15
4.1 Overview.....	15
4.2 Relational representation of Simple Reference set members.....	15
4.3 History mechanism	17
RefSet groups	18
4.4 Overview.....	18
4.5 Relational representation of RefSet groups	18
4.6 History mechanism	18
5 Tagged RefSets	19
5.1 Overview.....	19
5.2 Relational representation of Tagged RefSets	19
5.3 History mechanism	19
6 Language Reference Sets	20
6.1 Overview.....	20

6.2	Requirements	20
6.2.1	Requirement summary	20
6.2.2	Current approach	20
6.3	Language Reference set member table structure - summary	21
6.4	Proposed changes to the Descriptions table	24
6.5	History mechanism	25
7	Navigation Reference Sets	27
7.1	Overview	27
7.2	Requirements	27
7.3	Navigation Reference set member table structure - summary	27
7.4	Navigation Reference Set example	28
8	Aggregation Reference Sets	30
8.1	Overview	30
9	Prioritized Reference Sets	31
9.1	Overview	31
9.2	Proposed Prioritized Reference set member table structure - summary	Error!

Bookmark not defined.

Document History

The SNOMED CT Reference Set specification document would be released as a draft specification for external comment in October 2005.

Version Control

Version Date	Notes	Editor
20040521	Original draft based on TWG proposals	SC
20050113	Revision of original draft with additional notes drawn from TWG minutes	SC
20050424	Changes from the TWG meeting of 04/11/2005	SC
20050524	Changes from the TWG meeting of 04/28/2005	SC
20050602	Language RefSet Proposal derived from original generic RefSet proposal	GR
20050606	Structure overview diagram added, correction of RefSetVersionId field name, inclusion of a new field (priority)	GR
20050614	Changes from TWG meeting of 06/06/2005 and subsequent discussion	SC
20050824	Changes from TSC meeting of 06/15/2005 to Language Refsets	GR
20050906	Added chapters for simple and navigation RefSets.	GR
20050926	Changes from TWG meeting of 09/07/2005	GR
20051005	Changes for Editorial Board Meeting	GR
20060315	Changes after Technical Design Team meeting <ul style="list-style-type: none"> - Revision of history mechanism proposal - Transference of some enumerated DesignationType values to the new field Degree of Synonymy in the Language Refset member file. - Updated figures 	GR
20060601	Changes after April Technical Design Team meeting <ul style="list-style-type: none"> - History tables for RefSet and Members table adopt the new approach for generic history tracking using state-valid log tables - Removal of RefSetVersion table since it is not necessary with the new history tracking and component versioning system. 	

1 Introduction

1.1 Purpose

The purpose of this document is to inform wider discussion about the proposed Reference set mechanism, an evolution of the original SNOMED CT subset specification.

1.2 Who should read this guide?

The intended audience for this document is any individual or organization that wishes to develop or apply systems that will use SNOMED Clinical Terms. Comments are especially welcome from users/developers with experience of systems that have used Read Codes or SNOMED codes.

This document is mainly for:

- ❖ Clinical software developers - who have worked with Clinical Terms (the Read Codes) or versions of SNOMED and others who may be planning to build clinical systems.
- ❖ Clinicians – with an interest in health informatics.
- ❖ Health informatics specialists – who may be asked to advise on the implementation, and migration towards a new terminology, within new and existing systems.
- ❖ NHS planners, managers and information specialists – who will be involved in both local (and wider) implementations of Information for Health.
- ❖ US government or private sector organizations with a need for a controlled healthcare terminology.

1.3 Status

This document is regarded as an External Draft and is subject to revisions and extensions without notice.

1.4 Feedback

Further information about SNOMED CT is available on the Internet at www.snomed.org.

Please send your feedback to email address snomed@cap.org or contact:

SNOMED® International

College of American Pathologists

325 Waukegan Road, Northfield, IL 60093-2750

800-323-4040; 847-832-7700

2 Introducing Reference Sets

2.1 Overview

The SNOMED CT subset mechanism was provisionally adopted by the SNOMED International Editorial Board in late 2001. As originally planned, a revision of the mechanism has been done based on comments and implementation experience. The original requirements were refined to address the increasing usage of SNOMED in a range of diverse applications and the need to advance the internationalization of the core SNOMED CT design. The current Reference Set specification is an evolution of the original subset mechanism which enhances the ability to localize SNOMED CT to accommodate diverse user preferences and use cases.

The Reference Set proposal also includes recommendations for the evolution of other SNOMED CT elements, notably the Descriptions table, that are either necessary for the Reference Set proposal or provide improvements to SNOMED CT that would be made possible by its adoption.

In the rest of this document the abbreviated form “RefSet” may be used to make references to Reference Sets.

2.2 Basic principles

The effective usage of SNOMED CT requires a way to refer to sets of components that are appropriate for a specific use case. The subset mechanism may be used to filter and arrange the concepts for display, and also for selecting the descriptions most suitable for a particular scenario.

The Reference Set specification incorporates changes and additions to the subset mechanism. The enhancements are based on the following set of principles:

2.2.1 Improve the internationalization of the SNOMED CT design

The distinction between internationalization and localization is important. Internationalization is the adaptation of products for potential use globally, while localization is the addition of special features for use in a specific region or realm.

2.2.2 Increase the support for SNOMED CT localization

SNOMED CT localization efforts would be facilitated by the concentration of language sensitive components in a separate layer, and by providing a mechanism to adapt navigation and aggregation of SNOMED CT content to fit diverse user needs while preserving the integrity of the core.

2.2.3 Enhance change management support

SNOMED content is distributed as snapshots of the current component status at a particular release date. The original SNOMED CT history mechanism does not include change tracking support for subsets and their membership. This Reference Set specification incorporates enhancements to the change tracking mechanism that are aligned with the enhancements recommended to the overall SNOMED CT history mechanism. These overall enhancements support component-level versioning by enabling the release of valid states and effective times for components. The status of each RefSet member and its attributes can be tracked over time at the maximum level of granularity. The enhancements will enable incremental updates of SNOMED content since last synchronization, and facilitate time-sensitive queries for point in time retrieval of the status of each component.

The release content can be delivered using state-valid tables that will include all valid (released) versions of each component. Applications would then extract snapshots of that content to fit the purposes of their implementation.

2.2.4 Facilitate the use of Refset groups

Certain use cases require the combination of different kinds of RefSets to accomplish the desired effect. For example a set of concepts that are frequently used in a particular specialty would be more usable if they are arranged in an intuitive navigational hierarchy, and eventually privileging the specialty preferences over the default language RefSet.

2.3 Background

Experience with earlier versions of SNOMED and with NHS Clinical Terms has indicated that while comprehensive terminologies are valuable, their size and breadth of scope can pose a challenge for users and implementers. There are many situations in which it is useful to limit the set of Concepts and/or Descriptions that are readily available. The multinational, multilingual nature of SNOMED CT adds to the size and complexity of the terminology, increasing the need for limiting access to sets of components appropriate within a particular setting.

Experience since the first release of SNOMED CT suggests that the subset mechanism is an important element in making SNOMED CT implementable. However, there have been several cases where it has seemed that the mechanism could be enhanced and made easier to understand with a few relatively modest changes.

The current mechanisms have been shown to work and deliver significant benefits to SNOMED CT in terms of:

- ❖ Language (English, Spanish and German)
- ❖ Dialect (US and GB English)
- ❖ Realm Concept Subsets (Non-human, UK GP, Pathology)
- ❖ Navigational (Top-level and CTV3 hierarchies)

While the subset mechanism has gained broad acceptance by the user community, the original design had some limitations:

- ❖ Only active members of the subset were included in the Subset members table. Changes in membership composition were not registered in the Component history table.
- ❖ The approach brought together many different requirements and addressing these with a single file structure, tended to overload fields within that structure. The subset members table field MemberStatus was overloaded, having different meanings for each subset type.
- ❖ Experience with navigational subsets suggested more details were needed at the node level to support complex navigational use cases.
- ❖ Subset groups and the ability to define a subset based on another subset were only supported in the XML distribution format.

2.4 Proposed Reference set types

Seven types of RefSets are proposed in this document:

- ❖ Simple RefSet – selects a set of elements from SNOMED CT
- ❖ RefSet Group – groups RefSets that are designed to work together.
- ❖ Tagged RefSet – facilitates the marking of RefSet members
- ❖ Language RefSet – identifies elements and their role for a specific language
- ❖ Navigation RefSet – provides an alternate navigational hierarchy for the selected elements
- ❖ Aggregation RefSet – provides an alternate aggregation hierarchy for the selected elements

2.5 Relational representation of RefSets – summary

The RefSet mechanism has two tables to store the RefSets and the RefSet members. The structure of the RefSet table is the same for all RefSet types, and this specification includes five variants of the file structure of the RefSet Members table. The basic structure of the RefSetMember table support the distribution of RefSet groups, Simple and Tagged RefSets, while the other RefSet types require a specialization of the Members table structure to support additional attributes. Additional RefSet types might be defined in complementary specifications.

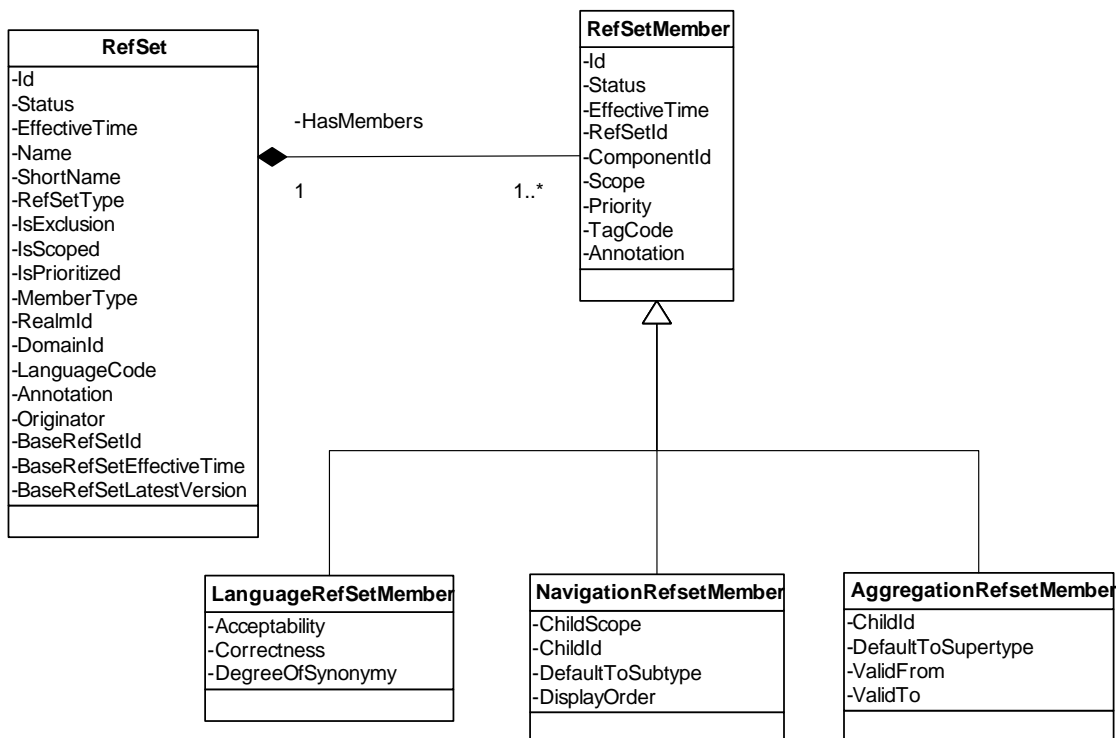


Figure 1 - The Reference Set Mechanism model

3 RefSets

3.1 Overview

The original Subset mechanism included a Subset table to store information about each subset version. A new Subset Id was issued for each new release (version) of the subset, and the field OriginalSubsetId stored the Id of the first version. The subset member table only included active members.

This versioning scheme has been replaced in the RefSet mechanism by log tables that enable the tracking of valid (released) component states over time.

The RefSet table includes a row for each valid state of a RefSet that has been released over time. The field EffectiveTime stores the Universal Time (UTC) at which the component representation has become effective (published). If the attributes of the RefSet row have not changed in a new release, the state remains valid and the value of the EffectiveTime field is maintained. This table is therefore a history table itself for the purpose of tracking RefSet attributes over time.

The RefSetMember table (and the related specializations described later) also includes a row for each valid state of a RefSet member and the corresponding EffectiveTime.

3.2 The RefSet table structure

The RefSet table includes information about each RefSet. The RefSetId is functionally equivalent to the OriginalSubsetId field in the Subset specification. Published subsets that would eventually be released in the new RefSet format will keep the OriginalSubsetId value as the RefSetId. The component identifier of a RefSet would be stable once it has been assigned.

Table Structure Key:

Italicized Field Name indicates a Primary Key

Valid Field Contents (for RefSet type):

[value]:	must contain this value
[value][value]:	must contain one of these values
M:	mandatory – field must exist and value may not be null
R:	required – field must exist but may be value may be null
C:	conditional – if field exists for any row, it must exist for all rows

Boolean values:

0:	False
1:	True
0 1:	True or False

Tables are tab-delimited

Tables have field name headings

RefSets Table:

Field Name	Type	Notes	
<i>Id</i>	SCTID	Identifies the Reference Set, using the same partitionId allocated originally to SubsetIds (3).	M
<i>EffectiveTime</i>	String[14]	Universal Time (UTC) at which the component representation becomes effective. The component representation remains valid until it is superseded. The date and time should be represented using ISO 8601 conventions with no punctuation and no letter T between date and time, with a granularity up to the level of seconds.	M
Status	Enum	Status of the RefSet record	M
Name	String[255]	Name of the RefSet	M
ShortName	String[20]	Short name for the RefSet	R
RefSetType	Enum	1= Simple; 3=Navigation; 4=Tagged; 5=Language; 6=RefSet Group 8=Aggregation	M
IsExclusion	Boolean	0=RefSet members are included 1=RefSet members are excluded	0
IsScoped	Boolean	0=Not Scoped 1=Scoped (members must have the Scope property)	0 1
IsPrioritized	Boolean	0=RefSet member rows are not prioritized 1=RefSet member rows are prioritized	0 1
MemberType	Enum	0=Concept, 1=Description, 2=Relationship, 3=RefSet (only applicable to RefSet groups), 6=CrossMap 7=RefSetMembers (only applicable to Tagged RefSets)	1
RealmId	SCTID	Country or other jurisdiction for which this RefSet applies (null=applies for all). It is a SNOMED CT concept.	R
DomainId	SCTID	Discipline, specialty or business area for which this RefSet applies (null=applies for all). It is a SNOMED CT concept.	R
LanguageCode	String[8]	Language for which this RefSet applies	M
Annotation	String[255]	Text to associate with the RefSet	R
Originator	SCTID	Namespace concept for the creator of the RefSet	R
BaseRefSetId	SCTID	Foundation RefsetId on which this Refset Version is based	R
BaseRefSetIdEffectiveTime		Effective time of the foundation Refset state on which the Refset is based.	R
BaseRefSetLatestVersion	Boolean	0=Use the specified BaseRefSetIdEffectiveTime of the Base RefSet members 1=Use the latest version of the base RefSet members if a more recent version is available	0 1

Notes:

RefSet.Id – each RefSet is given a unique RefSetId. For RefSets that include content previously released in Subset format, the SubsetOriginalId (the SubsetId of the first release of the Subset) is used as RefSetId.

RealmId, DomainId – it is proposed that SNOMED CT concepts will be used as the values for these fields.

LanguageCode – consists of a code and optional sub-code, separated by a dash. Code=ISO639-1 language code; sub-code=ISO3166 country code (or IANA code for countries not listed in ISO3166). If the LanguageCode value=0, the RefSet applies to all languages – however, this is not a valid value for Language RefSets.

BaseRefSetId – identifies a foundation RefSet on which the RefSet is based. The original Subset Definition File in XML format allowed the specification of such a base subset to facilitate the definition of new subsets by only describing the differences or exceptions. This feature is now included in the relational RefSet distribution.

BaseRefSetEffectiveTime – Effective time of the foundation RefSet state on which the RefSet is based

Scope - Some exclusions, inclusions and prioritizations may be fixed for a given installation, configurable by user login or user selection. Others may apply to specific data entry fields, templates or protocols. In these cases, the context in which each set of included, excluded or prioritized components needs to be represented. Implementations need to support dynamic real time context-sensitive switches in these constraints. In the current subset mechanism, context subsets are intended to fulfill this role. In the proposed RefSet mechanism, this role is fulfilled by scoped RefSets.

IsPrioritized – Prioritized RefSets establishes the order of precedence according to a particular use case.

3.3 The RefSet members table structure

The members table includes a row for each valid state of a member of a RefSet. In addition to creating new functionality, the design goal was to remove the ambiguity created by having multiple uses for the MemberStatus and LinkedId fields in the Subset Members table. The structure of RefSet members table is specified for each type of RefSet in the forthcoming sections. Several RefSet types use the basic RefSetMember table, while other require specialization of that structure to accommodate additional fields.

4 Simple Reference Sets

4.1 Overview

The requirements of the following subset types, described in the original subset specification could be addressed with the simple Reference Set table structure:

- Realm concept subset (the concepts applicable for a particular Realm)
 - A realm concept subset refers specifically to concepts, but it also affects the availability of descriptions and relationships. Only descriptions and relationships associated with the included concepts are allowed. Descriptions and relationships not associated with concepts in the subset are not referenced.
- Realm relationship subset
- Context concept subset (now a Scoped Simple RefSet)

Please note that functionality provided by realm description subset and context descriptions subsets is now intended to be provided through Language Refsets.

When a Simple RefSet refers specifically to concepts, it also affects the availability of descriptions and relationships. Navigational RefSets (described later) could also serve the purpose of limiting availability of concepts and descriptions while providing alternate navigational links between concepts.

4.2 Relational representation of Simple Reference set members

Each valid status of a Simple Reference Set member is represented by a row in the RefSetMember table.

Field Name	Type	Notes	
<i>Id</i>	SCTID	Unique SCT identifier for the RefSet member row. It uses the partitionId 7, that identifies RefSet members.	M
<i>EffectiveTime</i>	String[14]	Universal Time (UTC) at which the component representation becomes effective. The component representation remains valid until it is superseded. The date and time should be represented using ISO 8601 conventions with no punctuation and no letter T between date and time, with a granularity up to the level of seconds.	M
Status	Enum	0=Current, 1=Non Current	M
RefSetId	SCTID	Refset to which the component reference belongs	M
ComponentId	SCTID	Unique SCT identifier of the referenced component	M
Scope	String[20]	Context in which this member row applies	C
Priority	Integer	Relative priority of the member in the set	M
TagCode	String[20]	Coded attribute to associate with this member	R
Annotation	String[255]	Text to associate with this version of the RefSet	R

Notes

If the Status = 1, then the value of the fields Scope and Priority should be ignored. The row will be kept in the distribution as a non current member of the RefSet. RefSet membership status should not be confused with the status (Concept.Status, Description.Status) of the referenced component.

Once published, the referenced ComponentId can not be changed. Changes in other fields like Order, Scope, Priority, TagCode and Annotation would result in the insertion of a new row reflecting the state of the reference and the time at which it should be considered effective. The Id will be preserved within the new row.

The RefSetMembers table has a composite primary key that includes the Id and the EffectiveTime field.

4.3 History mechanism

The current and previous statuses of the references are included in the RefSetMember table. Therefore it serves as a history table supporting several use cases:

- Retrieve a snapshot with the current status of the member references.
- Retrieve the member rows that have been updated since a particular release or since the last time an application has performed an incremental update.
- Retrieve the value of an attribute at a specific point in time.
- Retrieve a collection of members active at a specific point in time.

The proposed distribution table structure is not necessarily the most efficient implementation format in all scenarios. Applications that only require current data might choose to take a snapshot of the current content only, while implementations that make intensive use of temporal queries might consider the addition of a SupersededTime column. This data can be calculated using the distributed EffectiveTime to facilitate more efficient queries.

Note that the content of EffectiveTime may indicate a future date. This possibility is discussed in detail in the specification that revises the History mechanism.

RefSet groups

4.4 Overview

Experience gathered during the development of subsets for specific specialty domains demonstrated that a suite of coordinated subsets may be needed to provide the desired functionality.

- ❖ For example, a specialty Simple Refset would facilitate the filtering of the content to concepts relevant to the domain. A Language RefSet could be used to store the preferences for the display of descriptions and to provide intuitive navigation of the content, a companion Navigational RefSet may be convenient.,
- ❖ The requirements for efficient browsing of the content of a Simple RefSet may include the ability to change the “view” depending on the user role or task. This could be accomplished with a group of Navigational and Language RefSets working together.

4.5 Relational representation of RefSet groups

RefSet groups could be represented using the Simple RefSet members table structure. In this case, the referenced component is RefSet identifiers.

4.6 History mechanism

Since this type of RefSet is actually a Simple RefSet with specialized membership, the Simple RefSet history mechanism is pertinent for tracking changes in RefSet group composition.

5 Tagged RefSets

5.1 Overview

- ❖ In some scenarios, it is useful to flag members of a RefSet with user-defined tag codes. This can be accomplished using the Simple RefSet Member table structure. The referenced members are then Ids

5.2 Relational representation of Tagged RefSets

Tagged RefSets could be represented using the Simple RefSet table structure, but the member types could also include RefSet members in addition to other SNOMED CT component types; a non null value is mandatory for the field TagCode.

Note that the TagCode field is a free text field. Alternatives for the provision of a more structured definition of allowable codes for this field were examined. Since Tagged RefSets could be useful in very diverse scenarios, additional constraints were avoided. Content of this field is thus implementation specific. This decision may need revision according to user feedback.

A potential use of this RefSet type is the definition of value, using the TagCode field to store the value set identifier. This usage would require further constraints and/or explicitly enumerated value set identifiers to facilitate data set exchange.

5.3 History mechanism

Since this type of RefSet is actually a Simple RefSet with specialized membership, the Simple RefSet history mechanism is pertinent for tracking changes in the membership of Tagged RefSets.

6 Language Reference Sets

6.1 Overview

SNOMED CT is designed as a multilingual terminology. The Descriptions Table will contain Descriptions for every Concept in all supported Languages and Dialects.

A user or group of users normally only require Descriptions in their own Language. The preferences for particular descriptions within the Language would depend on many factors e.g. dialects, regional and cultural aspects, user specialty, local variants, and intended use.

The RefSet mechanism extends the SNOMED CT potential for localization by clearly separating language (descriptions) from language usage (local preferences for particular descriptions). To achieve this, preferences would be stored in the Language RefSet. To resolve the potential overlap with features previously provided by the Descriptions table, slight modifications of the table are proposed for the distribution format and the enumerated values in certain fields. Those modifications would be backward compatible with the original distribution format.

6.2 Requirements

6.2.1 Requirement summary

- ❖ Some Descriptions may be preferred in one dialect or setting and but not acceptable in another dialect of the same language.
- ❖ A Language RefSet should refer to each Description that is used in a given language, dialect or local variant and state the appropriate preferences for language usage in that setting.
- ❖ User preferences may favor the usage of terms that are discouraged or considered incorrect by a Language Authority. Terms borrowed from other languages for example anglicisms, galicisms and Latinisms are used in everyday practice. These User preferences (or actual language usage) should be distinguished from the language Authority recommendations.
- ❖ While synonymy is usually relative, it is expected that terms that are true synonyms could be exchanged without affecting the meaning of the phrase. In restricted implementation contexts, there is a need to support terms that are beyond the specific context of use, would be ambiguous or more general.
- ❖ In certain implementation scenarios that involve Navigational RefSets, there is a need to support context specific descriptions that otherwise could not be considered synonyms of the associated concept. Those descriptions should not be used in search indexes, and should only be displayed when referenced in conjunction with a navigational RefSet within an explicit scope.
- ❖ A Language RefSet should keep references to Descriptions that have been previously published but become non active members of the RefSet.
- ❖ The same mechanism used for the distribution of Language Editions should be available to implementations that wish to localize the linguistic preferences at the regional, institutional or even user level.

6.2.2 Current approach

- ❖ The Descriptions table stores terms with three description types: Fully Specified Names (FSN), Preferred Terms, and Synonyms. The LanguageCode field stores Language and Dialect information.

- ❖ Language subsets define what descriptions are included in a particular Language or dialect distribution and the description type specific to the subset, which might eventually override the “Preferred term” defined in the Descriptions table.
- ❖ Changes in the Descriptions table are recorded in the Component History table. But changes in the membership status of a subset are not tracked since the subset identification does not include unique Ids for member rows or membership status.

6.3 Language Reference set member table structure – summary

Field Name	Type	Notes	
<i>Id</i>	SCTID	Unique SCT identifier for the RefSet member row. It uses the partitionId 7, that identifies RefSet members.	M
<i>EffectiveTime</i>	String[14]	Universal Time (UTC) at which the component representation becomes effective. The component representation remains valid until it is superseded. The date and time should be represented using ISO 8601 conventions with no punctuation and no letter T between date and time, with a granularity up to the level of seconds.	M
Status	Enum	0=Current, 1=Non Current	M
RefSetId	SCTID	Refset to which the component reference belongs	M
ComponentId	SCTID	Unique SCT identifier of the referenced component	M
Scope	String[20]	Context in which this member row applies	C
Priority	Integer	Relative priority of the member in the set	M
TagCode	String[20]	Coded attribute to associate with this member	R
Annotation	String[255]	Text to associate with this version of the RefSet	R
Acceptability	Enum	Acceptability of the description to users: 0=Not specified, 1=Preferred, 2=Acceptable, 3=Not recommended, 4=Not acceptable, 5=Invalid	M
Correctness	Enum	Correctness of the member in the language: 0=Not specified, 1=Recommended, 2=Acceptable, 3=Not recommended, 4=Incorrect, 5=Invalid	R
DegreeOfSynonymy	Enum	Degree to which a term covers the same concept covered by the original, primary Fully Specified Name: 0=Not specified, 2=Synonymous, 3=Near synonymous (depending of context of use), 4=Non synonymous, 5=Invalid	R

Id – a SCTID that uniquely identifies the component as a member of the RefSet; the Id is retained across versions of the RefSet. This is not the component’s own unique SCTID within SNOMED CT, which is indicated in the ComponentId field.

Status – once a component has been released as a member of a Language RefSet, it persists as a member of every subsequent version to enable interoperability among versions. If the reference is retired (the description is no longer considered as an active reference) then the RefSetMemberStatus would record the new status of the reference. The status of the referenced component is tracked in the corresponding component table. Descriptions with LanguageAcceptability of “Invalid” will always be “Not Current”.

Scope – this property defines the context in which this member-row is valid as part of a scoped RefSet. Within a scoped RefSet, a component may appear as a member more than once. Each instance will have a unique Id and must have a unique Scope value.

ComponentId – the SCTID that uniquely identifies the component within SNOMED CT.

Acceptability – indicates the user preferences (within the dialect/realm) for descriptions of a concept.

Allowed values for Acceptability are:

- 0 = **Not specified** – the acceptability of the referenced term is not specified in the RefSet or is not considered relevant.
- 1 = **Preferred** – indicates the most common or accepted designation for a concept (within the realm). There may be only one Preferred designation of each DescriptionType (see *Changes to the Descriptions Table*, below)
- 2 = **Acceptable** – accepted by most users as a valid representation of the concept. Equivalent to a synonym in the original descriptions table.
- 3 = **Not recommended** – understood as a valid representation of the concept, but not frequently used and possibly carrying undesired connotations
- 4 = **Not acceptable** – not a valid designation in this realm or dialect, although it may be valid in other, related realms that share the same Language.
- 5 = **Invalid** – not a valid designation for the concept; it may have previously been published as a valid designation. Descriptions that are retired at the Language level (Descriptions table) would become invalid in all Language RefSets referencing that description.

Please note that traditionally only positive preferences have been considered (“Preferred terms”), and those preferences were usually perceived as an attribute of the description. This proposal considers that preferences depend on the usage of the language rather than on the language itself. In addition, negative preferences are also recorded in this field. For example, in a certain region, a term that is acknowledged as a valid designation for the concept in the language may be considered not acceptable at the local level.

Correctness – indicates the status of the description as determined by the official Naming Authority for the language/realm, which may or may not correspond with user preferences (Acceptability). If no Authority exists for the language/realm, Correctness=5. Allowed values for Correctness are:

- 0 = **Not specified** – the correctness of the referenced term is not specified in the RefSet or is not considered relevant.
- 1 = **Recommended** – indicates the preferred designation for a concept (according to the Authority).
- 2 = **Acceptable** – accepted by the Authority as a valid designation of the concept (agreed upon as an alternative)
- 3 = **Not recommended** – Discouraged by the Naming Authority
- 4 = **Incorrect** – the linguistic expression is not accepted by the Authority, however it may be acceptable to, and understood by, users

5 = **Invalid** – not a valid designation for the concept; it may have previously been published as a valid designation. Descriptions that are retired at the Language level (Descriptions table) would become invalid in all Language RefSets referencing that description.

Degree of Synonymy – Degree to which a term covers the same concept as the primary, original Fully Specified Name. There is a tension between the need to accommodate expressions common in natural language and a more restrictive policy to only include actual synonyms. A central role in this dilemma is played by the representation of context. In natural language, the context in which an expression is used might modify the intended meaning.

Allowed values for Degree of Synonymy are:

0 = **Not specified** – the Degree of Synonymy of the referenced term is not specified in the RefSet.

2 = **Synonym** – a term that represents the same concept as the original Fully Specified Name in the American English Edition.

Two terms are considered synonyms if they represent a single concept, within a specific semantic category as the only additional context. Ideally, synonyms should unequivocally represent a concept regardless of the context; however, it is not uncommon to find a term representing two different concepts. For the purposes of this specification, two terms will be considered synonyms provided they represent the same concept in a specific semantic category (e.g. <drinking> in <devices>, <procedures>, <findings>, etc.).

Two terms are considered synonyms provided the concept represented by one of them in a statement does not vary if the terms are interchanged (i.e. the first term is replaced by the second one). This interchangeability property should be assessed in most use cases.

Two terms are considered synonyms provided they share the same level of granularity for a concept. In the case where one term should be more specific than the other, they might be representing different concepts.

Two terms are considered synonyms provided they represent the same concept in all or most contexts.

3 = **Near synonym** – a term that represents the same or a similar concept as the original Fully Specified Name in the American English Edition, depending of context of use. The interchangeability is limited. Implementers should decide whether to consider the referenced term as a valid concept designation in the context on which the users would interact with the application. It is included as a relative synonym since users frequently use it to reference the concept.

4 = **Non synonymous** – a term that cannot be considered a synonym of the concept designated by the Fully Specified Name in the American English Edition, but which is included to support implementation specific scenarios. The Scope attribute of the corresponding member row should be other than the default scope, since this term should only be displayed when referenced by a scoped navigation RefSet. The term should not be included in retrieval scenarios that are not restricted to the scope specified. The referenced descriptions are included as a particular case of near synonyms when the phrase is commonly used to reference the associated concept. While the release of this kind of description is discouraged, the enumerated value is provided for its potential use in extensions that require the inclusion of colloquial terms, jargon, and clipped forms.

5 = **Invalid** – not a valid designation for the concept; it may have previously been published as a valid designation. Descriptions that are retired at the Language level (Descriptions table) would become invalid in all Language RefSets referencing that description.

6.4 Proposed changes to the Descriptions table

The inclusion of the Acceptability, Correctness and DegreeOfSynonymy attributes in the RefSetMembers table removes dialect-dependent information from the Descriptions table, improving support for international editions. This proposal also includes changes to the enumerated values allowed in three fields, to avoid overloading in their meaning.

Fields	Type	Notes
<i>DescriptionId</i>	SCTID	The unique SNOMED CT Identifier for this description.
DescriptionStatus	Enum	The status indicates whether the description is in current use for any language. 0=Current, 1=Not Current, 10=Moved Elsewhere, 11=Pending Move
ConceptId	SCTID	Unique SNOMED CT Identifier for the clinical concept to which this description applies. This field provides an association between the Designations and Concepts tables.
Term	String[255]	The text of a designation used to represent the associated concept.
InitialCapitalStatus	Boolean	Indicates whether the case of the first character of the term is significant. 0=False (Case of the first character may be changed without changing the meaning), 1=True (Case of the first character must not be changed)
DescriptionType	Enum	Indicates whether the term is a 3=FullySpecifiedName, 4=Designation, 5=Explanation or Definition, 6=External reference
LanguageCode	Enum	Language code (the dialect is specified at the Language RefSet member level, since a description might be used in multiple dialects with diverse preferences)
DescriptionForm	Enum	Bit field indicating special designation types; see notes. 0=Unabbreviated singular form, 1=Plural form, 2= Incomplete or Abbreviated form, 4=Symbolic form

Notes:

DescriptionStatus – this property indicates whether the description is active in the language. The status of a description with respect to any specific dialect is indicated within the Language RefSet in the Acceptability field.

A description will remain current if it is an active description of a concept, even if the concept itself is inactive (or limited); statuses 6 and 8 are no longer valid for descriptions.

DescriptionType – indicates whether the term is a Designation, a FullySpecifiedName, a Definition or an external reference. The acceptability field in the Language RefSet would specify the preferred designation for each of these DescriptionTypes. For example, “Preferred terms” in the original SNOMED CT distribution format will have a DescriptionType=4 (designation), and Acceptability=1 (preferred) under this specification.

- 3 = **Fully Specified Name** – a term that provides an unambiguous way to describe a concept. Its purpose is to uniquely designate a concept and clarify meaning rather than present a commonly used or natural phrase. The original Fully Specified Name used when the concept was added to SNOMED CT is considered the Primary FSN, and the gold standard representation of the intended meaning of the concept. The Primary FSN should follow American English spelling guidelines and conventions. A copy of the Primary FSN is stored in the FullySpecifiedName field of the Concepts table, and distributed with all Language Editions. Secondary Fully Specified Names are the result of the representation/translation of the Primary FSN in a given language. All secondary FSNs are stored in the Descriptions table.
- 4 = **Designation** – a term that is used in a Language to represent the same concept designated by the primary, original Fully Specified Name of SNOMED CT.
- 5 = **Explanation (definition)** – a statement that describes and clarifies the meaning of a concept.
- 6 = **External reference** – An expression referencing an external resource associated with the concept. It could be an URL, URI, bibliographic citation, etc.

DescriptionForm – indicates if the designation falls into one or more special types.

- 0 = **Unabbreviated singular form**
- 1 = **Plural form**
- 2 = **Incomplete or Abbreviated form**– abbreviation or acronym (“designation formed by omitting words or letters from a longer form and designating the same concept” ISO-1087-1:2000)
- 4 = **Symbolic form** – A formal expression that is a valid designation of the concept.

LanguageCode - In the original structure, this field stored Language and dialect codes (e.g. en-US). It is proposed that in the future this field should be limited to store Language (en), since the inclusion and status of a description in a dialect would be handled at the Language RefSet level.

6.5 History mechanism

The history of valid states of the Language RefSet Members is stored in the LanguageRefSetMember table itself, as described earlier.

Examples of time-sensitive queries using this history table:

What was the description with acceptability = Preferred for concept X at time Y for RefSet Z?

```
Select descriptionId, term from Descriptions D
  inner join LanguageRefsetMemberHistory LRS on LRS.componentid=D.descriptionid
  where d.conceptId=X
  and LRS.RefSetId=Z
  and D.DescriptionType=4
  and LRS.Acceptability=1
  and LRS.EffectiveTime=
  (Select max(EffectiveTime) from LanguageRefsetMemberHistory LRS2
```

```

where LRS2.componentid=D.descriptionid
      and LRS2.RefSetId=Z
      and LRS2.Acceptability=1
      and LRS2.EffectiveTime<=Y)

```

How can a SupersededTime be calculated, in order to simplify queries over state-valid tables?

```

select * ,
      isnull((select min(EffectiveTime) -1 from LanguageRefsetMemberHistory LRS2
              where LRS2.ComponentID = LRS.ComponentID
              and LRS2.EffectiveTime > LRS.EffectiveTime) , '2050-01-01')
      as supersededTime from LanguageRefsetMemberHistory LRS

```

Then the previous query could be re-stated as:

```

select descriptionId, term from Descriptions D
      inner      join      LanguageRefsetMemberHistory      LRS      on
LRS.componentId=D.descriptionId
      where D.conceptid=X
              and LRS.RefSetId=Z
      and D.DescriptionType=4
      and LRS.Acceptability=1
              and LRS.EffectiveTime<=Y
              and (LRS.supersededTime>Y or LRS.supersededTime is null)

```

7 Navigation Reference Sets

7.1 Overview

There is a requirement to be able to organize concepts into hierarchies suitable for navigation. This requirement arises because the subtype hierarchy is often not intuitive for navigation due to its depth, size and lack of a definable order. The introduction of support for negation and context will make some subtype hierarchies more accurate but even less intuitive to a human reader.

7.2 Requirements

An enhanced version of this requirement includes the ability to determine the navigational children of a concept, based on the manner in which that concept was reached. For example, if diabetes mellitus is found by navigation down a hierarchy of disorders then one set of refinements might be presented as its navigational children. However, if the same concept is reached as a cause of cataract then a much more limited selection of options might be available.

7.3 Navigation Reference set member table structure – summary

Field Name	Type	Notes	
<i>Id</i>	SCTID	Unique SCT identifier for the RefSet member row. It uses the partitionId 7, that identifies RefSet members.	M
<i>EffectiveTime</i>	String[14]	Universal Time (UTC) at which the component representation becomes effective. The component representation remains valid until it is superseded. The date and time should be represented using ISO 8601 conventions with no punctuation and no letter T between date and time, with a granularity up to the level of seconds.	M
Status	Enum	0=Current, 1=Non Current	M
RefSetId	SCTID	Refset to which the component reference belongs	M
ComponentId	SCTID	Unique SCT identifier of the referenced component	M
Scope	String[20]	Context in which this member row applies	C
Priority	Integer	Relative priority of the member in the set	M
TagCode	String[20]	Coded attribute to associate with this member	R
Annotation	String[255]	Text to associate with this version of the RefSet	R
ChildScope	String[20]	Target scope to be considered to select the Child	C
ChildId	SCTID	SCTID of the referenced child component	
DefaultToSubtype	Boolean	Indicates whether drill-down navigation should continue using the subtype hierarchy if the concept referenced by the ChildId has no explicit navigational children in the subset.	C
DisplayOrder	Integer	Indicates the default order for member rows	R

Notes

The root members of the navigational hierarchy can be identified by comparing the fields ComponentId and ChildId for the default scope. If these fields match, then it is a root member.

7.4 Navigation Reference Set example

- Works by direct reference to a component.
- Scope added to modulate the children referenced by a parent according to situation.
- ChildScope added to modulate the components referenced by after reaching a Child.
- This approach mean simple cases (i.e. with same value for Scope and ChildScope) work exactly like a simple navigations Refset (or current subset).

C1: Diagnoses

C2: Asthma

C5: Investigation

C7: Peak flow

C8: Chest X-ray

C6: Treatment

C9: Salbutamol

C10: Beclomethasone

C3: Pneumonia

C5: Investigation

C8: Chest X-ray

C11: Sputum Culture

C6: Treatment

C12: Amoxil

C13: Clindamicin

C4: Appendicitis

C6: Treatment

C14: Appendectomy

C15: Antibiotics

C16 Amoxyclav

C12: Amoxil

C17: Metronidazole

Note the parents and children of concepts change with scope **A**, **B**, **C**, **D** so that C6: Investigation has different children in each of the three scopes and C8: Chest X-ray and C12: Amoxil have different parents according to scope.

See below the table that delivers the scope based navigation hierarchy on previous page.

Scoped Navigation example (partial view, simplified)

ComponentId	Scope	ChildScope	ChildId	DefaultToSubtype	DisplayOrder
C1	A	A	C2	0	1
C1	A	A	C3	0	2
C1	A	A	C4	0	3
C2	A	B	C5	0	1
C2	A	B	C6	0	2
C3	A	C	C5	0	1
C3	A	C	C6	0	2
C4	A	D	C6	0	1
C5	B	B	C7	0	1
C5	B	B	C8	0	2
C5	B	B	C8	0	2
C6	B	B	C9	0	1
C6	B	B	C10	0	2
C5	C	C	C8	0	1
C5	C	C	C11	0	2
C6	C	C	C12	0	1
C6	C	C	C13	0	2
C6	D	D	C14	0	1
C6	D	D	C15	0	2
C15	D	D	C16	0	1
C15	D	D	C12	0	2
C15	D	D	C17	0	3

8 Aggregation Reference Sets

8.1 Overview

Aggregation Reference Sets would allow the distribution of alternative aggregation hierarchies for SNOMED Concepts. The preliminary table structure is included in figure 1 to stimulate discussion at this stage of the Draft document while ongoing refinement work is underway.

9 Prioritized Reference Sets

9.1 Overview

Prioritized Reference Sets would allow the prioritization of certain components over others according to the intended usage. The preliminary table structure is included to stimulate discussion at this stage of the Draft document while ongoing refinement work is underway.

- Prioritization of concepts that are commonly used, so they appear at the top of searches or pick lists.
 - For example, a user may prefer to see the ten most common procedures they perform at the top of a pick list.
- Prioritization of descriptions which, while possibly not marked as *Preferred Terms* are preferred by a particular user.
 - For example, a user may prefer to see "diabetes mellitus" as the first match in a search for "diabetes".