Chapter 9
SNOMED CT

Abstract SNOMED CT is a comprehensive multilingual clinical terminology used in electronic health records and interoperability. Its components are concepts (codes), descriptions (terms) and relationships. Concepts are organized into hierarchies. SNOMED CT expressions are either a single concept (pre-coordinated) or more complex post-coordinated expressions using compositional grammar. Reference sets are used to provide subsets of the whole terminology for specific purposes.

Keywords SNOMED CT • Components • Concept • Description • Relationship • SctId • Defining relationship • Qualifying relationship • Hierarchy • Description logic • Expression • Compositional grammar • Pre-coordination • Post-coordination • Subtype qualification • Axis modification • Subsumption testing • Transitive closure table • Reference set • Language reference set • Navigation reference set • Map reference set • Release format 2 (RF2) • Delta release • SNOMED CT documentation

Introduction

SNOMED CT is the most comprehensive multilingual clinical healthcare terminology available. It is used in electronic health record systems to facilitate clinical documentation and reporting and to retrieve and analyse clinical data.

SNOMED CT is both a coding scheme, identifying concepts and terms, and a multi-dimensional classification, enabling concepts to be related to each other, grouped and analysed according to different criteria. IHTSDO (see Chap. 6) describes SNOMED CT as the global language of healthcare. While SNOMED CT has some of the attributes of a language, it is probably more useful to think of it as a coding system, where the codes used are unambiguous and are designed for computer processing.

Today most uses of SNOMED CT use single codes, although SNOMED CT provides a compositional grammar allowing complex expressions to be built up. Such expressions create challenges for reporting and analysis, but in future such post-coordinated expressions may become widely used.

T. Benson, G. Grieve, *Principles of Health Interoperability*,
Health Information Technology Standards, DOI 10.1007/978-3-319-30370-3_9

SNOMED CT has two key features that make it a major improvement over most other coding systems used in healthcare. First it is virtually future-proof, by being inherently evolvable. Concepts, terms and their codes can be added or deprecated (once added codes are not deleted) without limit and their relationships with others can be changed. Another aspect of its flexibility is support for multiple languages and dialects. Furthermore, SNOMED CT supports multiple relationships, including multiple parent-child hierarchical relationships. This simply reflects practical reality, but, as described in Chap. 8, many current coding schemes can only support a single axis of classification, which can be severely limiting.

At its heart, SNOMED CT is a model of meaning (see Chap. 11). The focus is on what is technically true and correct. For example in SNOMED CT, urine and glucose are *substances*, urine glucose concentration is an *observable entity*, a urine glucose test is a *procedure* and a urine glucose test result is a *clinical finding* and if a urine glucose test is not done it is a *situation with explicit context*. This can be quite confusing (we explain this in Chap. 10).

On the other hand, users are only interested in what is useful to them, and this is known as the model of use (see Chap. 11). However the model of use varies according to context. GPs, urologists and nurses are interested in different things about urine. In contrast, the model of meaning is always correct. Good implementations use a model of use for the user interface, supported by the universal model of meaning in the background, to provide the best of both.

On its own SNOMED CT does very little, but its value is realized when it is built into software, such as an EHR. The best implementations will be designed specifically for use with SNOMED CT.

Kent Spackman, the leader of the team that developed SNOMED CT postulated two golden rules: [1]

> *The first rule of coding is that yesterday's data should be usable tomorrow.*

Clinical data needs to be treated as being permanent. We have to be able to use yesterday's and today's data for the indefinite future. On the other hand, if nobody is going to re-use the data, there is no need to code it.

> *The first rule of data quality is that the quality of data collected is directly proportional to the care with which options are presented to the user.*

There is an enormous variety of medical activity and any attempt to impose a *one size fits all* approach is doomed to failure. This is why the model of use, implemented using context-specific reference sets is a key aspect of SNOMED CT implementations. Good implementation invariably takes full account of context.

SNOMED CT provides tools to record information about patients in a way that can be indexed and retrieved for reuse clinically at the point of care and subsequently for management, surveillance and research. It provides the clinical content and expressivity required for precise clinical documentation.

SNOMED CT has a broad coverage of routine clinical medicine, but still needs extension in areas such as social care and some specialist areas.

Origins of SNOMED CT

In 1999 the English National Health Service (NHS) and the College of American Pathologists (CAP) agreed to merge SNOMED RT (Reference Terminology) with the Read Codes Version 3, also known as the NHS Clinical Terms Version 3 (ctv3), to produce a single joint clinical terminology – SNOMED CT (Clinical Terminology). The merger was completed in 2002 with the first release of SNOMED CT.

SNOMED CT was a true merger. Every Read Code and previous SNOMED RT code ever released was incorporated into SNOMED CT so that migration to it would not result in loss of information. In 2007, the International Health Terminology Standards Development Organization (IHTSDO) acquired all rights to SNOMED CT (see Chap. 6).

Cimino's Desiderata (see Chap. 7) were key design criteria. Additional criteria were that all content should be understandable, reproducible and useful. Understandable means that definitions should be understood by average clinicians, given brief explanations. Reproducible means that retrieval and representation of the same item should not vary according to the nature of the interface, user preferences or the time of entry. Usable leads to the conclusion that we should ignore distinctions for which there is no use in healthcare.

SNOMED CT is large. The number of concepts, descriptions and relationships varies with every release. SNOMED CT contains more than 300,000 active concepts, about one million English descriptions and more than 1.4 million relationships. There is no paper version. It can only be accessed using specialised software, such as a SNOMED CT browser. Its sheer size is a significant issue in development, use and maintenance.

SNOMED CT cannot be used manually, partly because it is too big, but more importantly because it works in a way that is completely different from earlier coding schemes such as ICD or the Read Codes. The reference structure of SNOMED CT relies on explicitly defined relationships that need computer software to work. This is considerably more complex than code-dependent hierarchies, but is more powerful, flexible and future-proof. It allows any concept to be defined or qualified in as many ways as are needed.

SNOMED CT provides an extensible foundation for expressing clinical data in local systems, interoperability and data warehouses. The terminology is composed of *concepts*, *descriptions* and *relationships* that provide a way to represent clinical information across the broad scope of healthcare and can support analysis and clinical decision support.

SNOMED CT is organised into *hierarchies*. A node in a hierarchy represents each concept, with one or more subtype relationships to its parent(s). Understanding these hierarchies is important and their content is described in Chap. 10 SNOMED CT Concept Model. This chapter describes the structure and components of SNOMED CT.

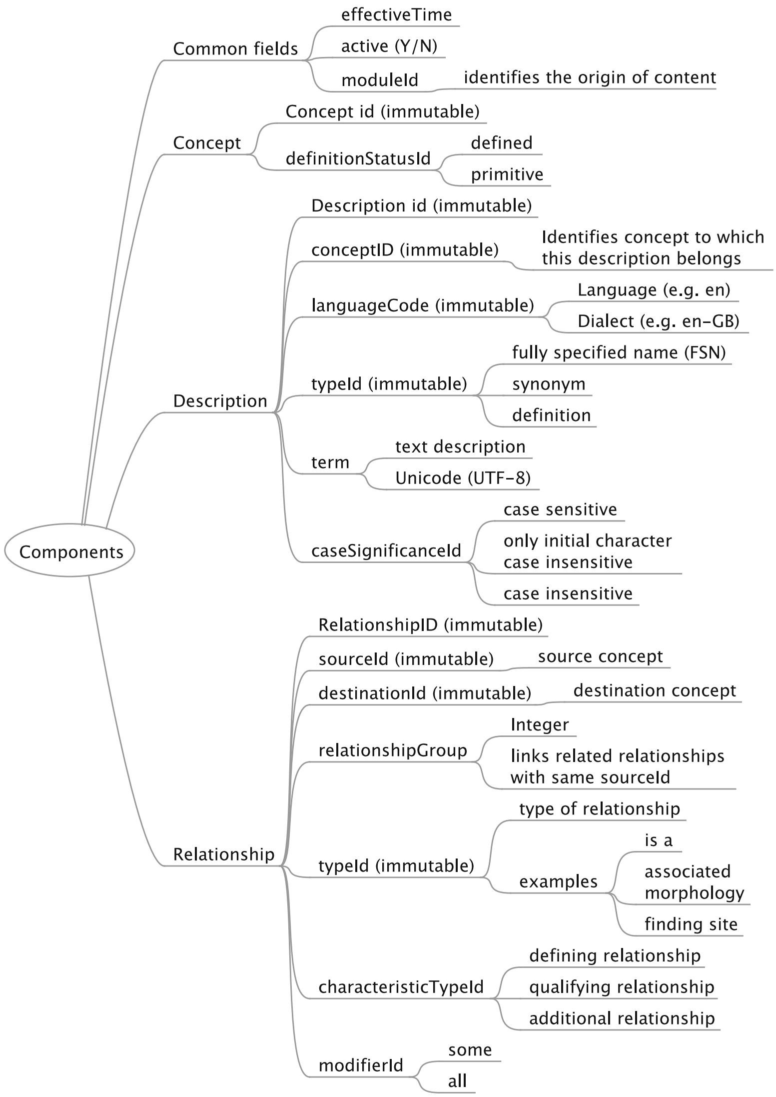

Fig. 9.1 SNOMED CT components

Components

SNOMED CT is composed of components, such as concepts, relationships, descriptions, reference sets and cross maps. A SNOMED CT Identifier (*sctId*) identifies every component. Figure 9.1 shows the main SNOMED CT components and their fields in RF2 release format.

All components have an *effectiveTime* field which states the time when this version of a component superseded the previous version or was created initially. Use of *effectiveTime* allows a single file/table to hold all released versions of SNOMED CT, computation of *snapshot* view for any specified date and *delta* release files of new rows supporting incremental updates.

All components carry an *active* field. An important principle of SNOMED CT is that of permanence. Once a component, such as a concept or description has been created it is never deleted, but the active flag may be set to inactive status. They also have a *moduleId*, which is used to identify the origin and organization responsible for maintaining this component.

Concept

SNOMED CT is concept-oriented. Concepts are clinical meanings that do not change. Each concept has a unique concept identifier (*conceptId*), which is an *sctId*. The conceptId is important because it is the code used to represent the meaning in clinical records, documents, messages and data.

SNOMED CT concept identifiers are a sequence of digits that do not reflect the meaning of the concept. There is seldom any value in displaying these to end users. Concept identifiers are simply unique identifiers used within computer systems and are not intended to be used by clinicians.

Description

Each concept is associated with a set of text descriptions, which provide the human-readable form of the concept. Every concept has at least two descriptions (terms) – the fully specified name (FSN) for that concept and a display term in the language being used.

The FSN is a phrase that names a concept in a way that is unique and unambiguous. Each FSN contains a suffix in parentheses that indicates its primary hierarchy, eg `myocardial infarction (disorder)`. The display term is often the FSN without its suffix (hierarchy tag), eg `myocardial infarction`.

All other descriptions are synonyms. Synonyms may be marked as being preferred or acceptable. The preferred term is a common phrase or word used by clinicians to name a concept and is used as default display term for that concept in a particular language or context.

Other synonyms may be marked as acceptable eg `heart attack` or `cardiac infarction`. A list of synonyms for a concept shows the various ways a concept may be described, rather like a thesaurus (see Fig. 9.2).

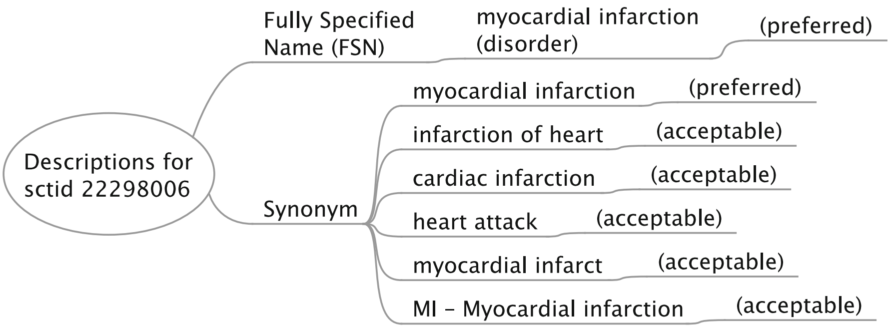

Fig. 9.2 Example descriptions for myocardial infarction

SNOMED CT terms can also be homonyms, where the same term is used for different concepts. The preferred terms and synonyms are not necessarily unique within a language or dialect. For example, the FSN `cold sensation quality (qualifier value)` has a preferred term of `cold`, but `cold` is also a synonym of `common cold (disorder )`. The two meanings of `cold` have different concept Ids.

Each description links a human-readable term with a concept. It has an associated unique numeric *descriptionId*. Terms are encoded using Unicode, which supports all languages. The language code of each description is recorded using ISO 2-character codes (eg `en` for English). The case of the first letter of the term can flagged as having case significance, as in `pH` or `Alzheimer's`.

Relationship

Relationships are the distinguishing feature of any reference terminology such as SNOMED CT. More than 1.4 million relationships have been defined in SNOMED CT and this number continues to grow.

Each relationship is defined as an Object-Attribute-Value triple, which can be processed by a computer. The Object is the source concept – the one that has the relationship, identified by a concept identifier (*sourceId*). The Attribute specifies the type of relationship (*typeId*), and is also a SNOMED CT concept. The Value is the target (*destinationId*).

All relationships are written using a notation known as Description Logic (DL), such as:

```
|concept|:|attribute|=|value|
```

The allowable attributes and values, which may be used to define or qualify concepts, are set out in the SNOMED CT Concept Model (see Chap. 10).

Four different types of relationship (*characteristicType*) are: defining relationship, qualifying relationship, historical relationship and additional relationship.

Defining Relationships are used to define each concept by its relationships with other concepts. Only relationships that are always true are used as defining relationships. Defining relationships specify the concept's supertypes (parents) or defining attributes. Defining relationships are specified as attribute-value pairs, where each attribute and value is a SNOMED CT concept. Supertypes, used in hierarchies are specified using the `|is a|` relationship, which has the conceptId `116680003`.

Every active SNOMED CT concept except the SNOMED CT root has one or more supertypes. Supertype relationships allow users to identify whether a patient with a specific condition has a more general condition that subsumes the specific one. It lets you answer questions such as *"is angina pectoris a type of heart disease?"*

A concept is *defined* if its defining relationships are sufficient to distinguish it from all its supertype and sibling concepts. If a concept is sufficiently defined we can say that another concept, which is represented as a combination of the same defining characteristics, is equivalent to it or a subtype of it. This can be important in search.

Large parts of SNOMED CT are not yet sufficiently defined. *Primitive* concepts are not fully defined and do not have the unique relationships needed to distinguish them from their parent or sibling concepts. For example, pneumonia is a lung disease but unless defining characteristics are specified that effectively distinguish pneumonia from other lung diseases then it is a primitive concept.

SNOMED CT concepts are formally defined by their relationships with other concepts. These defining relationships may be either subtype relationships or attribute relationships. For example the concept `|Appendicectomy|` is a subtype of `|procedure|` and has defining attributes `|method|=|excision|` and `|procedure site|=|appendix|`.

Qualifying Relationships are used in post-coordinated expressions in health data. They are optional non-defining relationships that may be applied by a user or implementer. The range of possible values that can be used in qualifying relationships is constrained by the SNOMED CT Concept Model (see Chap. 10).

Additional Relationships allow non-definitional information to be distributed that may change over time or is specific to a particular national or organizational context (eg that a medicine is prescription only).

SNOMED CT Hierarchies

SNOMED CT is organised into hierarchies in which concepts are explicitly related by subtype relationships to parent concepts immediately above them in a hierarchy. A particular concept may have parents (immediate supertypes) and ancestors, as well as children (immediate subtypes) and descendants.

General concepts are at the top of the hierarchy. At each level down the hierarchy, concepts become increasingly specialized. Unlike a tree-structure, any SNOMED CT concepts can have more than one parent

SNOMED CT has about 19 top-level hierarchies (the number changes from time to time as the system evolves), which descend from a single *Root* concept `|SNOMED CT concept| 138875005`.

Some hierarchies have well-defined sub-hierarchies. For example, the Clinical Finding top-level hierarchy has a sub-hierarchy for Disease (or disorder); the Organism hierarchy has separate sub-hierarchies for Animals, Plants and Microorganisms.

The SNOMED CT hierarchies fall in three main groups: *object hierarchies*, which mainly comprise concepts that are likely to be qualified; *value hierarchies*, which are mainly concepts that act as values in object-attribute-value triples; and a *miscellaneous* group.

A *definitionStatusId* shows whether the concept is *sufficiently defined* (using relationships) or is a *primitive* concept.

SNOMED CT Identifier

The SNOMED CT identifier (*sctId*) is used to identify all types of component, including concepts, descriptions and relationships (see Fig. 9.3).

The is sctId is an integer between 6 and 18 digits long. One way of thinking of the sctId is as a 64-bit integer, although it has an internal structure. The internal structure of the sctId includes a meaningless *item identifier sctid* (between 3 and 8 digits), a 7-digit *namespace identifier* (which is only used in *extensions*), a 2-digit *partition identifier* and a single *check-digit*. The 8-digit item identifier allows almost 100 million items within any namespace.

Extensions are additions to SNOMED CT, usually specific to a single country or organization, and each is identified using a meaningless 7-digit *namespace identifier*, giving a theoretical potential of up to 10 million namespaces. An sctId including a namespace identifier is also known as a *long format sctId*. Namespaces are themselves defined as SNOMED CT concepts within a namespace hierarchy

If no namespace is identified in an sctId, this is a *short format sctId*, and it is assumed that the component is part of the International Release of SNOMED CT.

The *partition identifier* indicates the type of component referred to by that sctId. The *partition identifier* is a 2-digit number. If the first digit of the partition identifier is a zero (0), this component is part of the International Release; if it is a 1, then the component is part of an extension set. The second of the two digits in the partition identifier indicates which of the partitions of SNOMED CT the sctId is identifying, where:

- Concept (0)
- Description (1)
- Relationship (2)

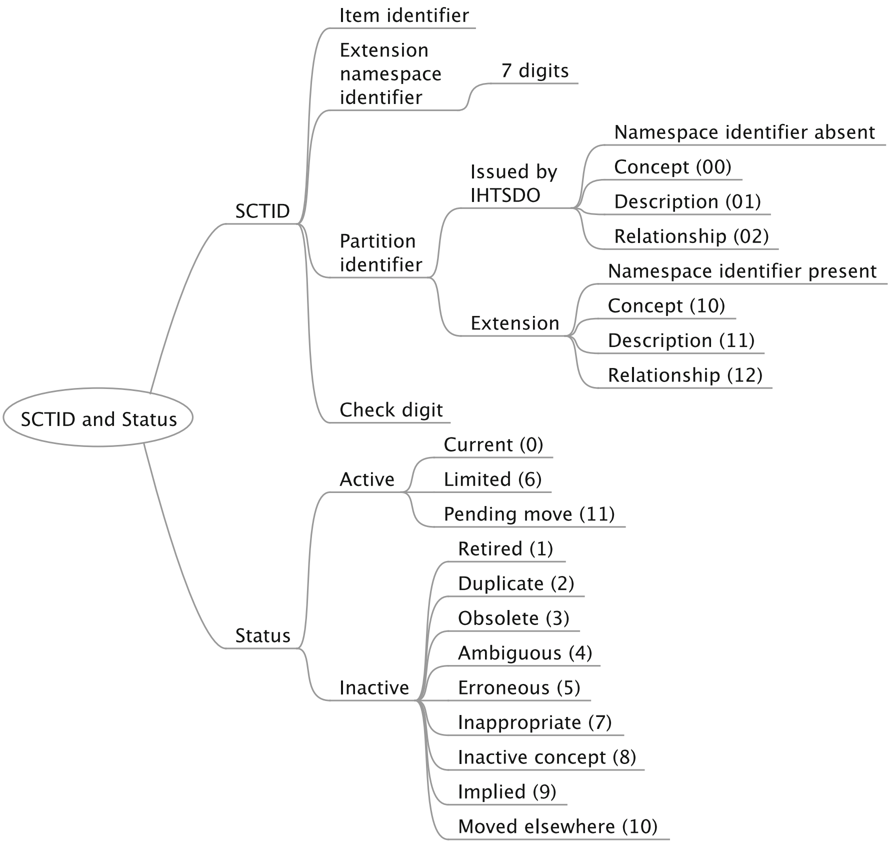

Fig. 9.3 SNOMED CT identifier

The *check digit* at the right hand end of the sctId uses *Verhoeff's Dihedral Group D5 Check* algorithm, which detects all single-digit errors and adjacent transpositions and about 95 % of twin errors and jump transpositions [2].

Expressions

A SNOMED CT expression is a collection of references to one or more concepts used to express an instance of a clinical idea (Fig. 9.4). It expresses an instance of a real world phenomenon (such as a headache) in a particular patient.

An expression is said to be *pre-coordinated* when a single concept identifier is used to represent a clinical idea. Including commonly used concepts in a pre-coordinated form makes the terminology easier to use. All of the 300,000 concepts in the SNOMED CT release are pre-coordinated, which allows a wide range of clinical information to be expressed in pre-coordinated form. It is generally easier

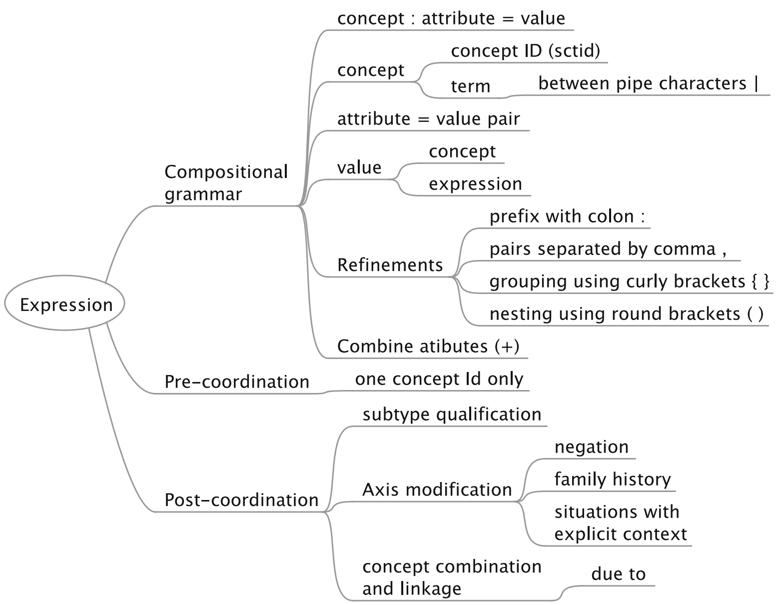

Fig. 9.4 SNOMED CT expressions

to handle pre-coordinated expressions than post-coordinated but this is at the cost of reduced flexibility.

SNOMED CT allows the use of *post-coordinated* expressions to represent a meaning using a combination of two or more concept identifiers. Post-coordinated expressions may be single level expressions or nested to any number of levels of detail. In a nested expression each attribute value is itself an expression, which can be nested. Nested post-coordinated expressions provide a powerful but complex means to allow SNOMED CT to describe things in great detail and cover unexpected requirements.

Compositional Grammar

SNOMED CT expressions are presented using compositional grammar [3]. This same compositional grammar is used to define SNOMED CT Concepts

At the simplest level, a single SNOMED CT concept identifier is a valid expression. Concept identifiers (conceptId) are shown as a sequence of digits. Other sctIds are not usually shown in compositional grammar. For example:

```
80146002
```

A concept identifier may be optionally followed by a term enclosed by a pair of pipe "|" characters. The term must be the term from a SNOMED CT description that is associated with the concept identified by the preceding concept identifier. For example, the term could be the preferred description, or the preferred description associated with a particular translation. The term may include any valid UTF-8 characters except for the pipe "|" character. Whitespace before or after the conceptId is ignored as is any whitespace between the initial "|" character and the first non-whitespace character in the term or between the last non-whitespace character and before the second "|" character. For example:

```
80146002 | appendectomy |
```

A concept identifier (with or without a following term) can be followed by a *refinement*. A colon (:) is used as a refinement prefix, between the concept to be qualified and the qualifying expression. A refinement consists of a sequence of one or more attribute-value pairs. The attribute and the value are both represented by concept identifiers (with or without a following term). The attribute is separated from the value by an equals sign

```
80146002|appendectomy| : 260870009|priority| = 25876001|emer
gency|
```

If there is more than one attribute-value pair, the pairs are separated by commas, representing a logical AND

```
80146002|appendectomy|:260870009|priority|=25876001|emer
gency| , 425391005|using access device|=86174004|laparoscope|
```

Curly braces represent grouping of attributes within a refinement, for example to indicate that the method applies to a specific site

```
80146002|appendectomy|: { 260686004|method|=129304002|ex
cision - action|, 405813007|procedure site - direct|= 181255000|entire
appendix| }
```

The ungrouped attributes, if any, are all listed first, followed by all the grouped attributes.

Round brackets represent nesting to allow the value of an attribute to be refined

```
161615003|history of surgery|:363589002|associated proce-
dure|= ( 80146002|appendectomy|: 260870009|priority|=25876001|emer
gency )
```

Subtype Qualification

Subtype qualification is where the concept is linked with an attribute concept in such a way that the post-coordinated expression is equivalent to a subtype of the unelaborated concept. For example, the concept `|asthma|` could be qualified with the attribute concept |severe| to produce an expression for |severe asthma|, which is a subtype of the concept |asthma|. Where expressions have been post-coordinated and saved in this way the application can compute equivalence and hence subsumption when retrieving the stored expression.

There are four types of subtype qualification:

- Qualification

```
|fracture of femur|:|laterality|=|right|
```

- Refinement of a defining attribute
- Addition of unsanctioned qualifiers
- Addition of nested qualifiers.

Concepts can also be linked together to indicate causality or temporal relationships. For example, |Anaemia| can be linked by the attribute |Due to| to |Ascorbic acid deficiency|. The resulting post-coordination is equivalent to |Anaemia due to ascorbic acid deficiency|.

Axis Modification

Axis modification occurs where elaboration fundamentally changes the meaning of the concept, rather than simply refines it. Such an elaboration of the concept means that it is no longer subsumed as a subtype of a parent concept (subsumption is discussed in the next section). For example, if we elaborate the concept |asthma| to associate it with the mother of a patient, the meaning of the resulting expression has a different meaning from the concept of |asthma| by itself, and therefore different clinical implications. If a clinician runs a query of all instances of asthma in their practice, they would not expect get back instances of asthma linked to a family member.

Concepts that are negated by being post-coordinated with a negation concept (such as |known absent|) have their meanings fundamentally shifted. For instance, |asthma not present| is not a subtype of |asthma|. To say a person does not have |status asthmaticus| (an acute exacerbation of asthma that does not respond to standard treatments) is not to say that they don't have |asthma|.

Negated concepts subsume in the opposite direction to their positive counterparts. Whereas positive expressions are subsumed by more general instances, negated expressions are subsumed by more specific negative expressions. Concepts with axis modification, such as negation and family history must be treated differently from concepts that have been refined through subtype qualification. A special hierarchy of |Situations with Explicit Context| is used to express this type of concept that overrides the standard contextual defaults of SNOMED CT.

Subsumption Testing

Subsumption testing is used in information retrieval because most research, audit and decision support applications usually assume that a supertype includes all its subtypes (children and descendants). For example, a project may need to identify all patients with diabetes, which implicitly assumes that all types of diabetes should be included.

In any subsumption test there are two SNOMED CT expressions (or codes), one of which is tested for subsumption by the other. The *candidate expression* is tested to see if it is subsumed by (is a descendent of) another expression. The *predicate expression* is tested to see if it subsumes (is an ancestor of) another expression.

A **transitive closure table** is a list of all the ancestors of each concept. Transitive closure tables provide a fast direct way of checking whether one concept is a subtype (child) of any other and provides a means of high-performance subsumption testing. High-speed subsumption testing is essential for clinical decision support and is very useful in all types of analysis and reporting.

Reference Sets

Reference sets (refsets) are important in the practical application of SNOMED CT. Reference sets provide a single mechanism for referencing and adding information to SNOMED CT components (Fig. 9.5). All reference sets have common metadata, but the fields (columns) used vary according to use and purpose. When using a system, a user is only interested in a tiny proportion of the whole of SNOMED CT and reference sets provide a way of enabling this.

The number of members in a reference set may vary enormously. A language reference set may have hundreds of thousands of members. A realm reference set containing the concepts commonly used in a clinical specialty may contain several thousand members, but the set of concepts or descriptions in context reference sets for a specific clinical protocol, template or data entry field may only contain a few members.

One way to think of a reference set is as an index entry pointing to a set of pages relevant to a topic. Each reference set member is uniquely identified by a UUID, can be inactivated using the *active flag* and is versioned using an *effectiveTime*. Reference sets are extensible with additional columns and are not limited by fixed specifications.

Language Reference Sets allow descriptions to be set for a language, dialect or context of use. SNOMED CT can be translated into any language or dialect. Each translation uses existing concepts with new language-specific descriptions. A language Reference set is a set of references to the descriptions that make up that language or dialect edition. For example, British English (`en-GB`) and US English (`en-US`) are different dialects of English in which many medical terms have different spellings; English, French and Spanish are different languages.

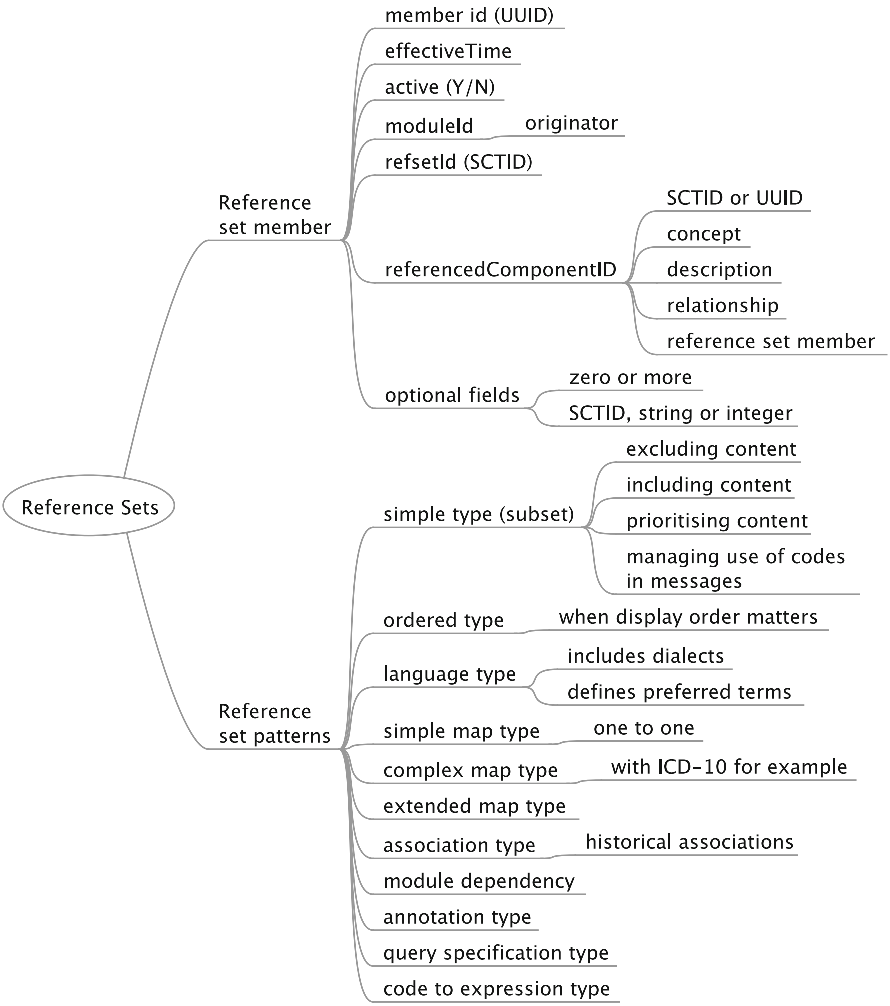

Fig. 9.5 Reference sets

Realm Reference Sets cover the terminology used in a specific area of expertise or locality. Examples of realms include: a specialty, a professional discipline, an organization, a country, or a specialty within a country (eg US dentists).

Context Reference Sets identify components that are included in or excluded from the set of values that can be used in a particular context. Simple reference sets are not necessarily mutually exclusive and the content of different reference sets may overlap. For example this type of reference set may be used to limit the content of a field to those permitted by an interoperability message standard.

Navigation Reference Sets provide an alternative tree-view of a set of terms in a specified order. Navigation hierarchies can reflect the way that people think when entering data, sometimes referred to as the model of use. Navigation hierarchies are useful for display, navigation and data entry. These are usually handcrafted, to limit the number of levels, the number of choices at each level, to list terms in a sensible order and ensure consistency over time. However, large numbers of handcrafted hierarchies are difficult to maintain. Each reference set member includes a reference to the parent concept, a child concept and the sequence order of that child.

Map Reference Sets are used to reference other terminologies and classifications, such as the International Classification of Diseases (ICD). Mapping is needed to allow data collected for one purpose such as clinical care to be used for another purpose such as reimbursement, avoiding the costs and errors of having to re-enter data. It is also needed when data needs to be migrated to newer systems. Ideally, computer programs will use the mapping tables to translate codes automatically, but unfortunately the rules of many coding systems, such as ICD-9 CM and ICD-10 are such that fully automated coding is not yet feasible.

A single SNOMED CT concept may need to be mapped to one or more target codes. The map from SNOMED CT to ICD-10 CM involves two main reference sets: a Descriptor reference set and a Complex Map reference set. Other reference sets may also be involved but are less important.

The Descriptor reference set contains metadata that describes the attributes of all publication reference sets and their information content. The ICD-10 map is one item in this file. The metadata items are themselves SNOMED CT concepts in the foundation metadata concept hierarchy.

The complex map reference set contains one or more map records for each source concept mapped, including the ICD-10 target codes. Each SNOMED CT concept may have none, one or more mappings to ICD-10 CM.

History Reference Sets SNOMED CT includes component history files, which maintain a record of changes to existing components, in line with the principle of permanence. The reasons for why it is inactive may be declared in a reference set. These reasons may include being: retired, duplicate, obsolete, ambiguous, erroneous, inappropriate, inactive concept, implied, or moved elsewhere.

Reference Set Development

The process of reference set development and maintenance is challenging and time-consuming. It is no easier to develop and maintain a reference set that has the support of a large clinical community than it is to develop any other consensus standard. Clinicians want reference sets that meet their particular needs, complete and yet focused. Reference set development is likely to remain a growth area for many years to come.

A number of tools have been developed to help with the tasks of building, maintaining and using reference sets.

Creating a new reference set requires access to a namespace in order to generate the sctId needed. Within that namespace, at least one module ID concept (with an FSN and preferred term) is required under the *module* sub-hierarchy (within the Core Metadata) for each authoring organization. The steps required to create a new reference set include:

- Create the reference set concept in the foundation metadata hierarchy.
- Create the descriptor for the reference set (by adding members to the reference set descriptor reference set).
- Add members to the reference set.

For each reference set, a formal document should record the rules, principles and approach used to determine the members of that reference set.

A typical reference set development project is likely to involve the following steps:

- Establish scope and team and identify which existing pattern (if any) can be used.
- Identify relevant terms from existing records and evidence base (literature)
- Compare to SNOMED CT content.
- Derive reference sets, including hiding some of the complexity of SNOMED CT, allocating priorities and the sequence order of terms.
- Validate using panels and in practice for comprehensiveness, relevance, reliability and usability.
- Implement and deploy software that enables users to achieve their goals.
- Maintenance.

Reference sets need to be maintained and the content re-examined when new releases of SNOMED CT are made available. Processes need to be established to address any concepts that have become inactive and new concepts added in each new release.

Releases

The SNOMED CT International Edition is released twice a year by IHTSDO. This may be supplemented by national Extensions. All releases use the same file formats, known as RF2 (Release Format 2), which was introduced in 2011. Three release types are supported.

A *snapshot* release contains the most recent version of every component. This is useful for installing SNOMED CT.

A *delta* release contains only those components that have been changed in any way since the previous release. This is useful for updating a system.

The *full* release contains every version of every component that has ever been released. It is voluminous and provides a complete historical record and can be used to obtain a view of the state of any component at a particular time.

The core terminology data files are a Concepts file, Descriptions file and Relationships file. Reference set files have common metadata but a number of different structures.

The pattern for release file names consist of five elements, each separated by an underscore "_" and followed by a full stop "." and a file extension:

```
<FileType>_<ContentType>_<ContentSubType>_<Country|Namespace>_<Ve
rsionDate>.<Extension>
```

The FileType specifies the type of file and the release format. For example the main terminology files have File type *sct2*, where the *2* tag refers to Release Format 2.

Documentation

The SNOMED CT Starter Guide, the Technical Implementation Guide and the Editorial Guide are three key reference documents, available from IHTSDO, which describe SNOMED CT in detail. These are aimed at different audiences and contain a good deal of overlap.

SNOMED CT Starter Guide [4] (56 pages) provides a good overview of a range of topics associated with SNOMED CT, organised into 15 Chapters.

SNOMED CT Technical Implementation Guide (TIG) [5] (757 pages) is intended for SNOMED CT implementers, such as software designers who need an authoritative point of technical reference and advice to support their involvement in designing, developing, acquiring or deploying software applications. It includes sections on: implementation, structure and content, release file specifications, the Concept Model, terminology services, record services, change management and extension services.

SNOMED CT Editorial Guide [6] (171 pages) describes editorial policies regarding the purpose, scope, boundaries, requirements, concept model, hierarchies, terming, and other policies related to the content in SNOMED CT. It is primarily intended to guide those who are responsible for editing the content of the International Release, but secondarily is important for those creating extensions.

References

1. Spackman K. SNOMED User Group Meeting. Chicago. 2005.
2. Wagner NR, Putter PS. Error detecting decimal digits. Commun ACM. 1989;32(1):106–10.
3. SNOMED CT Compositional Grammar Specification and Guide. The International Health Terminology Standards Development Organisation, Copenhagen. Version 2. 03 July 2015.
4. SNOMED CT Starter Guide. IHTSDO. December 2014.
5. SNOMED Clinical Terms Technical Implementation Guide. IHTSDO. January 2015.
6. SNOMED CT Editorial Guide. IHTSDO. January 2016.