Binding SNOMED CT to Information Models - An Introductory Guide

MARCH 2016

Date: 2016-03-29
Version: 0.01
Status: DRAFT

TABLE OF CONTENTS

Copyright and Acknowledgements

The document is a draft publication of the International Health Standards Development Organisation (IHTSDO), the association that owns and maintains SNOMED Clinical Terms. IHTSDO acknowledges the work undertaken by Kirstine Rosenbeck in co-authoring this document during her participation in the SNOMED CT Implementation Advisor (SIA) scheme.

1 Introduction

1.1 Background

SNOMED CT is a clinical terminology with global scope covering a wide range of clinical specialties and requirements. The use of SNOMED CT in Electronic Health Records (EHRs) provides a standardized way to represent clinical meanings captured by clinicians and enables the automatic interpretation of this meaning. Using SNOMED CT in clinical records offers many benefits from enabling guideline and decision support systems, through to supporting reporting, clinical research and population health monitoring.

Most EHRs are designed and developed using one or more information models, which describe the information that is collected, stored, communicated and displayed. Some information models are designed for a specific proprietary system, while others are based on a common health information standard (e.g. HL7 FHIR resource, HL7 CDA template, ISO-13606 archetype). Information models may also be defined using a wide variety of representations (e.g. UML class diagram, database table design, Archetype Definition Language, or XML Schema). Irrespective of the purpose, design and representation of the information models, however, the use of clinical terminology is an important part of making the models complete and useful.

1.2 Purpose

The purpose of this document is to provide a practical starting point for anyone who needs to understand the basics of binding terminology to information models. This includes the why, what, when, where and how of terminology binding, with a specific focus on using SNOMED CT in information models.

1.3 Scope

This document is written as an introductory guide to binding SNOMED CT to information models. In presenting this topic, however, the broader scope of binding any terminology to information models is also described.

This guide does not provide:

- An introduction to SNOMED CT. It is recommended that the reader has some familiarity with SNOMED CT prior to reading this guide. For a useful introduction to this topic, please refer to the *SNOMED CT Starter Guide*[1]
- Detailed implementation guidance. Implementation-specific guidance is available for some standard implementation formats – for example, in HL7 Information models[2].
- Best-practice guidance. This document primarily focuses on presenting the alternative approaches that may be used in binding terminology to information models. It is anticipated,

however, that best practice guidance will be developed subsequent to the publication of this Introductory Guide.

1.4 Audience

The target audiences of this document include:

- IHTSDO members and affiliates;
- SNOMED CT designers and developers, including designers and developers of EHR systems, information models, data entry interfaces, storage systems, decision support systems, retrieval and analysis systems, communication standards and terminology services;
- SNOMED CT users, including people responsible for clinical information entry, content management and utilization;
- SNOMED CT terminology developers, including concept model designers, content authors, translators, map developers, subset and constraint developers and release process managers;
- SNOMED CT educators and trainers;
- Individuals involved in healthcare strategy or the procurement of systems that use SNOMED CT.

1.5 Topics

The topics covered in this *Introductory Guide to Binding SNOMED CT to Information Models* are:

- What is terminology binding?
- Why is terminology binding important?
- What is the purpose of terminology binding?
- Why is understanding SNOMED CT important for terminology binding?
- When should terminology binding be done?
- Where and how should terminology binding be represented?
- Which information model artefacts should be bound to terminology?
- Types of terminology binding
- Value set binding
- Model meaning binding
- Metadata
- Principles
- Learning more
- Use cases

Each of these topics is presented in one of the following chapters.

2 What is terminology binding?

2.1 Definition

A terminology binding is a link between an *information model artefact* and a *terminology artefact*. Terminology artefacts that may be used in a binding include codes, expressions and value sets. When a terminology binding is created, the *terminology artefact* is said to be *bound* to the *information model artefact*. Information model artefacts that may have a terminology binding include data groups, data elements, data type attributes, and the information model itself.

Terminology bindings must contain enough information to identify both the information model artefact and the terminology artefact involved in the binding. They must also include metadata to describe the type of link being established, and any other additional information which may be required to correctly interpret the binding.

Terminology bindings each serve a specific purpose – for example, they may be used to record the set of possible values which can populate a given coded data element in the information model; or they may define the meaning of an information model artefact using a concept or expression from the terminology.

2.2 Information models

Information models are a formal description of how information may be structured, interrelated and accessed. This includes multiple types of models that are used to collect, store, communicate, and display clinical information, as well as the structured clinical knowledge designed to support the health sector. Information models that may require binding to terminology include:

- User interface models which describe the way content is captured and displayed in the user interface;
- Information storage models which support the real-time storage of patient data in clinical information systems;
- Shared clinical information models which allow a consistent specification of content in clinical information systems across global, national or regional perspectives (e.g. openEHR archetypes, ISO13606 archetypes and HL7 FHIR profiles);
- Clinical reference models upon which other information models are based for consistency (e.g. openEHR reference model, the HL7v3 RIM and HL7 FHIR resources);
- Communication models including messages and services (e.g. HL7 messages and ISO13606 extracts);
- Data warehousing and analytics models which support the efficient querying and analysis of clinical information; and

- Knowledge models including models used to hold clinical guidelines and rules, such as Map of Medicine.

These different types of information models are illustrated below in Figure 1.

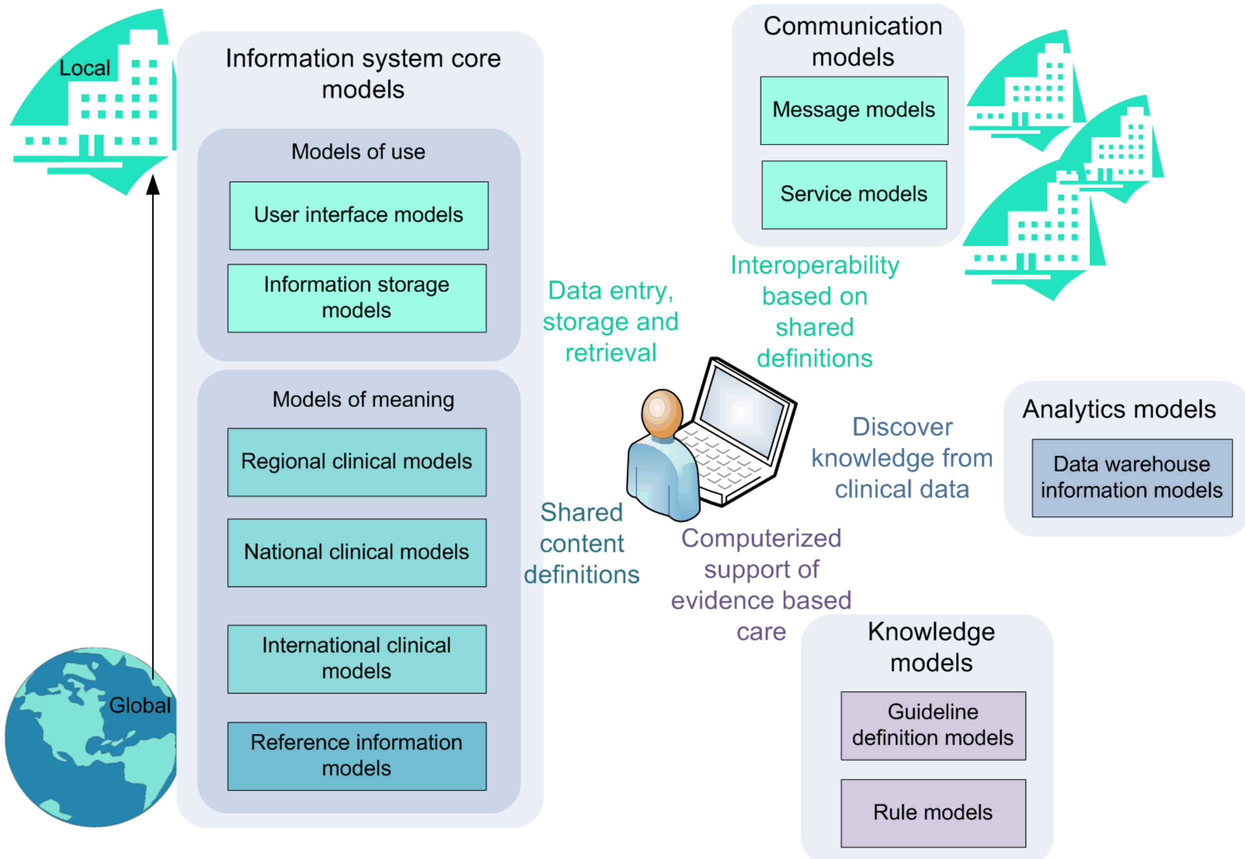

Figure 1 - Information models

It is important to note that many healthcare systems today are built on proprietary information models, which are directly represented by the implemented database structures. These systems often use more standardized models for communication. However, even when only a few of the above types of information models are used, terminology binding is nearly always an important part of making the models complete.

For further details about specific information modelling techniques please refer to Appendix A.

2.3 Terminology

In this guide, the primary focus will be on terminology binding to SNOMED CT. However, we cannot ignore the fact that most information models also need to bind to other terminologies (e.g. LOINC, ICD, and local demographic codes). There are a number of different types of health terminologies available which each serve a different purpose including:

- Interface terminology (e.g. local coded terms, SNOMED CT) which defines clinical terms in the language of the user, and is therefore used to support data entry and retrieval of clinical information;
- Reference terminologies (e.g. SNOMED CT, LOINC) which are concept oriented and provide a consistent representation of shared meaning;
- Classifications (e.g. ICD-9, ICD-10 and ICPC) which are spatial, temporal or spatiotemporal segmentations of the world, and which use mutually exclusive categories. Classifications are mainly used for aggregation purposes, such as statistics and reimbursement;
- Other terminological resources, which serve a variety of other purposes e.g. UMLS and MeSH. MeSH is used to support the classification and retrieval of knowledge sources such as PubMed articles.

These different types of health terminologies are illustrated below in Figure 2.

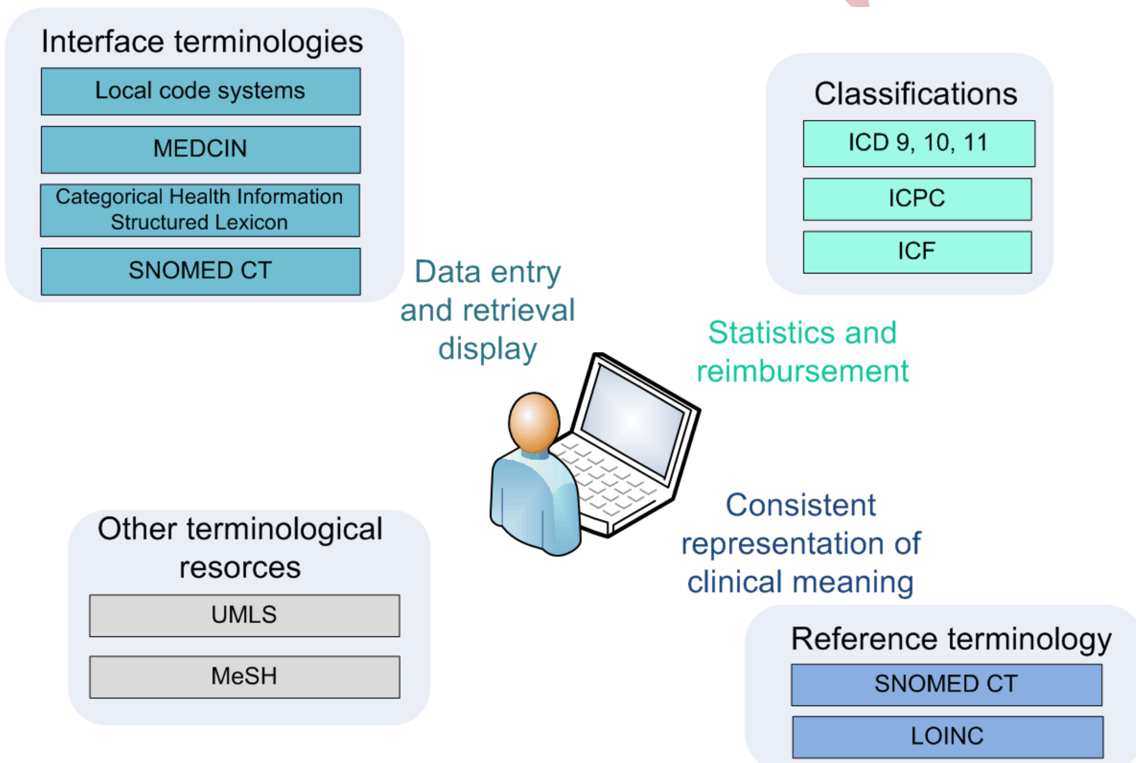

Figure 2 - Health terminologies

2.4 Representation

Information models can be represented using a variety of different formats, including:

- Formal data model diagrams, such as Unified Modelling Language (UML) class diagrams, Entity-Relationship (ER) diagrams and Object Role Modelling (ORM) diagrams;
- Machine-readable data representation languages, such as Archetype Definition Language (ADL), XML Metadata Interchange (XMI) format, XML Schema, JSON and SQL data definition language; and
- Other visualizations, such as mindmaps, hierarchical diagrams, logical database table designs, and user interface designs.

For the purposes of this guide, however, most of the examples presented will use a simple hierarchical diagram to illustrate the information model artefacts to which terminology is bound (as shown on the left side of Figure 3).

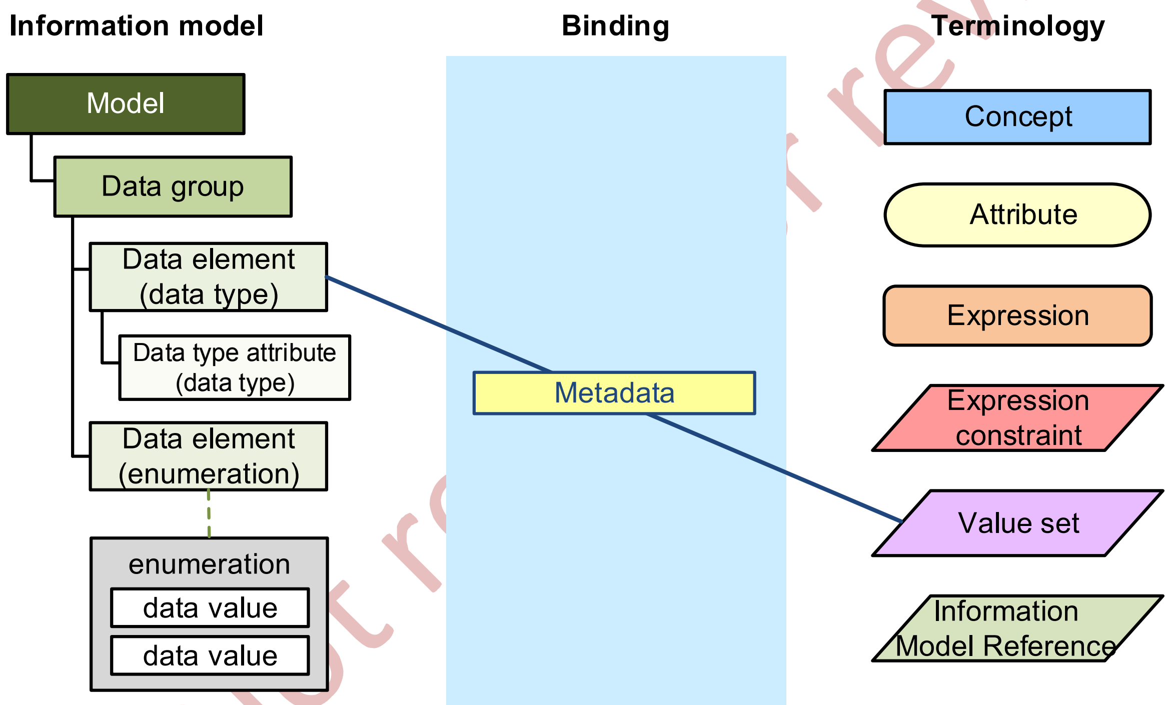

Figure 3 – Graphical representation of terminology binding

These hierarchical diagrams illustrate the following information model artefacts:

- Model: The information model as a whole (e.g. 'Procedure' model);
- Data group: A subpart of the model that logically groups data elements and/or other data groups together (e.g. 'Medication list' data group);
- Data element: A part of the model that is assigned a single (possibly complex) data type (e.g. 'Method' data element);
- Data type attribute: A subpart of the data type associated with a given data element (e.g. 'Units' attribute of the 'Quantity' datatype); and
- Enumeration: A set of data values extensionally defined within the information model.

Despite the simple information model representation used in this guide, the terminology binding approaches and principles are equally applicable to the other model representations. For further details about information modelling artefacts please refer to Appendix A.

Similarly, a number of approaches to representing terminology artefacts are available, including lists, tables, hierarchies and description logic rules. For the purpose of this guide, however, we will use the SNOMED CT Diagramming Guidelines, with a few notation extensions where required (as shown on the right side of Figure 3). The notation used includes the following terminology artefacts:

- Concepts: A single clinical meaning referred to using a unique identifier. For example, 233604007 |pneumonia|.
- Attributes: A type of relationship that may exist between two concepts. For example, 363698007 |finding site|. Attributes are themselves a type of concept.
- Expressions: Structured combinations of one or more concept identifiers used to express an instance of a clinical idea. For example, 284196006 |burn of skin|: 363698007 |finding site|= 33712006 |skin of hand|.
- Expression constraints: Computable rules that can be used to define a set of concepts and/or expressions. For example, << 64572001 |disease| (which refers to the descendants of the concept 64572001 |disease| plus the concept itself)
- Value sets: Sets of concept codes and/or post-coordinated expressions. Some value sets are implemented as a SNOMED CT reference set. For example, 571751250000000102 |surgery list reference set|".
- Information model references which refer to an information model artefact or its value. For example, $ Method refers to the value of the 'Method' data element.

Note that the expression constraint language makes it possible to express SNOMED CT components, expressions and value sets. However, for this guide to be illustrative, it is important to be able to distinct the types of terminology artefacts easily. Consequently, all these six terminology artefacts have a distinct graphical representation, see Figure 3.

The middle column in Figure 3 illustrates the terminology binding itself, which links the information model artefact on the left of the diagram to the terminology artefact on the right. The metadata, which is used to describe the terminology binding link further, is represented by a yellow box on the line that represents the binding.

3 Why is terminology binding important?

3.1 Overview

Terminology binding is important because there are no such things as a terminology independent information model and an information model independent terminology. Reliable interpretation of meaning depends on

- The way information is structured
- The way clinical concepts are represented
- The way the terminology is used within the structure

Consequently, the interface between structural and terminological representations have to be explicitly defined i.e. a consistent approach to terminology binding is necessary. The importance of terminology binding is illustrated in Figure 4.

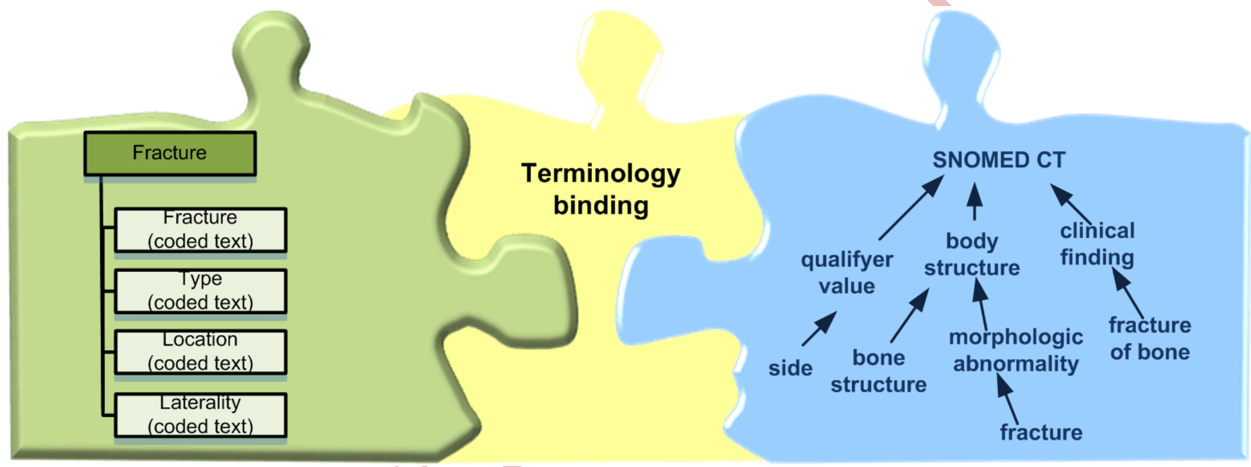

Figure 4 - Importance of terminology binding

For some use cases, solely relying on either information model or terminologies can be meaningful, but terminology binding is important whenever the link between information model and terminology is significant in achieving specific business or clinical objectives. In the following section the boundary between information models and terminology is described in more details.

3.2 Balancing structure and terminology

Figure 5 illustrates the information types that are best represented using terminology in the blue part of the model. In the green part of the model, the information types that are best represented using information models are illustrated. In the yellow part of the model, the information types that can be represented using both information model artefacts and terminology artefacts are illustrated. In the yellow area, consistent terminology binding approaches are most important because they will help combine information model and terminology meaningfully. The meaningful combination of terminology and structure will allow identification of semantically similar representations and help avoid over or under specification. Over specification is when some attribute is represented both in the terminology artefact and in the information artefact. For example, if 'X-ray of the shoulder' is included in the value list for 'Procedure', then 'Shoulder' should not be included in the value list for 'Body site' and vice versa. Under specification is when the information model artefact presumes that some attribute is specified within the terminology, and it is not, and vice versa.

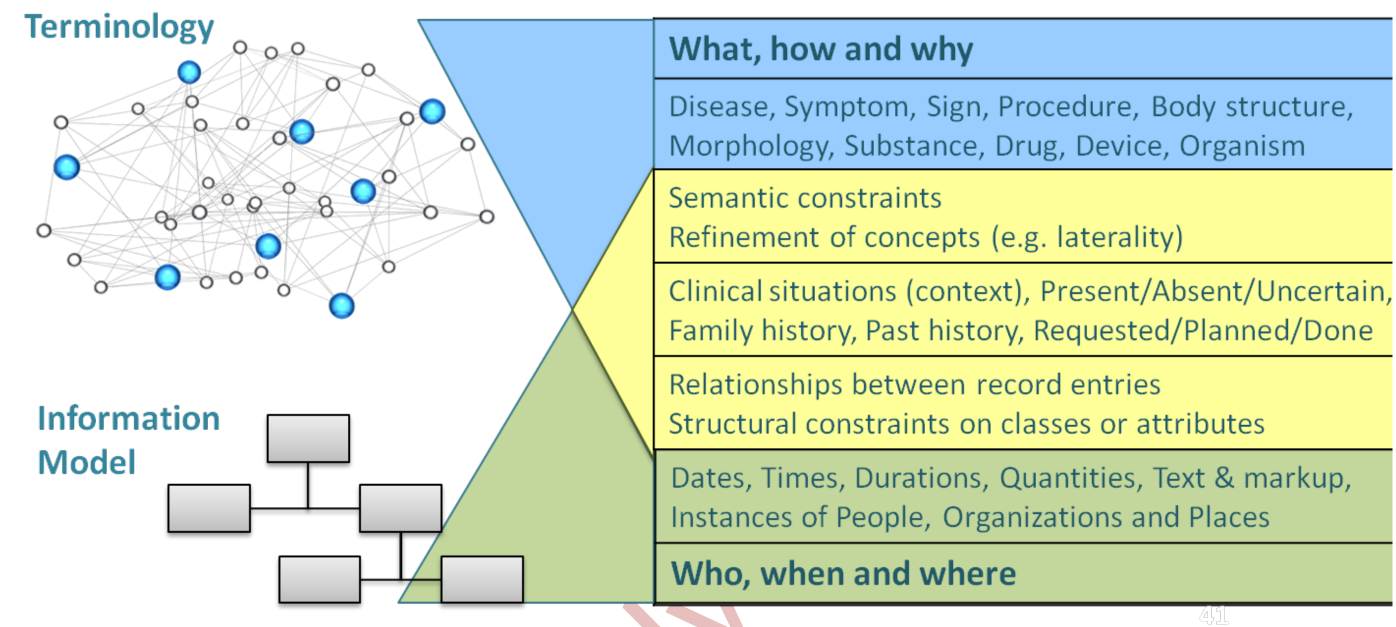

Figure 5 – Representation of information by terminology and information models respectively.

In the next section, examples of information types that can be represented using both information model artefacts and terminology artefacts will be presented.

3.3 Alternative representations

The same clinical idea can be expressed in different ways. One representation could be a user interface with a list of different diagnoses e.g. "asthma", "diabetes" and "COPD" with radio button adjacent to each with the possibilities of ticking "yes", "no" or "unknown". A second representation of a similar meaning could be a user interface with a field called diagnosis with a value list where "asthma", "diabetes", "COPD" etc. can be chosen. If a hospital manager want to query on the number of asthma patients it is important with a consistent representation of the structural and terminological elements in these two interfaces to receive a reliable answer. Terminology binding is an important part of achieving this goal.

Another example is when the same information model structure is populated with different amounts of pre-coordination. Figure 6 shows different representations of suspected lung cancer with different amounts of pre-coordination. A consistent approach to terminology binding will support the identifications of these three instances as semantically equivalent and help transforming them to a common model of meaning.

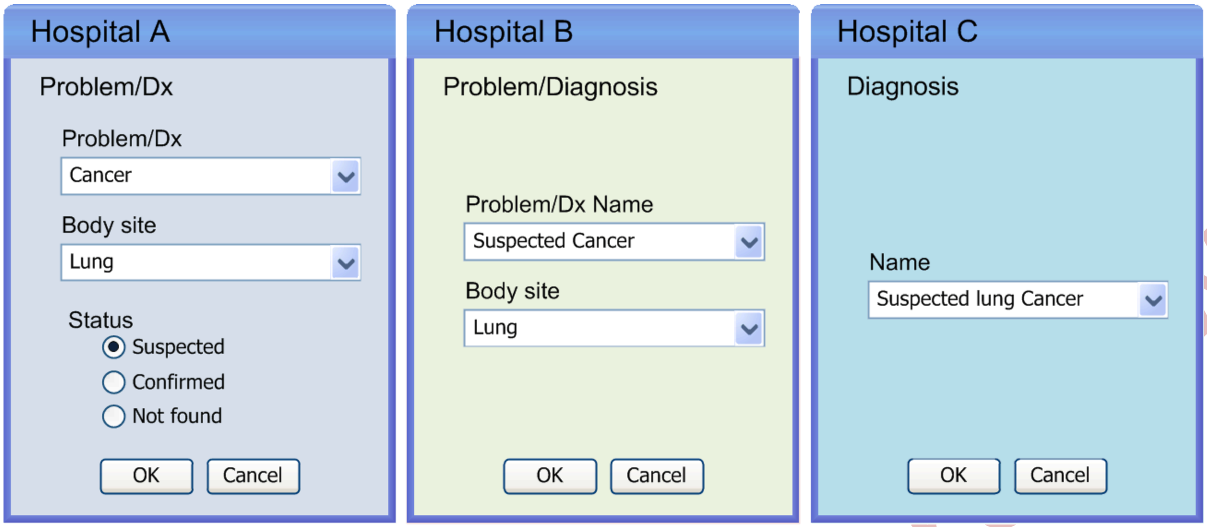

Figure 6 - Different representations of suspected lung cancer

However, identifying a common model of meaning is a challenge, because of the overlap between terminology and information model e.g. an information model attribute might indicate "absence" or "negation" and a SNOMED CT context can be express "known absent". Consequently, negation can be expressed two different places and opens the possibility of double negation, and double negation is difficult to interpret e.g. the negation of conscious is unconscious, but what is the negation of unconscious? It can be interpreted

- logically. Then the answer is "conscious"
- as an error, then the answer is "unconscious"
- an indication that a person in risk of loss of consciousness remained conscious. Then the answer is "no loss of consciousness"

To avoid misinterpretation there need to be clear decisions, within organizations, about the way information model and terminology semantics combine.

3.4 Meaning preservation

In some cases, a terminological representation is preferable to an information model representation to preserve meaning. The solutions are preferable because they make consistent retrieval of information easier. In the following two recommendations are presented, but there may be equivalent design constructs where meaning preservation can be strengthened my using a terminological representation.

1. Terminology binding should wherever possible avoid coded values with the meaning equivalent to 'Other'. Instead, the information model should use a sufficiently general code, which captures all meanings not included in the list. Note that if an information model binds to a classification codes with meaning equivalent to 'Other' should not necessarily be avoided because they typically plays an important role in using classifications for statistics.

2. Information models should wherever possible avoid the use of Boolean attributes, and instead use terminology values to elucidate the meaning of the positive and negative. Note that the user interface may display terminology codes (e.g. 'asthma present', 'asthma absent') as "Yes", "No" if preferred by the end user

4 What is the purpose of terminology binding?

4.1 Overview

While clinical information systems may be developed without terminology binding, consistent terminology binding becomes crucial if meaningful information sharing, integration with knowledge sources or analytics are required. Terminology binding is an important part of supporting the following clinical information system functions:

- Data capture;
- Retrieval and querying;
- Information model library management; and
- Semantic interoperability.

In this chapter, we consider each of these four functions and explain how terminology binding plays an important role in supporting them. We also present a number of activities and use cases to illustrate ways in which terminology bindings can be used. In Chapter 9 we explain the types of terminology bindings suitable to support each activity, and in Chapter 14 we present some detailed examples which show how the techniques described in this document can be applied to these activities.

4.2 Data capture

Terminology binding plays an important role in data capture by defining the valid values for data entry, enabling integration with clinical decision support tools, supporting exception handling and providing flexible data entry options. In Figure 7, we show the main data capture activities of which terminology binding is an important part. The primary participant in these activities is the clinician recording patient information at the point of care.

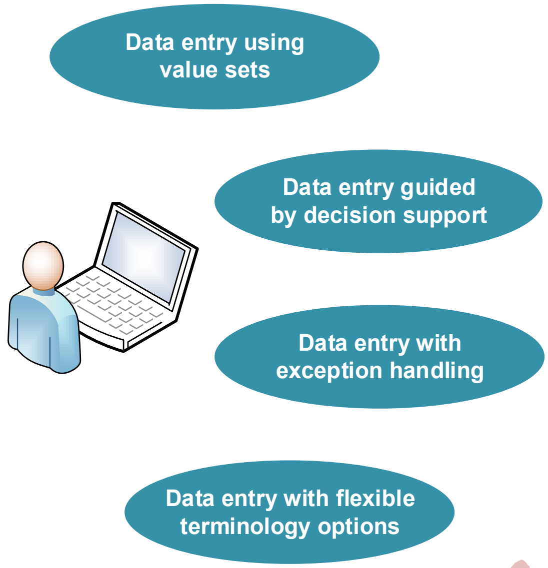

Figure 7 – Data capture activities using terminology binding

Each of these activities is described in more detail below. For a description of the types of terminology bindings required to support each of these activities, please refer to section 9.4.2. For examples of how terminology binding may be applied to support data capture, please refer to section 14.2.

4.2.1 Data entry using value sets

Terminology binding can be used to establish the link between a user interface model and a set of valid values. By restricting the values entered to a predefined set, data consistency can be improved, which in turn increases the effectiveness of data retrieval and other analytics functions. It may also help the user to better understand what to enter into the given field, and enable the tool to provide effective data entry features such as drop-down value lists, word auto-complete, search by synonym, and selection using navigation hierarchies. See more about this in the search and data entry guide [3].

> ***Use Case 1:*** *A local hospital implements a national "discharge summary" standard, which uses a national diagnosis value set to define the valid values for the 'diagnosis' data element. This helps the hospital to provide a drop-down value list on their discharge summary user interface and add word auto-complete functionality. It also enables all hospitals in the country to use a consistent diagnosis list, from which population-health studies can be performed.*

4.2.2 Data entry guided by decision support

Terminology binding helps to support the integration of clinical systems with decision support tools by providing a list of values upon which the rules can be written. When using SNOMED CT as the terminology, additional features such as hierarchies, defining relationships, reference sets and expression constraints can also be used to support more sophisticated decision support rules.

Use Case 2: *A medical practice has decided to implement a clinical decision support tool to guide the prescribing of antibiotics. The tool requires the local medical record to supply information about the patient's symptoms, based on a terminology binding to SNOMED CT. The tool then uses these symptoms to recommend suitable antibiotics to the clinician. The symptoms can be specialized (via SNOMED CT's is a hierarchy) to improve the precision of the recommended antibiotics.*

4.2.3 Data entry with exception handling

In some situations, dependencies may exist between the values entered into different fields. The values that are valid to enter into one field, for example, may depend on the values that have previously been entered into other fields. Terminology bindings can define these dependencies, and thus support the identification of exceptions during the data entry process.

Use Case 3: *A clinical department has identified the fact that some male patients are being registered for female-specific procedures (and vice versa). To detect these avoidable errors, a terminology binding is added in the clinical system which uses the sex of the patient to determine the set of valid values for the procedure field. As a result, it is no longer possible to register male patients for female-specific procedures (and vice versa).*

4.2.4 Data entry with flexible terminology options

Different work practices and clinical preferences affect the amount of terminology precoordination that clinicians prefer in their data entry forms. In some systems, both precoordinated and primitive codes are requested as alternative ways of entering exactly the same semantics. In these situations, it is important to ensure that any clinical meaning that may be recorded in multiple different ways is checked for consistency (to avoid contradictory statements being made).

Use Case 4: *A regional emergency department has some clinicians that prefer recording fractures as pre-coordinated concepts (e.g. "open fracture of tibia"), while others prefer to record the fracture and then describe the affected bone (e.g. "tibia") and morphology (e.g. "open"). The hospital fears that different data entry styles may lead to inconsistent fracture descriptions (e.g. an "open fracture of tibia" combined with a morphology of "closed"). Terminology bindings are therefore introduced to ensure that fields with semantics overlap are populated consistently.*

4.3 Retrieval and querying

Terminology binding plays an important role in data retrieval, clinical querying, reporting, research and analytics. It does this by providing a predefined set of values over which to query and by defining the meaning of the information model in a queryable way. Binding to a rich reference terminology such as SNOMED CT enables queries to also utilize the logic-based definitions of each concept to provide more advanced capabilities. In Figure 8, we show the main retrieval and querying activities of which terminology binding is an important part. In this diagram we illustrate the two main types of users - clinical users who usually retrieve information about individual patients, and health authorities or administrators who query over groups of patients.

Each of these activities is described in more detail below. For a description of the types of terminology bindings required to support each of these activities, please refer to section 0. For examples of how terminology binding may be applied to support retrieval and querying, please refer to section 14.3.

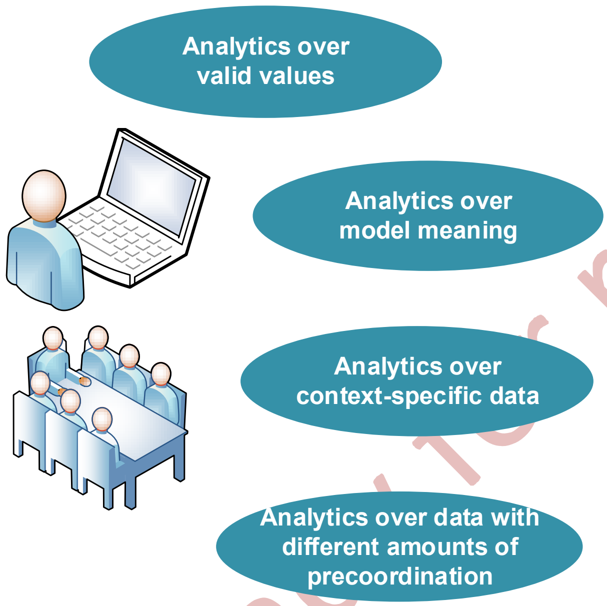

Figure 8 – Data capture activities using terminology binding

4.3.1 Analytics over valid values

Terminology binding helps by providing the set of valid values over which querying can be performed. In many cases, this enables the range of possible query values to be enumerated, which can assist in the development of comprehensive queries. When SNOMED CT is used in the binding, queries can also perform aggregation over the stored data (supported by the 'is a' hierarchy) and can filter the query results based on the defining properties of the stored values.

> ***Use Case 5:*** *A patient comes into the emergency department with an ECG indicating a Myocardial Infarction. Before having angioplasty, the surgeon wants to know whether the patient has previously suffered from heart disease or unusual cardiovascular signs. The surgeon performs a search on the patient's health record to find any events that contain a diagnosis that is a descendant or self of 56265001 |heart disease|. This longitudinal search is much faster than manually searching through all patient records from earlier encounters.*

4.3.2 Analytics over model meaning

Terminology binding can also help to define the meaning of a model, so that users can more easily identify which information models, data groups and data elements are relevant to a query. This enables analytics to be performed based on the meaning of the models rather than just the values.

> ***Use Case 6:*** *A clinician wants to review all information for a given patient that is recorded as a cardiovascular finding. This includes free text (e.g. describing heart-related symptoms), quantitative measurements (e.g. blood pressures, heart rates etc) and coded findings (e.g. cardiovascular-related diagnoses). As part of this query, the clinician searches on all models, data groups and data elements whose meaning is a descendant or self of 106063007 |cardiovascular finding|. (To complete the search, the clinician also looks at the coded values which are descendants of self of 106063007 |cardiovascular finding| which is analytics over valid values)*

4.3.3 Analytics over context-specific data

Context, such as whether a procedure is planned, requested or performed and whether a finding is confirmed, suspected or excluded can be vital to the correct design of a query and the correct interpretation of its results. Context such as this, however, may be recorded in many different ways within both the data and the model structures themselves. Terminology binding can provide a mechanism by which this context can be located in a consistent and reproducible way, so that queries are shielded from the complexities of searching all the possible combinations of context representations.

> ***Use Case 7:*** *A researcher wants to count how many patients at each hospital in the region had an appendectomy within a given time period. To ensure that the query does not count procedures that were planned but not performed, the researcher uses the terminology binding to exclude those appendectomies which were only planned during the relevant time period. The terminology binding provides a consistent way of indicating the 408730004 |procedure context| of each procedure, whether this is represented in the model structure or the coded values themselves.*

4.3.4 Analytics over data with different amounts of precoordination

As discussed in subsection 4.2.4, different clinical systems use different amounts of terminology precoordination in their coded values. In some systems, both precoordinated and primitive codes are used as alternative ways of entering exactly the same semantics. In these cases, terminology binding can help to ensure a consistent interpretation of query results, irrespective of whether the values are precoordinated in a single field or primitive codes stored in an equivalent structural format.

> ***Use Case 8:*** *In the regional emergency department described in use case 4, some fractures are recorded as precoordinated concepts (e.g. "open fracture of tibia"), while others are stored as a group of fields storing individual values (e.g. bone="tibia", morphology="open"). A clinician wants to find all patients with a lower limb fracture, so she search for all 125605004 |fracture of bone| where 363698007 |finding site| is a descendent or self of 72001000 |bone structure of lower limb|. The query engine then uses the terminology bindings to identify both precoordinated and structural representations that match this search criterion.*

4.4 Information model library management

Managing a library of information models can be extremely challenging when the library contains a large number of different models covering a variety of clinical specialties and types of clinical encounters. Terminology binding can play an important role in effective information model library management by enabling models to be searched, validated and compared based on the meaning of the model and its contents. In Figure 9, we show the main information model library management activities of which terminology binding can play an important part. The participants in these activities are health information modellers at hospitals, health authorities or standards organisations.

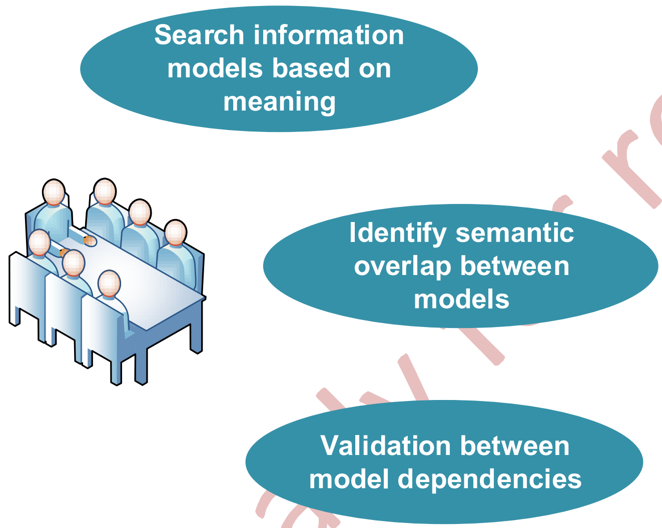

Figure 9 – Information model library management activities using terminology binding

Each of these activities is described in more detail below. For a description of the types of terminology bindings required to support each of these activities, please refer to section 9.4.4. For examples of how terminology binding may be applied to support information model library management, please refer to the use case examples in section 14.4.

4.4.1 Search information models based on meaning

Terminology binding helps to support the searching of information models in a library based on the meaning of the models and its contents. This is particularly important when the information model library is large or has been built up over a period of time by multiple users. When SNOMED CT is used to define the meaning of the models and its contents, searching can be done using

- synonyms of the meaning of the models;
- abstract concepts which group together models with more specific meanings; and
- properties of the model meanings.

Use Case 9: *A regional healthcare department has a large library of clinical information models to support its shared Electronic Health Record. To avoid creating duplicate models, the agreed business process is to search the existing library of models for a topic before developing a new information model in this area. A clinical modeller has been asked to create a new information* *model for a cardiovascular physical examination. Before starting, the modeller first checks the existing library of models by searching for models whose meaning is 363003006 |cardiovascular physical examination|. When no models are found that match this search, the modeller then checks all models whose meaning is a descendant or self of 5880005 |physical examination| to consider whether any existing model can be reused or specialised.*

4.4.2 Identify semantic overlap between models

When information model libraries become large or are managed by multiple users (or both), it is quite easy for multiple unrelated models to be created with similar clinical content. When this occurs, querying over the data recorded becomes more challenging, as there is less consistency between the models that store data with a similar meaning. One approach to addressing this problem is to provide tools that can analyse the semantic overlap between models in a library, to identify potential cases in which two or more models would be more effectively designed as a single merged model, or as specializations of each other. One example of a semantic overlap analysis can be found in this scientific study [4].

Use Case 10: *A clinical modeller is performing a quality review of the information model library. Using the model meaning terminology bindings, she discovers that there is significant semantic overlap between the vital signs model and the cardiovascular measurements model. To make querying over cardiovascular measurements easier and more consistent, irrespective of which model is used to record them, she decides to redesign these two models to ensure that they both reuse the same common model subcomponents for heart rate and blood pressure.*

4.4.3 Validation between model dependencies

Terminology binding can be used to ensure that dependencies between models are consistent. There are two main types of model dependencies, which may require validation – namely:

1. Model specialisation: This involves adding constraints to a broader model to meet more specific requirements. For example, a national 'discharge summary' model may be specialised into a regional 'discharge summary' model, or an 'observation' model may be specialised into a 'body temperature' model. When a model is specialised from a broader model, it is useful to be able to validate that its meaning, the meaning of its contents, and its associated value sets are all specialisations of those in the broader model.
2. Model composition: This involves combining a number of small model fragments into larger models to meet more complex requirements. By doing this, smaller models (e.g. 'serum creatinine test') can be defined once and then consistently reused in multiple larger models (e.g. 'complete blood count', 'basic metabolic panel'). When composing a model inside another model it is useful to be able to validate that the meaning of the smaller model is compatible with the location in which it is being incorporated into the larger model.

For more information on the recommended principles for the validation of model dependencies please refer to chapter 13.

Use Case 11: *A national model for medicine prescriptions is developed and agreed upon by a particular country. A regional hospital within this country, wishes to develop two different* *prescription models (one for small children, and one for adults) to cater for the differences in the permitted drugs, dispensation methods and dose sizes between these patient groups. Both these prescription models are required to be specialisations of the national prescriptions model. Using the terminology bindings, conformance testing is performed by comparing the meaning and value sets for each node in the regional models against each node in the national model to ensure that these are consistent with a specialisation. This quality assurance process helps to ensure that regional prescribing information can be effectively shared and queried across the country.*

4.5 Semantic interoperability

Terminology binding plays a key role in supporting semantic interoperability by defining the data values that are agreed upon for information sharing and by enabling the consistent representation of meaning irrespective of whether it is incorporated in the data values or the information model. While full semantic interoperability is difficult to achieve, terminology binding can help to make significant progress towards this goal. In Figure 10, we show the main semantic interoperability activities of which terminology binding is an important part. Users are any health organisations that needs to communicate and authorities that decides on how standardized information models and terminologies may be combined to achieve interoperability in health. For more information on supporting semantic interoperability by combining models and terminology, see e.g. the European Committees report on Semantic Health [5].

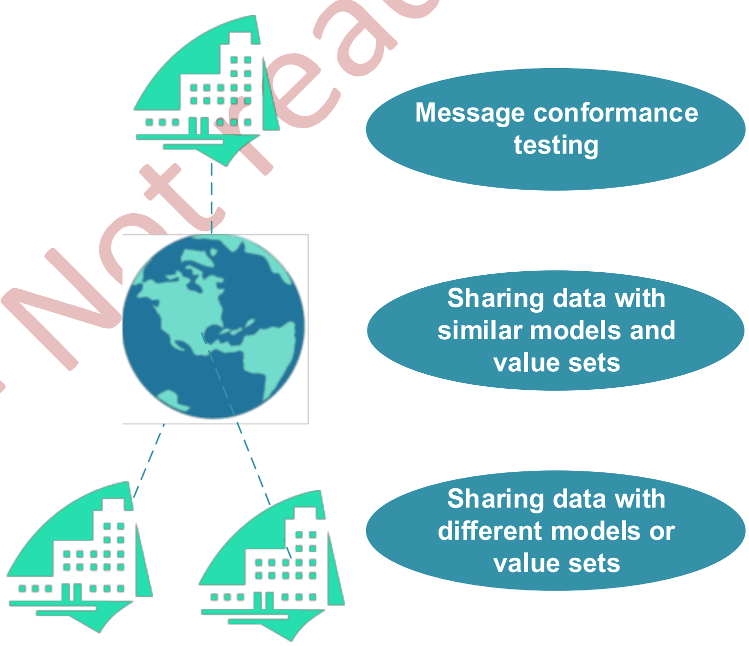

Figure 10 - Semantic interoperability activities using terminology binding

Each of these activities is described in more detail below. For a description of the types of terminology bindings required to support each of these activities, please refer to section 9.4.5. For examples of how terminology binding may be applied to support semantic interoperability, please refer to the use case examples in section 14.5.

4.5.1 Message conformance testing

To improve the level of semantic interoperability between systems, it is common practice to use a shared messaging model. However, to ensure that each message is populated in a consistent way it is also important to use an agreed value set for each coded data element. The value sets assigned to each coded data element by the terminology binding can be used during message conformance testing, to check that each coded value in the message is in fact a member of the correct value set.

> ***Use Case 12:*** *A national standard for discharge summary messages has been agreed upon to ensure that they can be shared and queried in a consistent manner. To support this standard, a national testing service has been established that tests each message for conformance against the standard. This service checks that each coded value included in the discharge summary comes from the agreed national value set. For example, each diagnosis must come from the national SNOMED CT value set for diagnoses. Messages that do not conform to this standard run the risk of not being able to be meaningfully interpreted by the receiving system.*

4.5.2 Sharing data with similar models and value sets

When data is shared between systems which use similar models and value sets, the terminology bindings can help by providing a common understanding of each model and set of values. In these situations, simple approaches to terminology binding are usually sufficient to achieve the required levels of semantic interoperability. In some cases, mappings between semantically similar value sets from different code systems are required in order to achieve interoperability between corresponding coded data elements.

> ***Use Case 13:*** *A university hospital and a nearby regional hospital agree to share cancer patient records, due to the high degree of shared care that is performed between the hospitals. To support this sharing they decide to develop a common information model for chemotherapy treatment and compliance. The university hospitals always calculate the chemotherapy dose and needs detailed patient information to do so. The regional hospital only needs summary data and a possibility to add compliance information. Consequently, the common model is extended and specialised to support their individual information requirements. When information is transferred between the hospitals a mapping enabled by the terminology binding is performed between the two models and their corresponding value sets.*

4.5.3 Sharing data with different models and value sets

When data is shared between systems which use different models and value sets, semantic interoperability is harder to achieve. Different information models can use different amounts of precoordination in the terminology, and the same semantics can be represented using different information structures. Binding to a common terminology such as SNOMED CT, enables a consistent representation of the meaning of data represented using each model, irrespective of whether this meaning is captured in the data values or in the model itself. In this way, terminology binding can support semantic interoperability by enabling the transformation of data instances between different information models.

> ***Use Case 14:*** *An orthopaedic department receives information about a patient fracture from a medical imaging centre. In the message, the fracture is described using two data elements: one for the type of fracture and one for the laterality (e.g. fracture type = 71555008 |closed fracture of ulna|, laterality = 7771000 |left|. In the orthopaedic department's system, fractures are described using three data elements: one for the type of fracture, one for the morphology and one for the laterality (e.g. fracture type = 54556006 |fracture of ulna|, morphology = 20946005 |closed fracture|, laterality = 7771000 |left|). Using the terminology binding, combined with the logic-based definitions in SNOMED CT, the data is transformed from the two data elements used in the sending system, to the three data elements required by the receiving system.*

5 Why is understanding SNOMED CT important?

5.1 Overview

In this chapter, we consider why it is important to understand the basic content and principles of SNOMED CT in order to create consistent and therefore useful terminology bindings.

Consistent concept selection with a strong relation to the intended purpose of the information model is a prerequisite for meaningful entry, querying and communication of patient information. Inconsistent terminology binding is in the best case purposeless and, in the worst case, a threat to patient safety.

Historically, there has been a perception that it was possible to develop clinical information models independent from terminology and that terminology value sets could be developed so that they would work with any information model. However, practical consequences of interdependency between terminology and structural information models are often underestimated, for details on the boundery between terminology and information model, see 3.2. Consequently, the IHTSDO recommends that there is a collaborative development between healthcare information models and SNOMED CT to achieve effective implementation. Guidance on creating consistent terminology binding is an important part of reaching this goal because the terminology binding specifies the interface between information model and terminology.

Information models and terminology artefacts have to be bound together consistently to ensure meaningful entry and usage of patient information in the clinical setting. A consistent terminology binding framework is also a necessity for analytics and interoperability use cases.

5.2 Consistent binding to value sets

Binding information models to value sets typically requires that the values are draw from the same top-level hierarchies in SNOMED CT. In most cases, the further down in the SNOMED CT hierarchy that the least common parent of the values is located, the more coherent the value set. A coherent value set can be a benefit for query use cases. For example in use case 5, section 4.3.1, searching for 56265001|heart disease| only retrieves relevant information if data has been consistently coded using a descendant (or self) of 56265001|heart disease|. This means that if a physician discovers a patient with extra heart beats and decides that it is pathological, it should be coded as 78250005|Ectopia cordis (disorder)| (which is a descendent of 56265001 |heart disease|). However, a purely lexical search on "etopic heart" (without consideration of the SNOMED CT hierarchies in which the concepts are located) could easily lead the physician to use the concept 408674005|Ectopic heart structure (morphologic abnormality)|. This would be incorrect, as this concept represents the abnormal body structure, but does not represent the disease or clinical finding itself. For this reason, the concept 408674005 |Ectopic heart structure (morphologic abnormality)| would not appear in the results of a search for 56265001 |heart disease|. An information modeller could have helped the physician by making a heart disease value set containing only descendants of 56265001 |heart disease| and binding it to the information model in which the physician enters heart related diagnoses.

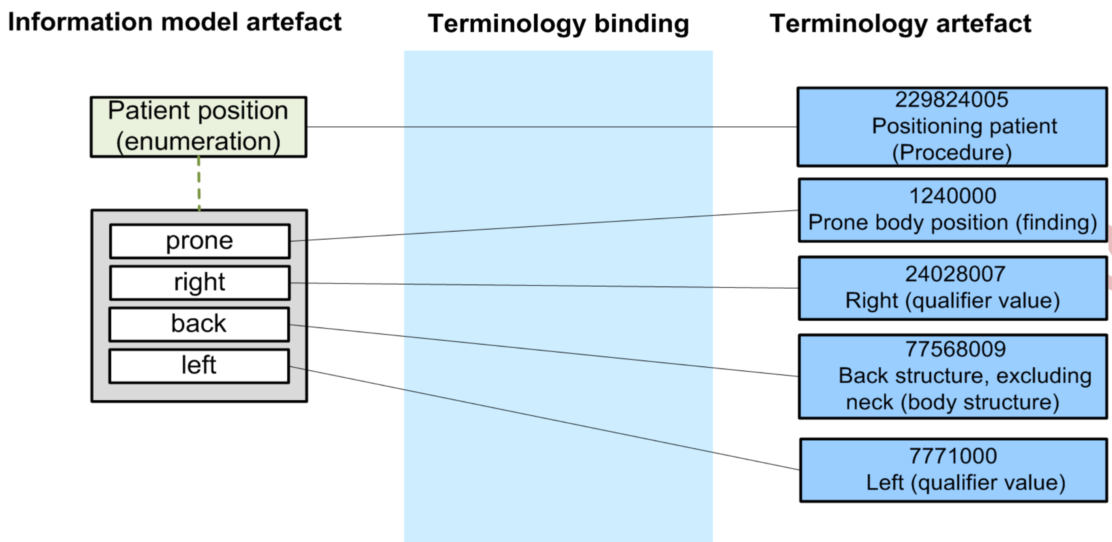

Figure 11 - Inconsistent terminology binding of patient position model

In Figure 11, another example of an inconsistent terminology binding is presented. The data element as well as each item in an enumeration is bound to a SNOMED CT concept, but the SNOMED CT concepts are drawn from various hierarchies, which make the purpose of the model as well as the consequences of using resulting patient information unclear. In the example, it is not clear whether a procedure or an observation is registered i.e. is the purpose changing the patient position or observing the patients current position? Moreover, a query to examine a patient's position could e.g. search for descendants of 9851009 | finding of position of body and posture |, but this would only give a result if the patient has had a prone body position because 1240000|Prone body position| is the only descendant of 9851009 |finding of position of body and posture | in the enumeration. Looking at the terminology artefacts using the SNOMED CT hierarchy, see Figure 12(a), reveals the non-coherent use of terminology.

In Figure 12(b), value set concepts are shown that subsumes *9851009 |finding of position of body and posture|*. This increases the coherency of the concept selection.

In Figure 13, the resulting terminology binding is shown. The data-element is bound to *271605009|Position of body and posture (observable entity)|*. In SNOMED CT *9851009 |finding of position of body and posture|* interprets *271605009|Position of body and posture (observable entity)|* which means that the structure of the information model becomes consistent with the structure of SNOMED CT's content model. Choosing the observable entity specifies that it is in fact an observation of the patients current position, and not a procedure to change the patient position. The consistency between information model and concept model is the theme in the next section.

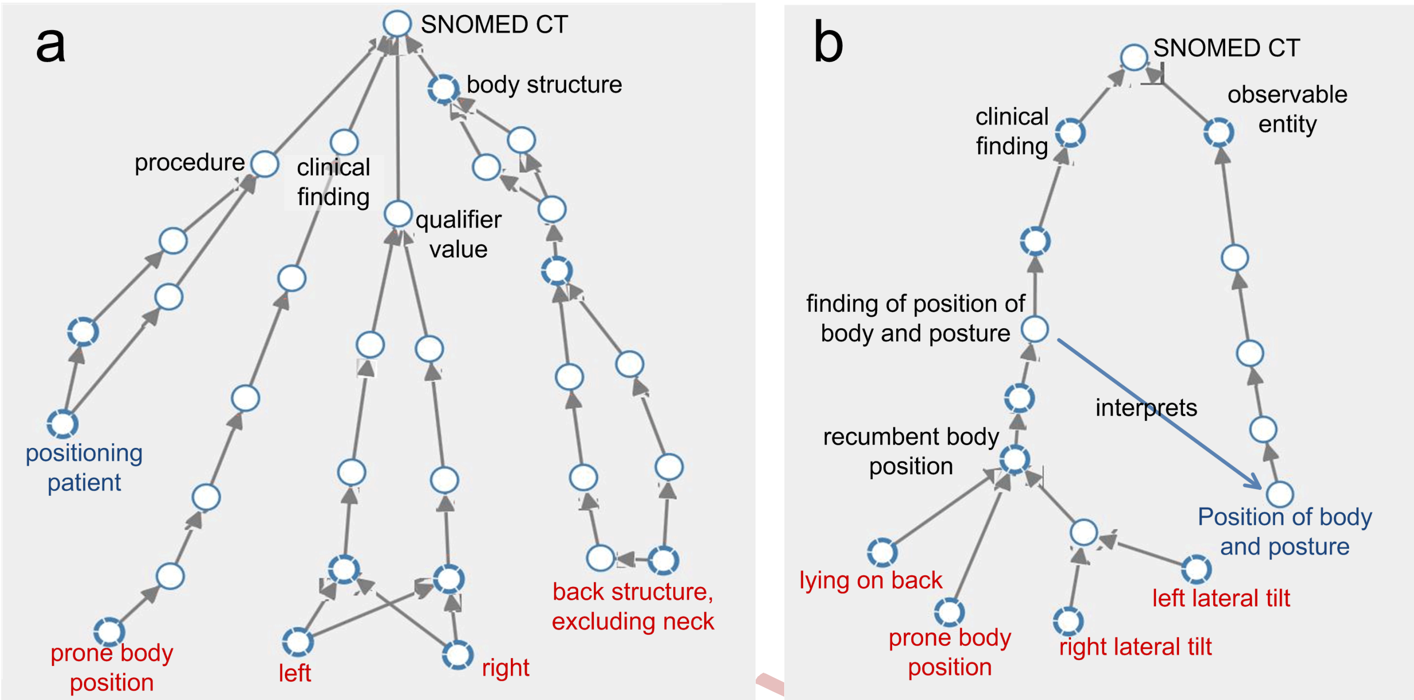

Figure 12 - (a) Inconsistent concept selection - (b) Coherent concept selection

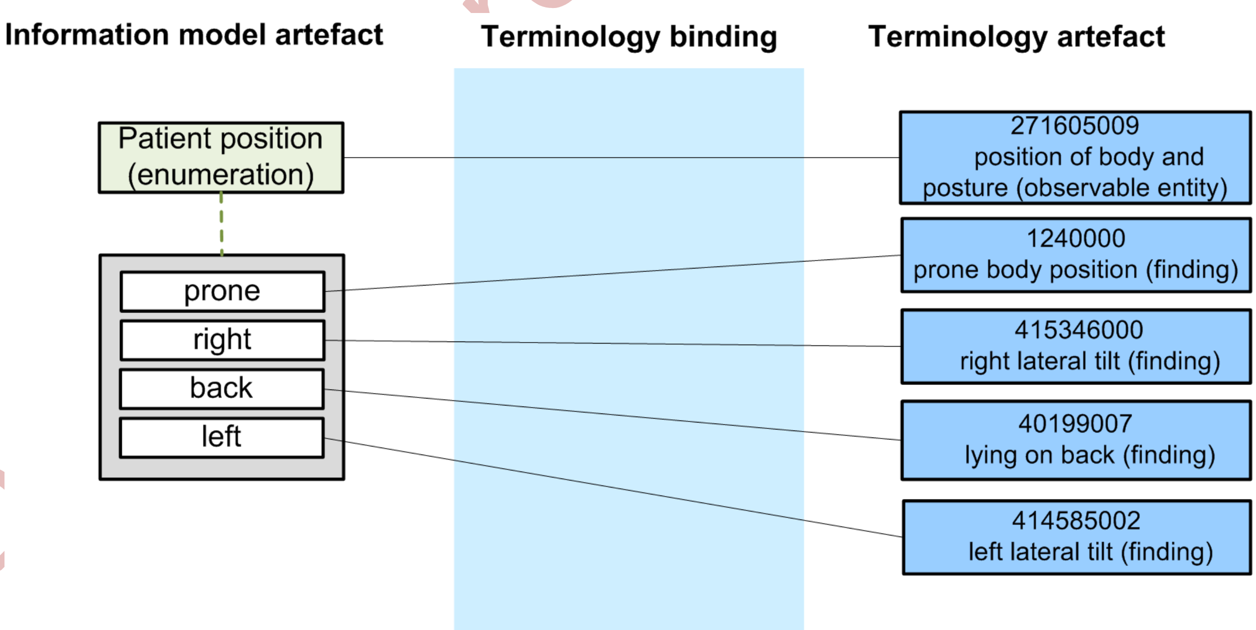

Figure 13 - Consistent terminology binding of patient position model

5.3 Consistent binding to model meanings

When aiming at binding consistently to model meaning, it is important that the concept model is not violated. In Figure 14 a fracture model is intended to represent patient fractures with additional attributes that clarifies different aspects of the fracture. Representing the fracture model with terminology requires content drawn from different SNOMED CT hierarchies, whereas a value set (as illustrated in Figure 11), should typically be drawn from the same hierarchy. Choosing the right hierarchies is dependent on what the intended purpose of the binding is, and what each element in the model semantically represents. The bindings in Figure 14 are literal, which results in inconsistencies that makes the bindings wrong and useless. E.g. The data element "Type" is bound to 410656007|Type|, which matches in a literal sense, but looking at the parent concept 118598001|Property of measurement| reveals that semantically this cannot represent anything related to a fracture. Figure 15(a) show the inconsistent bindings in the SNOMED CT hierarchy. In section 14.2, a consistent value set binding is illustrated for the fracture example. Making a consistent binding like this requires conformance to the concept model. This is illustrated in Figure 15(b) where the content model relationships (in blue) bind the different concepts used in the terminology binding together meaningfully. In Figure 15(a), one relationship exist (associated morphology), but when comparing to the terminology binding in Figure 14, it can be observed that the direction of the relationship is not expressed correctly. Refer to section 14.2 to see the correct terminology binding.

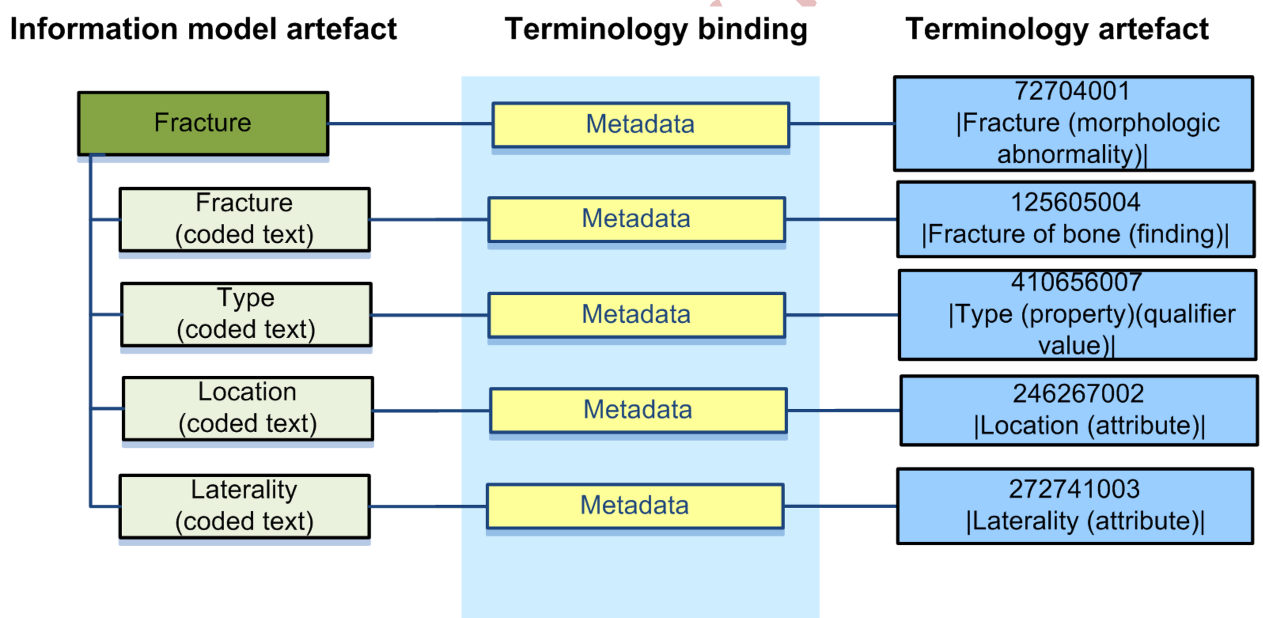

Figure 14 - Inconsistent terminology binding of a fracture model

Consistent concept selection with a strong relation to the intended purpose of the information model is a prerequisite for meaningful entry, querying and communication of patient information. In chapter 9, it is explained how to choose the right type of binding for a specific intended purpose.

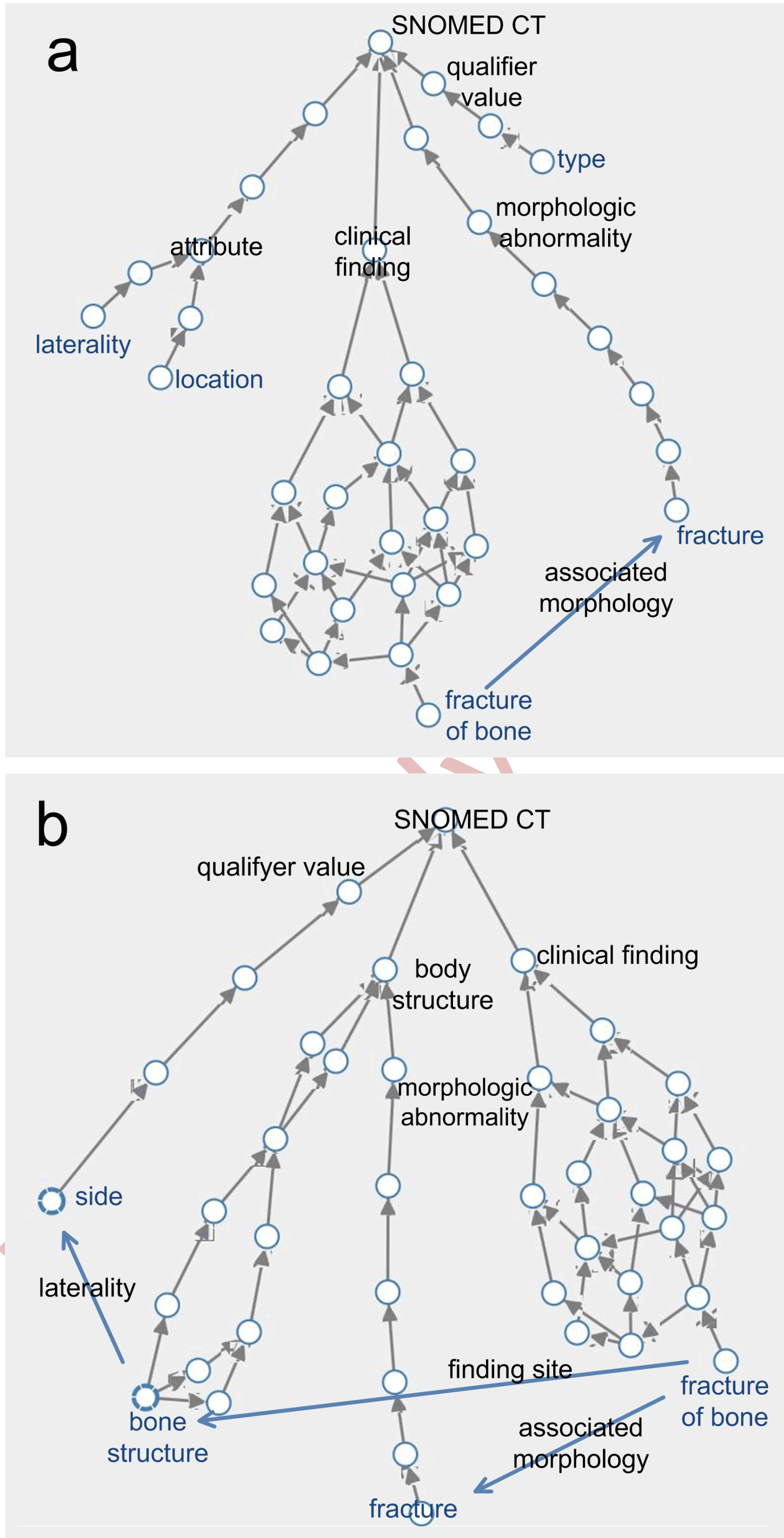

Figure 15 – (a) Inconsistent concept selection - (b) Coherent concept selection

6 When should terminology binding be done?

6.1 Overview

A classic approach to designing information models is to separate requirement, information modelling and terminology binding. But these siloed design processes tend to underestimate the importance of a consistent modelling approach where considerations about the terminology concept models should be included. Clinical involvement in modelling and terminology activities may also be de-emphasized in siloed processes. Consequently, terminology binding should be integrated with the information model development process, and collaboration between the healthcare professionals, information modellers and terminologists should be encouraged. The following chapter presents the challenges of a siloed design process and make recommendations for a collaborative design process.

6.2 Challenges of a siloed design process

A siloed design process is illustrated in Figure 16 and in the example below.

> ***Design of a radiology report 1:*** *A hospital decides to design a radiology report for an EHR system. Health care professionals are responsible for the requirements i.e. they hand over existing forms and guidelines, describe needed information and existing workflows. The requirements which are formulated as use cases allow information modellers to design appropriate information models. Relevant information models could be a storage models and clinical models that handle the organization of information within the EHR system. Interface models would also be relevant because radiologist should be able to enter radiology information and other health care professionals at the hospital should be able to access the information. The last step in the simple design process would include a terminologist that binds the designed information models to relevant standardised terminologies.*

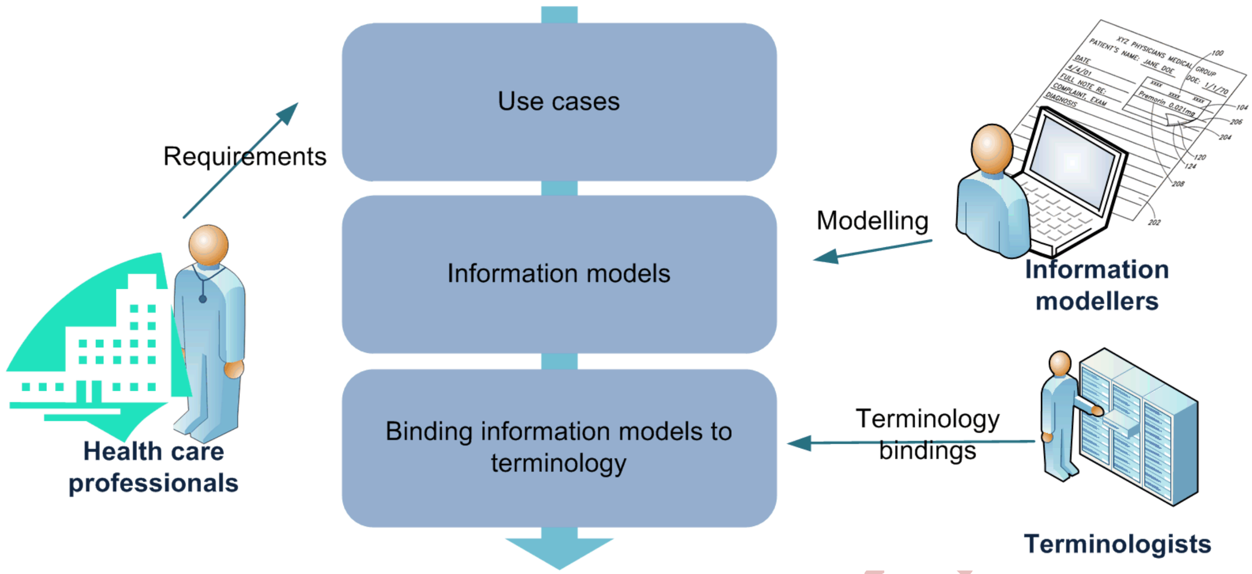

Figure 16 - A siloed information model design process

For small well-defined projects, such a simple design process might be possible, but it has several challenges that hinder applicability for more complex projects:

- **Consistent modelling** is a challenge. The simple design process lacks activities where information modellers re-use existing models or patterns
- **Model-terminology consistency** is a challenge. The simple design process lacks activities where the SNOMED CT concept model is allowed to influence information structure and content
- **The quality of the terminology bindings** is a challenge. The problem is that health care professionals are not involved in concept selection and information modellers propose an interface terminology before the model reaches the terminologist. Consequently, terminologists are left with the task of approximating an existing design with a standardised terminology. This increases the risk of non-matches or partial matches and categories that does not make sense from a semantic viewpoint. It is well-known that bridging clinical language, clinical pragmatics, and standardised terminology is a challenge [6]. A design process should take this into account.
- **Clinical validation** is a challenge. Requirements in health are complex and design processes are challenging .Getting feedback from Health Care Professionals several times during the design process is important. Using rapid prototyping or a participatory design approach could be useful methods to achieve this. Read more about the relation between clinical requirements and design in e.g. [7].

In the next section a comprehensive design process is proposed that takes into account the identified challenges.

6.3 Recommendations for a collaborative design process

Collaboration between different professionals and significant focus on consistent modelling as well as clinical requirement fulfilment are important aspects of a collaborative design process. In Figure 17, a design process for information models is suggested. The figure should be read as recommendations regarding activities that could improve collaborative information modelling and terminology binding. The figure should not be read as a finished plan which can be followed in any information modelling project. Generally, design processes progress as illustrated by the direction of the arrow in Figure 17. However, design processes are also iterative which means that each activity might require adjustment of earlier activities, for simplicity reasons this is not illustrated in the figure. Figure 17 also suggest that information modeller, terminologist are different persons with each their responsibility. In many cases, information modellers or information managers fill all these roles and ensure consistent modelling.

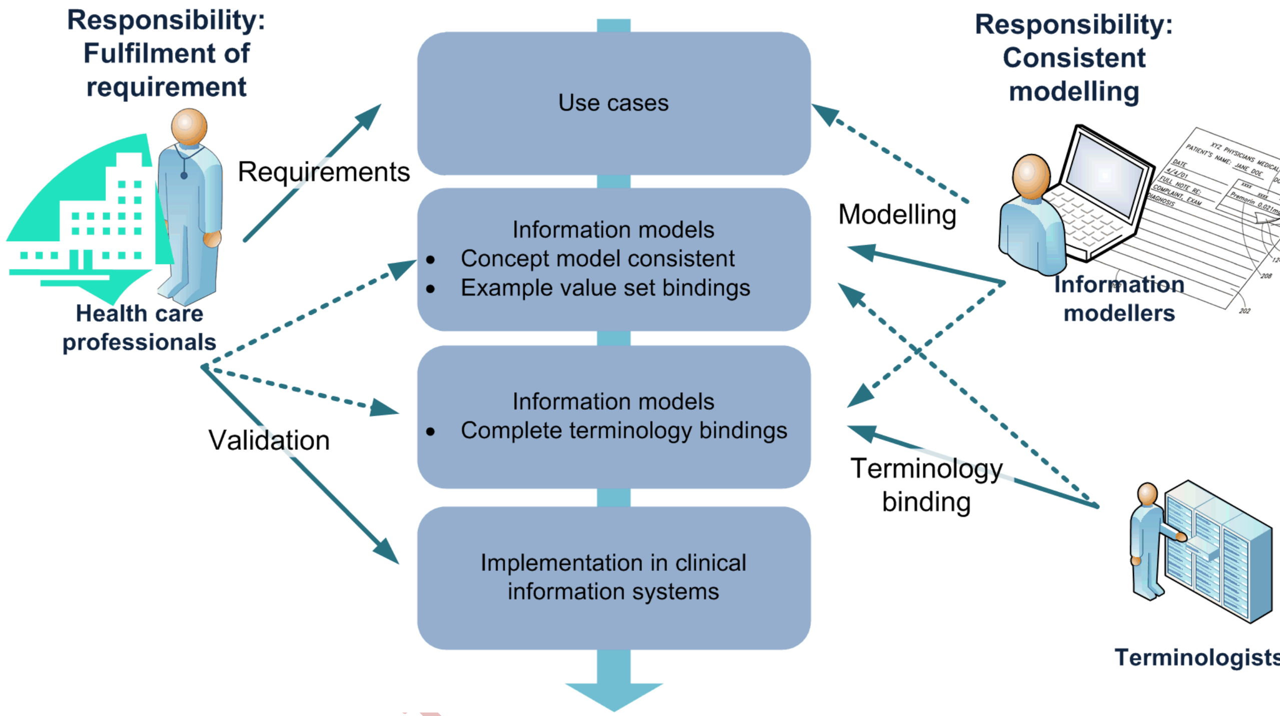

Figure 17 - Design process with focus on requirements and consistent modelling

The idea of the comprehensive design process is described in the following:

Use cases: In the first activity health care professionals would be responsible for the requirements. Others could be involved to structure or collect requirements.

Information models: Initial information models should be informed by existing information models either from the organisation in which they are build or international models or standards. If existing models are bound to terminology, identifying them will be easier as described in the section about information model library management in section 4.4. Design should also ideally be consistent with the terminology concept model as described in section 5.3 about consistent model meaning binding and expanded on in chapter 11. Model meaning bindings are important early in the process because they may influence the way information is structured in the information model. One way of achieving a consistent terminology binding is by using information model patterns which have established bindings to the SNOMED CT concept model. Some existing models or patterns may need specialisation to meet the use case. Additional, specific data elements may need to extend existing models and patterns. Both these activities require both modelling and terminology binding. Moreover, initial value set bindings can be suggested to make the purpose of the model clear. When finishing this activity validation with health care professionals may be appropriate, because the unpopulated model is now finished. If appropriate, models may be populated with appropriate valid values using value set bindings to meet health care professionals and analytics needs. Both health care professionals and terminologists are important at this level. Meeting analytics needs might require involvement of hospital management or medical societies.

Implementation: After implementation in clinical information systems, it is important to validate that an information model meets requirements. This activity may depend on how safety critical the processes affected by the information model is. For example, introducing a new physical examination template is not as safety critical as introducing a new chemotherapy dose calculation and record system. Below, the comprehensive design process is illustrated using the radiology report example.

Design of a radiology report 2:

Use cases: *Radiologists provide existing paper based report and tells about radiology workflows. Radiologists and information modellers discuss that a radiology report needs a section on which body structure should be examined, and the procedure details e.g. image modality and slide thickness. Dependent on the body structure of interest different findings may be reported. They consider whether each of these finding types should have their own structured entry or whether a general finding section should be modelled. They decide that fracture findings, nervous system findings and cardiovascular findings need a specialised section each.*

Information models: *Information modellers and terminologist studies each of the sections in the radiology report. The information modeller finds clinical models for vessel finding and heart finding in the information model library management system. The clinical models have been developed by cardiologist at an earlier stage. The terminologist suggests a concept model consistent pattern for fractures as illustrated by the hierarchy in Figure 15(b) where a fracture is described by its associated morphology, finding site and laterality.*

The existing vessel and heart models need specialisation too meet the radiology use case. In the procedure details, a procedure pattern is utilized, but slide thickness has to be added as an extension because none of the attributes in the pattern can be specialized to slide thickness. Radiologists validate the model. They believe that the nervous system finding section have been too granular so that data entry will be too time consuming. The nervous system finding section is re-designed.

A lot of the data elements in the radiology report should have an attached value set. Each instance is analysed by health care professionals and terminologists. Value sets are designed and bound to the information model. E.g. the value set for a fracture finding field is defined as all children of the SNOMED CT concept 125605044|Fracture of bone|.

Evaluation: *The radiology report information models are judges as non-safety critical compared to the existing paper forms. A heuristic evaluation [8] is conducted to ensure the usability of the user interface models.*

6.4 Maintenance of terminology bindings

Terminology binding should also be done after the design phase is over. Like all other design constructs, terminology bindings should be maintained. Maintenance of terminology bindings can be challenging because they depend on both terminology and information models. This means that there are some build-in requirements to maintenance.

With every bi-yearly release of SNOMED CT, it should be analysed whether changes affect existing terminology bindings. If the SNOMED CT components that the bindings are dependent on are retired or updated, the terminology bindings should be retired or updated accordingly. Changes to the information model will depend of the changes in the chosen modelling approach and changes to the clinical requirements. Typically, these changes do not occur on a regular basis, but they should never the less be tracked carefully. If changes are non-trival technical validation and clinical verification should be repeated.

7 Where and how should terminology binding be represented?

7.1 Overview

The terminology binding can be stored as a part of the information model, as a terminology artefact i.e. a specialized reference set or as an independent table. The choice between these options depends mostly on versioning because the version of the binding will tend to be consistent with the artefact in which it is stored. The three different storage options will be explored further in the following sections.

7.2 Representing bindings in the information model

Storing the terminology binding as part of the information model is the most common. The information model formalism will decide the precise syntax of the terminology binding.

Standardisation- and interest organisations have been implementing terminology binding specific attributes in their information models; however, most of these support value set bindings and simple model meaning bindings only. Additional cooperation between information model and terminology organisations are needed if support for e.g. intentional value sets expressed using the expression syntax language should be implemented.

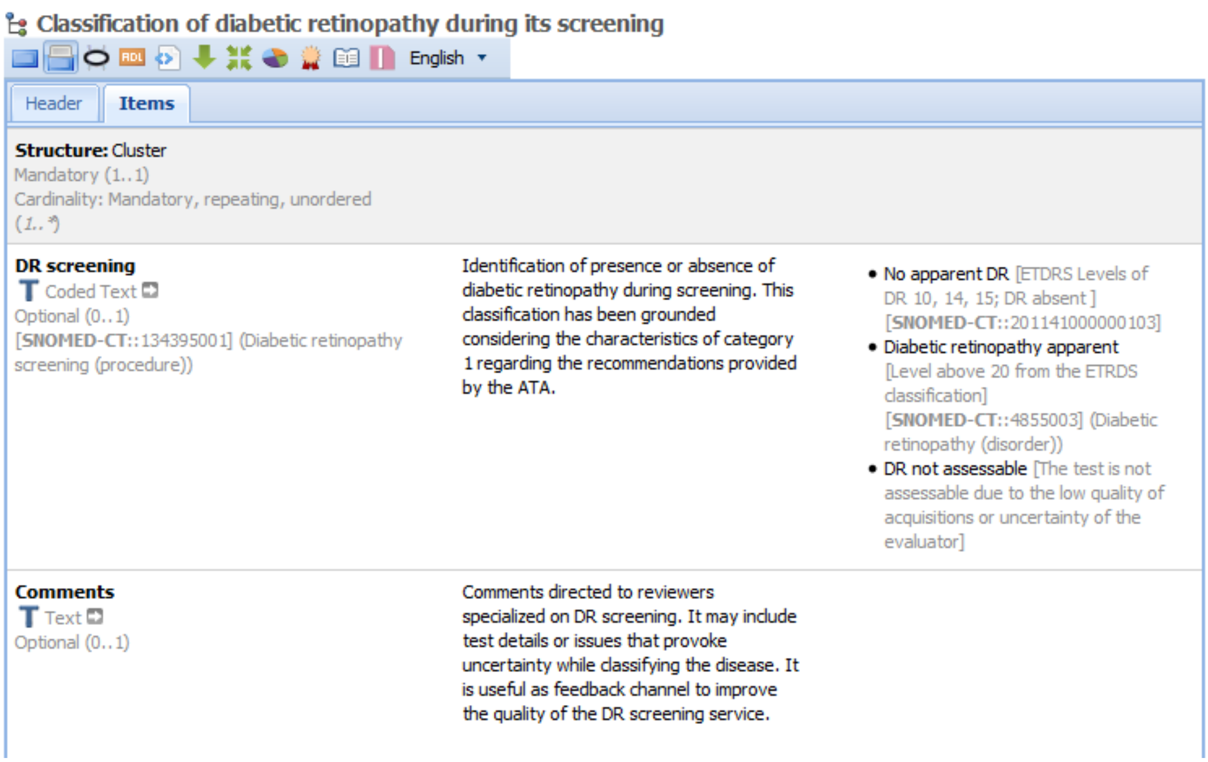

Figure 18 openEHR archetype: Diabetic retinopathy screening

One example of a terminology binding is the openEHR-EHR-CLUSTER.diabetic_retinopathy_screening_result.v1 openEHR archetype, see Figure 18 . The archetype shows that the data element 'DR screening' has been coded with 134395001|Diabetic retinopathy screening (procedure)|. The value set is defined by the model, and each node is bound to a SNOMED CT concept, each one drawn from the international edition of SNOMED CT except for' No apparent DR' which is drawn from a national extension. There is no separate version on the terminology binding, the versioning follows that of the information model, and in the retinopathy archetype case, the metadata (which cannot be seen of the figure) states that the date of authoring is 2013-04-22. Keeping the terminology bindings up to date i.e. identifying retired concepts is not an integrated part of the information model storing approach, and it should be handled by well-defined procedures locally following the bi-yearly releases.

7.3 Representing bindings in a terminology artefact

The reference set (refset) format makes it possible to customize attributes to be included. For terminology binding purposes a refset could be made with the columns "information model element", "terminology artefact " and "metadata". An example of a refset is included in Table 1. The example is related to the fracture terminology binding e.g. Figure 38. The table show the six mandatory fields of simple reference sets in green, and three additional fields related to the terminology binding. The only challenge is populating the ReferenceComponentId because it is defined as a SCTID that reference the component to include in the refset. However, it is not always a component (in the IHTSDO definition) that we want to bind to information model. Sometimes it may be a postcoordinated expression or an intentional refset. Two solutions are reasonable. One solution is to populate the referenceComponentID with the root concept namely 138875005|SNOMED CT Concept|. The second solution is to populate the field with the closest meaningful parent of the terminology artefact you are binding to e.g. the postcoordinated expression |fracture of bone|:|Finding site|=|bone structure of hand| and the expression constraint <<125605044|Fracture of bone| could both have 125605044|Fracture of bone| as referenceComponentID. This would allow using the referenceComponentID for indexing, even though it does not identify the rows uniquely. In the example, the only metadata represented is the binding type. However, the metadata could be represented by several attributes, see e.g. chapter 12 on metadata. Refsets have the benefit of being versioned like all other SNOMED CT artefacts, and terminology binding refset files can be included with other national extensions in the bi-yearly releases. Keeping the terminology bindings up to date should be an integral part of releasing the terminology binding refsets as well as all the other refsets included in national extensions. However, keeping track of the version of the information model is not an integrated part of this approach and should be handled by well-defined processes on a national or local level.

Table 1 - Fracture binding refset example

Id	Effective	Activ	moduleId	refSetI	Reference-	infoMod	Metadat	term-

	-Time	e		d	Component-Id	el	a	Artefact
someUUID	20153101	1	myModuleID	myRef-SetID	125605004	$Fracture.Fracture	Type: valueSet	<<125605004
someUUID	20153101	1	myModuleID	myRef-SetID	72704001	$Fracture.Type	Type: value set	xxx|fractureTypeRefSet|

7.4 Representing bindings in an independent table

Terminology bindings stored in an independent table enable an independent versioning of the terminology binding, which will be particular relevant in innovation projects where fast revisions of the binding itself is needed, and where it is not feasible to change the version of the information model repeatedly. The separate versioning means that neither terminology versioning nor information model versioning is an integrated part of this approach, and it should be handled carefully. An example of an independent table can be seen in Table 2, related to the fracture example. The table is comparable to Table 1, but when not building on the Refset format the following remarks should be made. The rows in the table have to be identifiable, in the example in Table 2 this is handled by the green fields Id and setId. If it is decided to make one terminology binding table only, the setID can be omitted. Else, the setID should identify each binding table uniquely. The table should also be versioned, in the example this is the purple fields, which are borrowed from the refset format. It should be considered to add the version of the information model and the terminology artefact as well, but this can also be handled implicitely by assuming the most recent versions compared to the effective time of the binding. The blue fields are the same as in Table 1.

Table 2 - Fracture binding example, independent table

Id	setID	effectiveTime	Active	infoModel	Metadata	termArtefact
someUUID	mySetID	20153101	1	$Fracture.Fracture	Type: valueSet	<<125605004
someUUID	mySetID	20153101	1	$Fracture.Type	Type: value set	xxx|fractureTypeRefSet|

8 Which information model artefacts should be bound to terminology?

8.1 Overview

For bindings to work properly, information models should be populated with information about the terminology binding. Dependent on the type of information model artefact there are restrictions on how they can bind to terminology, this is explores in the first section. In the second and third section, population information model attributes with information about the terminology to which it is bound and handling Null flavours is explored.

8.2 Information model artefacts and what they can bind to

Different data types of data elements make different bindings possible and meaningful. In Table 3, an overview of information model artefacts with different data types and their possible bindings are presented. In the table, the blue rows indicate a binding to model meaning whereas the purple rows indicate bindings to value sets. Generally, value set bindings require that the data type of the data element is <Coded text>. Some exceptions exist, se purple rows in the table e.g. the unit of a data element of type <Quantity> can be bound to a value set expressing the valid units for the quantity. Regardless of the data type, data elements can always bind to model meaning. Moreover, Data groups and models can also bind to model meaning. Model meaning bindings are illustrated in the blue rows in Table 3.

Table 3 - Information model artefacts and what they can bind to

Information model artefact	Type of binding	Purpose
Model	Simple model meaning	Simple model meaning makes it possible to express the conceptual meaning of the model as a whole. If necessary, the model context can be expressed as well to allow consistent querying. For the information model artefact" model", binding to simple model meaning, concept domains and attribute and range is the same thing.
	Compositional model meaning bindings	A compositional model meaning binding expresses the precise compositional meaning of the model by referencing data groups and data elements within the model including data element that alter the context of the model. A compositional model meaning binding will often make simple model meaning bindings unnecessary.
Data group	Concept domain model meaning binding	Expressing the conceptual meaning of the data group as a whole.
	Attribute and range model meaning binding	The attribute express how the data group is linked to the model. Range refers to a concept domain as explained above. Attribute and range can be used to de-compose meaning.

	Compositional model meaning bindings	A compositional model meaning binding expresses the precise compositional meaning of the data group by referencing subordinate data groups and data elements
Date element independent of type	Concept domain model meaning binding	Expressing the conceptual meaning of the data element.
	Attribute and range model meaning binding	The attribute express how the data element is linked to the data group. Range refers to a concept domain as explained above. Attribute and range can be used to de-compose meaning.
Date element of type Coded Text	Simple value set binding	Expressing valid values as refsets or expression constraint
	Conditional or dependent value set binding	Expressing valid values which are in some way dependent on the value of another data element within the model.
	Compositional value set binding	Expressing valid values as a composition of the values of other data elements within the model
	Concept domain model meaning binding	For coded text concept domains can be a first step in generating value sets because concept domains can be interpreted as conceptual categories of concepts
Data element of type quantity	Simple value set binding	The unit-part of the quantity data type can be bound to one or more units expressed in a standardized terminology e.g. SNOMED CT or UCOM
Data element of type ordinal	Simple value set binding	Each ordinal value may be bound to a terminological representation of that ordinal. For example the ordinal Apgar scores 1-10 might each be bound to the SNOMED CT concepts 77714001| Apgar score 0|, 43610007 |Apgar score 1| etc
Date element of type Enumerated list	Simple value set binding	Express valid values by binding each item in the list to one code in the terminology

Note that data elements of type <Coded text> should whenever possible have a value set binding associated, because this is the purpose of the data type. Terminology binding to all other information model artefacts is optional and should be use case driven.

8.3 Populating data-type fields

When a data element refers to a code system, a number of attributes have to be populated. The attributes that needs to be specified varies among modelling initiatives, but are typically including:

- Code: Which is a reference to one or more codes in a specific code system
- Code system: a system id for the code system, typically an OID or URI is used. The OID for SNOMED CT is 2.16.840.1.113883.6.96 and the URI is http://snomed.info/sct

- Code system name: A human readable specification of the code system e.g. "SNOMED CT" or "LOINC"
- Code system version: The version of the terminology, at the time where *Code* was specified.
- Term: Is typically a display term that can be used for user interfaces
- Original text: If the code is derived from a text e.g. using natural language processing techniques, the original text attribute is used to point at the original text

8.4 Effect of Null Flavour on terminology binding

In addition to the above mentioned attributes, it is often possible to specify Null Flavours of a data element. Null Flavours are used to indicate missing data. Common Null flavours are:

- "No information" that indicates nothing about the reason
- "Invalid" that indicates that the data in the instance is not a member of the value set which define the valid values
- "Unknown" that indicates that a possible value exist but is not provided
- "Masked" that indicates that the data is not provided due to privacy settings
- "Not applicable" if no valid data exist e.g. pregnancy for male patients

It is important for terminology binding to know the Null Flavours in the modelling framework because Null Flavours in some cases overlap with content that might be expressed using a terminological representation. For example, in SNOMED CT the finding context might indicate that a certain diagnosis is unknown, this could mean that an expression such as |Ulcerative colitis|:|Finding context|=|Unknown| would be saved to the database. If instead, the Null Flavours were utilized a data element |Ulcerative colitis| would save the Null Flavour "Unknown". Generally, it is important that the boundary between Null Flavours and terminology bindings is handled consistently throughout an implementation.

9 Types of terminology bindings

9.1 Overview

Two main categories of terminology bindings exist: value set bindings and model meaning bindings

- Value set bindings are used to express the valid values used to populate an information model artefact. For example the value set used to populate a drop down menu in a user interface or the valid values for a coded text in a message model.
- Model meaning bindings are used to express the meaning of the information model artefact. For example an entry-template can be named "COPD follow-up" and bound to 394702007 |Chronic obstructive pulmonary disease follow-up|.

In addition to the two super-types, various sub-types exist to be able to support different use cases. In the following, value set bindings and model meaning bindings are presented in more details.

9.2 Value set bindings

Value set bindings can be extensional i.e. a coded field bound to an explicit value set. In a SNOMED CT context these would often be expressed as reference sets. Value set bindings can also be intentional i.e. a coded field bound to an expression constraint. An expression constraint is a logical statement that expresses the valid values without mentioning them explicitly e.g. all subtypes of 64572001|disease| would be expressed as <<64572001|disease| using the expression constraint syntax (reference to language work).

Complex values set bindings are constructed by adding dependencies, conditions or a compositional structure to a value set binding.

Conditional value set bindings will have its valid values constrained by a condition expressed by a traditional if-then-else structure. E.g. **If** patient is male **then** the value set to use is the “male procedure reference set” **else if** patient is female then the value set to use is the “female procedure reference set” **else** the value set to use is “all procedure reference set”.

A dependency value set binding will have valid values that are depending on another coded text in the model. E.g. a model may have a procedure category and a procedure type. For the procedure type we may specify that it should be a descendant of whatever value was specified in the procedure category. In a user interface this would translate to first choosing the procedure type from a list such as [delivery procedure, management procedure, surgical procedure]. If surgical procedure was bound to 387713003 |surgical procedure| and the user picks this in the list, then the procedure type would be <<387713003 |surgical procedure| and the user could e.g. pick 80146002 |appendectomy|. Dependency value set bindings does not make sense in an extensional context.

In a compositional value set binding, the value set is defined by a composition that defines the valid pre- and post coordinated expressions. One example is that the coded text “fracture finding” can be bound to a composition that states that valid values are descendants of 125605004|fracture of bone| that can be refined by specifying the finding site as a descendant of 272673000|bone structure|. This can be expressed using expression template syntax so that the terminology artefact is expressed as <<125605004|fracture of bone|:363698007|finding site|=<< 272673000|bone structure|.

The different value set bindings and their properties are summarized in Figure 19.

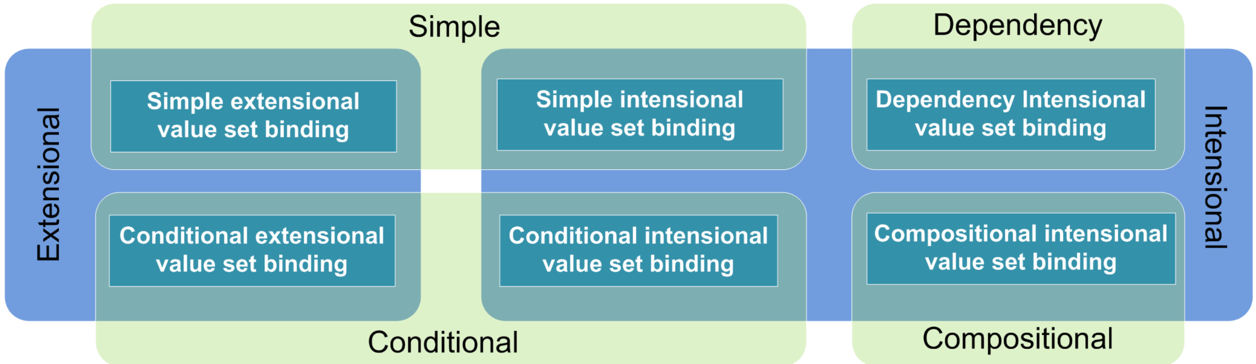

Figure 19 - Types of value set bindings

9.3 Model meaning binding

Model meaning bindings are used to express the meaning of the information model artefact at three different levels: model level, data-group level and data-element level:

- Simple model meaning bindings express the meaning of the model and identifies its context
- Compositional model meaning bindings express the composed meaning of a data group in the information model
- Attribute and range model meaning bindings express the meaning of the data element and its relationship to the data group or model
- Concept domain model meaning bindings express the meaning of a model, data group or data element by specifying a conceptual category of concepts.

The simple model meaning bindings link the entire information model to a single meaning. E.g. the observable entity 364075005|heart rate| can be used to define the meaning of a heart rate model. However, the meaning can be context dependent, therefore a simple model meaning can also contain a context wrapper that express the context or point to the context dependent parts of the model.

Compositional model meaning bindings link a data group to its valid compositions, which are expressed using expression template syntax much like the compositional value set bindings. E.g. a vital sign model can have a data group called heart rate. The heart rate data group can have its meaning defined by 364075005|heart rate|: 116680003|is a|=$observation code, 260686004|method|=$method , which means that the heart rate group has its meaning defined by its two subsequent data elements observation code and method, which can be composed together as specified by the expression template.

Concept domains can be interpreted as the type of information which can be expected in the model, data group or data element, whether or not it is a coded element and has an attached value set binding. For example, a simple physical examination model may be designed as a structured narrative i.e. each of the data elements eyes, ears, nose, throat etc. is populated with a simple text. Assigning a concept domain to each of the data elements i.e. 118235002|eye/vision finding|, 118236001|ear and auditory finding|, 118237005|nose finding| and 301186004 | upper respiratory tract finding| will allow for identification of model meaning for information management and retrieval purposes. For example, retrieving all earlier 297268004 |Ear, nose and throat finding (finding)| for a patient would allow extraction of the text entered in the ears, nose and throat data elements by inferring over model meaning.

In attribute-and-range model meaning bindings the “attribute” is an attribute linking the data element to the data group or in case of no data group the model itself. The “range” is a concept domain much like those of the simple model meanings. Other solutions could be chosen, but it is beyond discussion that consistent representation of non-coded elements is a prerequisite for semantic interoperability. An example of an attribute-and-range binding is that a fracture model has the range 125605004|fracture of bone| (no attribute because there is no super-level in the model). In the fracture model there is a data element called site. Site is bound to the attribute 363698007|finding site| and the range 272673000|bone structure|, because |finding site| links 272673000|bone structure| meaningfully to the model meaning i.e. 125605004|fracture of bone|.

Attribute-and-range and compositional model meaning bindings can be combined to form semantic patterns. Semantic patterns express the meaning of a model in such detail that equivalent meanings registered in different models can be identified and used as if they were native for all models. Compositional model meaning bindings are used to compose meaning whereas object and range model meaning bindings are used to de-compose meaning. Semantic patterns or ontology patterns are a key factor in achieving semantic interoperability and has been studied e.g. by the semantic health net [1]. The model meaning binding types are summarized in Figure 20.

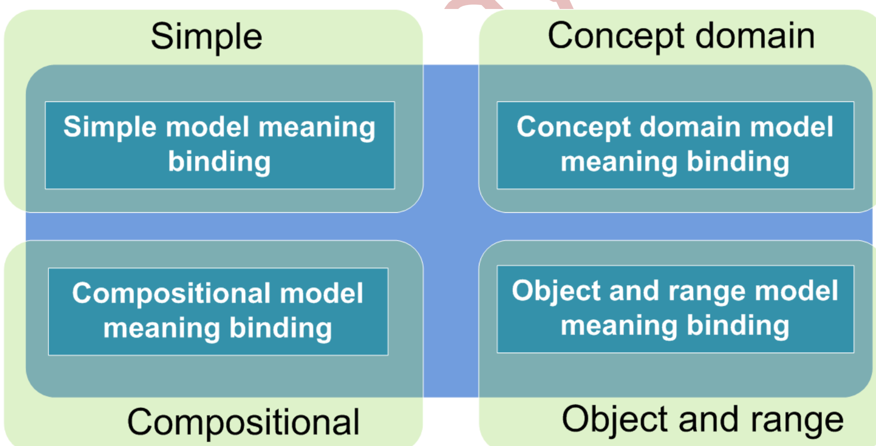

Figure 20 – Types of model meaning bindings

9.4 Linking use cases and binding types

9.4.1 Overview

Each of the healthcare system activities described in Chapter 3 have different terminology binding requirements. In this section, we explore the terminology binding functions that are needed to support each of these activities and consequently the binding type and approach that is most suitable to meet these needs.

The functions that value set bindings can perform include:

- Express valid values;
- Express valid values based on conditions or dependencies; and
- Define how coded values may be composed.

The functions that model meaning bindings can perform include:

- Express concept domains;
- Define meaning of each coded element;
- Define meaning of each data group;

In the following sections we match the activities described in Chapter 3 with these functions and binding types.

9.4.2 Data capture

As shown in Figure 21, the data capture activities described in Section 4.2 primarily require value set bindings.

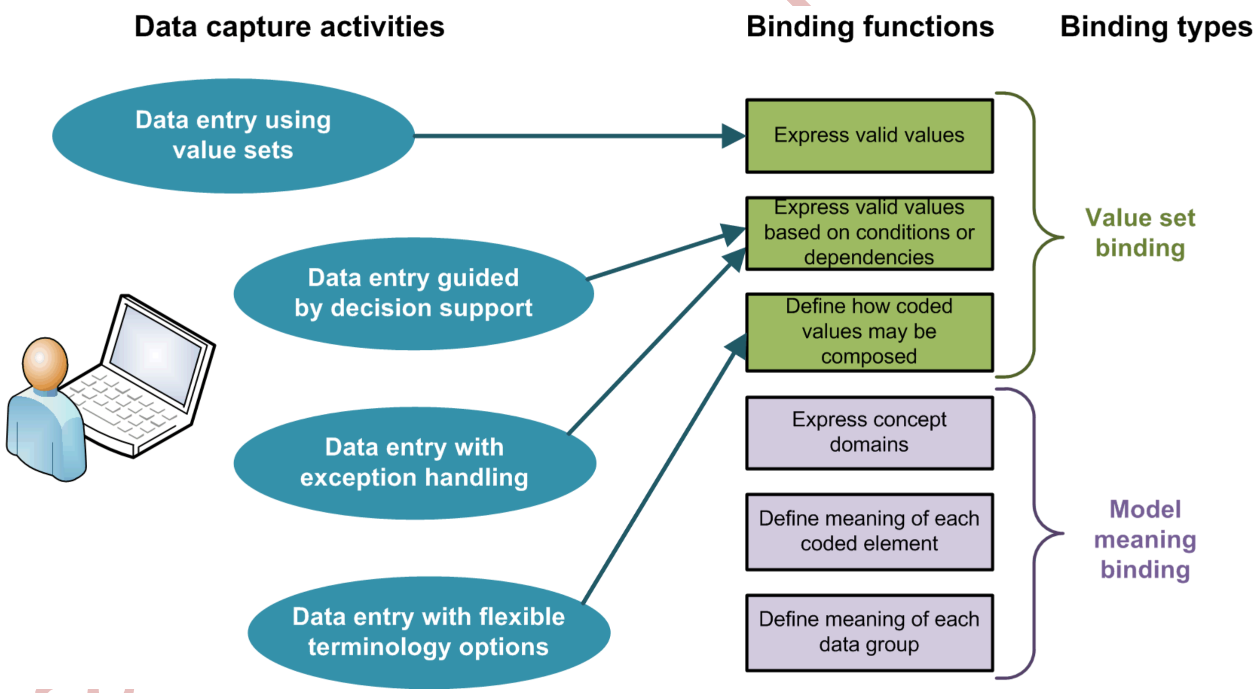

Figure 21 – Data capture activities with binding functions and types

Table 4 provides a reference to the sections of this document that describe the terminology binding approaches needed to support each data capture activity. This table also describes the level of difficulty of the binding – not for solving the use case as a whole.

Activity	Binding Solution	Document Reference

Data entry using value sets	Simple value set binding	Section 10.2
Data entry using clinical decision support	Conditional and dependent value set binding	Section 10.3 and 10.4
Data entry handling exceptions	Conditional and dependent value set binding	Section 10.3 and 10.4
Data entry combining pre- and post-coordinated expressions for flexibility	Compositional value set binding	Chapter 10.5

Table 4 - Overview of use case and solutions for data capture

9.4.3 Retrieval and querying

As shown in Figure 22, the retrieval and analytics activities described in Section 4.3 require both model meaning and value set bindings.

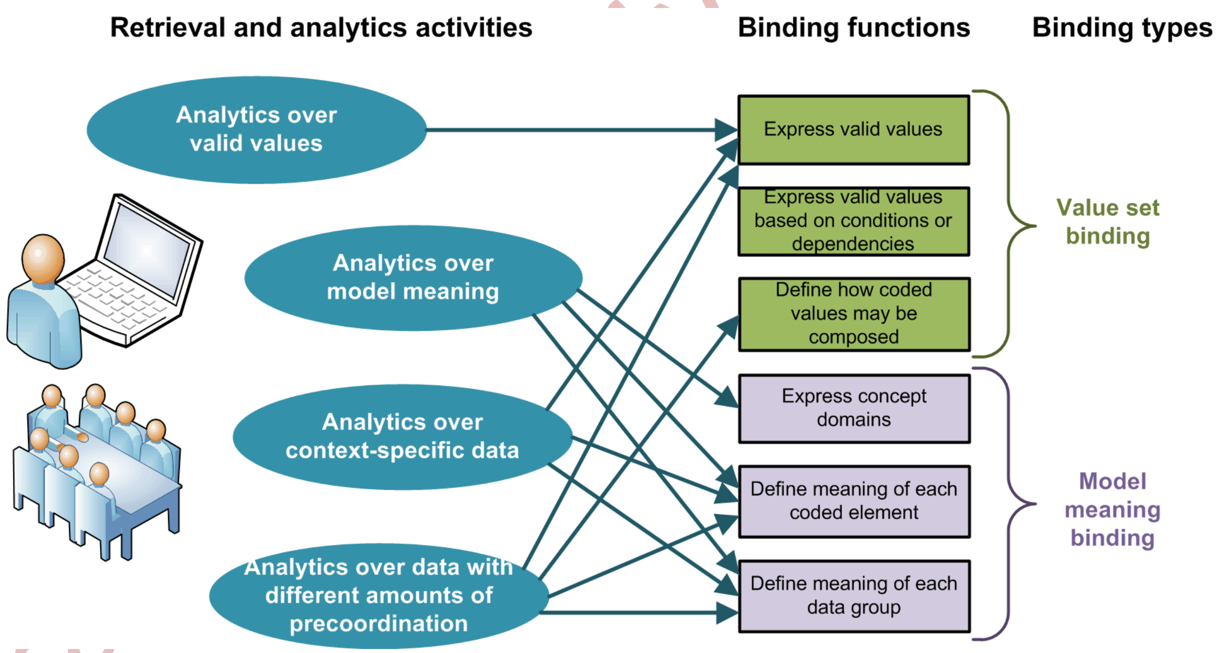

Figure 22 – Retrieval and analytics activities with binding functions and types

It should be noted that simple cases of analytics over model meaning can be based on simple model meaning bindings only. Compositional and Object-Range bindings are just another option giving other possibilities. Table 5 specifies this further and provides a reference to the sections of this document that describe the terminology binding approaches needed to support each analytics activity. This table also describes the level of difficulty of the binding – not for solving the use case as a whole.

Activity	Binding Solution	Document Reference
Analytics over valid values	Simple value set binding	Section 10.2
Analytics over model meaning	Simple model meaning binding OR compositional model meaning bindings OR Object-Range model meaning bindings OR a combination.	Section 10.2, 11.4 and 11.5
Analytics over context specific data	Simple value set binding AND compositional model meaning bindings AND Object-Range model meaning bindings	Section 10.2, 11.4 and 11.5
Analytics over data with different amount of precoordination	Simple value set binding AND compositional model meaning bindings AND Object-Range model meaning bindings	Section 10.2, 11.4 and 11.5

Table 5 - Overview of use case and solutions for retrieval and analytics

9.4.4 Information model library management

As shown in Figure 23, the information model library management activities, described in Section 4.4, primarily require expression of concept domains using simple model meaning bindings. However, if the more complex Object-Range model meaning bindings are available, they can be used instead.

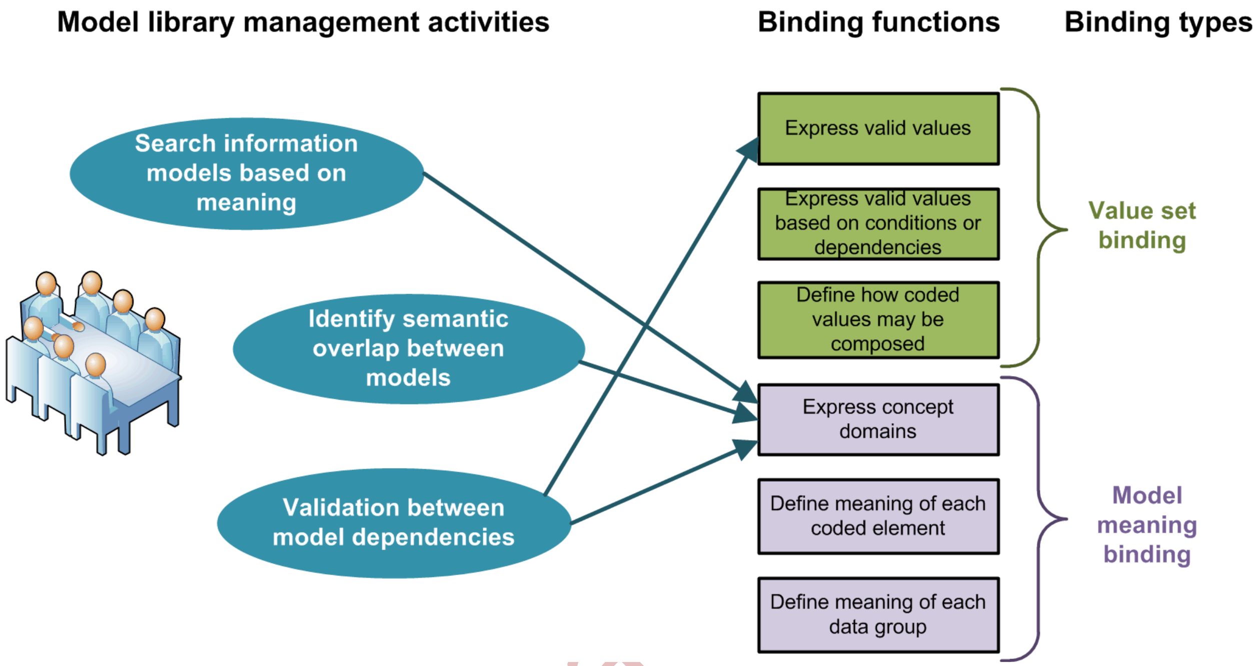

Figure 23 – Information model library management activities with binding functions and types

Table 4 provides a reference to the sections of this document that describe the terminology binding approaches needed to support each data capture activity. This table also describes the level of difficulty of the binding – not for solving the use case as a whole.

Activity	Binding Solution	Document Reference
Search information models based on meaning	Simple model meaning bindings	Section 11.2
Identify semantic overlap between models	Concept domain model meaning bindings	Section 11.3
Validation between model dependencies	Simple value set bindings and Concept domain model meaning bindings	Section 10.2 and 11.3

Table 6 - Overview of use case and solutions for model library management

9.4.5 Semantic interoperability

As shown in Figure 24, the semantic interoperability activities, described in Section 4.5, require extensive use of both value set and model meaning bindings.

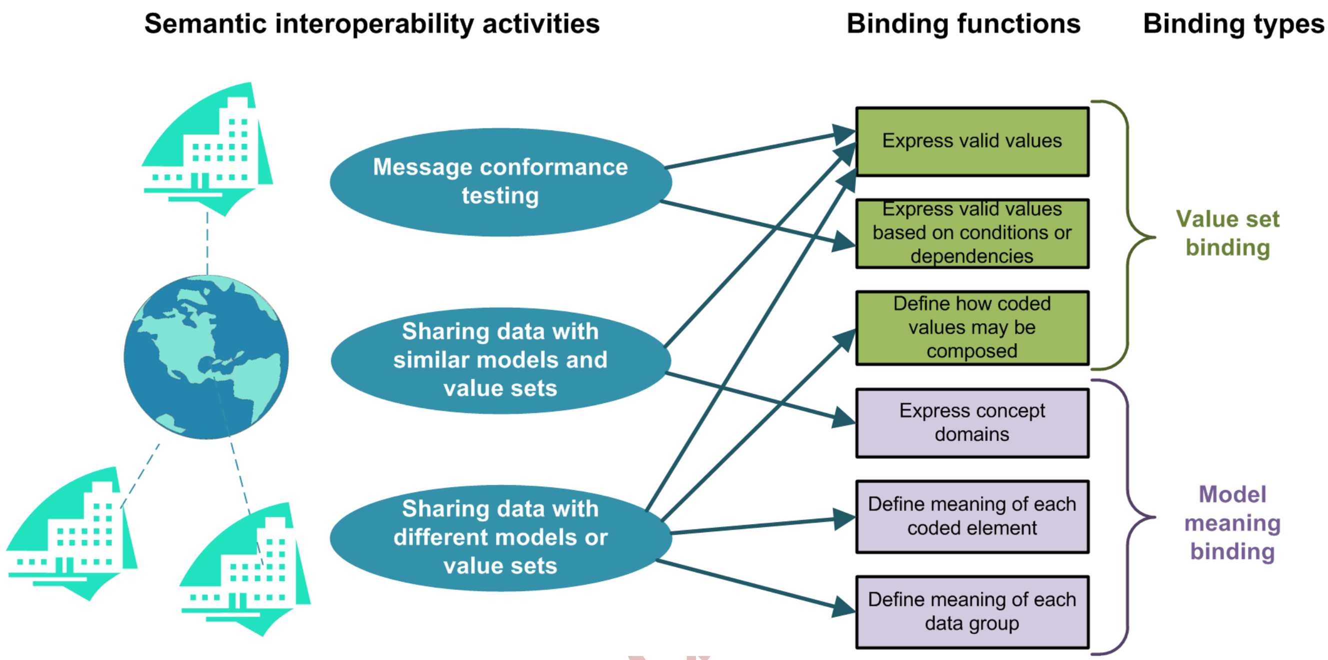

Figure 24 – Semantic interoperability activities with binding functions and types

Table 7 specifies this further and provides a reference to the sections of this document that describe the terminology binding approaches needed to support each analytics activity. This table also describes the level of difficulty of the binding – not for solving the use case as a whole.

Activity	Binding Solution	Document Reference
Message conformance testing	Simple value set binding AND/OR conditional value set bindings AND/OR dependent value set bindings	Section 10.2, 10.3 and 10.4
Sharing data with similar models and value sets	Simple value set binding AND Concept domain model meaning binding	Section 10.2 and 11.3
Sharing data with different models or value sets	Simple value set binding AND Compositional value set bindings OR compositional model meaning bindings AND Object-Range model meaning bindings	Section 10.2, 10.5 ,11.4 and 11.5

Table 7 - Overview of use case and solutions for semantic interoperability

10 Value set binding

10.1 Overview

Value set binding is the corner stone of terminology binding. The purpose of simple value set bindings is to express the valid values for a data element. This means that value set bindings are important for:

- Data capture. Simple value set bindings can simplify data capture e.g. by specifying the content of drop down list. Simple value set bindings can also improve the consistency and quality of data for reuse.
- Querying. Simple value set bindings provide can a common interpretation of meaning for analytics. However, handling post coordinated expressions especially expressions with explicit context often require the use of compositional value set bindings as well.
- Information model library management. Simple value set bindings is one element in validation of between model dependencies (Together with simple and concept domain model meaning bindings).
- Semantic interoperability. Simple value set bindings are one element in supporting message conformance testing (Together with the other value set bindings), and in enabling data sharing between models with different representations of the same content (together with complex model meaning bindings).

Variations of value set binding exists that adds functionality:
Both conditional and dependent value set bindings constrain the values of one data element based on the values of another, where at least one of these data elements is coded. The purpose of these binding types is to make decision support available at data entry and to allow for exception handling. The bindings also are also useful in message conformance testing.

The purpose of compositional value set bindings is to express how the values recorded in two or more fields may be combined to represent a composite meaning. The compositional binding helps to compose data values together into concept-model correct forms, which is relevant when transforming data from one model to another.

10.2 Simple value set bindings

Simple value set bindings can bind an information model artefact to a terminology artefact in several different ways.

- The valid values can be expressed in a subset, which in a SNOMED CT context is represented using a reference set. The binding is then between a data element in the information model and the reference set. This is an extensional binding.
- The valid values can be expressed in the information model as an enumeration type, and each data value can be bound to a concept or a post coordinated expression. This can be seen as an extensional binding; however, in a strict sense, it could be regarded as a model meaning binding because the terminology represents something which is already defined by the model. In this guide data values bound to concepts or expressions are considered extensional value set bindings because they can be used interchangeably with data elements bound to reference sets despite their different representations.
- The valid values can be expressed using an expression constraint. The binding is then between a data element in the information model and the expression constraint. These binding are called intentional.

Each of these is presented in the following sections.

10.2.1 Binding data elements to reference sets

Figure 25 shows a simple example of a value set binding, where the coded text data elements of the Observation model are associated with reference sets in the terminology. For example, the Observation code data element is linked to the |observation reference set|, and the reason data element is linked to the |reason reference set|.
These reference sets may either be intentionally or extensionally defined within the terminology, but are referenced in the binding using the reference set identifier. The last binding associates the 'units' attribute inside the Quantity data type of the Result data element with a subset of SNOMED CT units defined in a reference set. In this example, we have used 3 'x's before the name of the reference set to show that an identifier needs to be used.

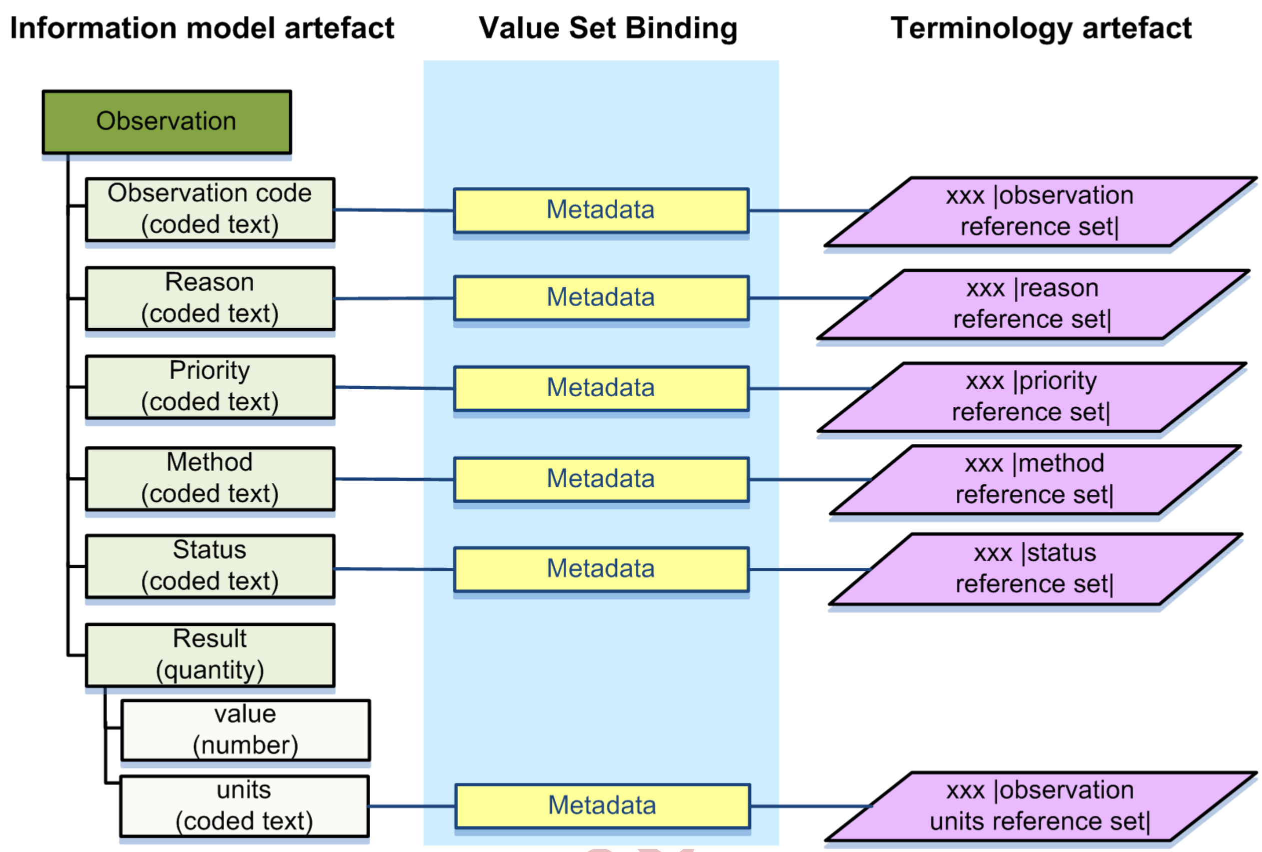

Figure 25 - Simple value set bindings from data elements to reference sets

The terminology binding approach should also allow bindings to other code systems because terminology bindings required in an information model could come from other terminologies or classification systems. For example, in the Observation model example above, we could refer to the whole of LOINC as a value set, to indicate that the values used to populate the 'Observation code' must come from LOINC. An URI can be used as a reference to the terminology artefact e.g. http://loinc.org. This could be specified further e.g. by binding the 'Observation code' to a subset of LOINC that is defined by the 'observation-code' value set in the FHIR specification using a URI such as http://hl7.org/fhir/vs#115|observation-codes|.

10.2.2 Binding data values in an enumeration to concepts or expressions

Enumeration types are value sets explicated in the information model. Most information models allow enumerated types, and they can be useful for small sets of terms, where it is not relevant to make a reference set or expression constraint. It can be discussed whether a binding to an enumeration type is actually a model meaning binding, because the terminology is expressing the meaning of entities in the model i.e. each item. However, the binding of concepts or expressions to items in an enumeration serves the same purpose as the value set bindings i.e. expressing valid values. Consequently, binding of terminology to enumerations have been defines as a value set binding for the purpose of this guide.

In Figure 26, an enumeration type is used to express the valid values for care needs and each data value is bound to a SNOMED CT concept or a SNOMED CT expression, related to which type of care the patient needs.

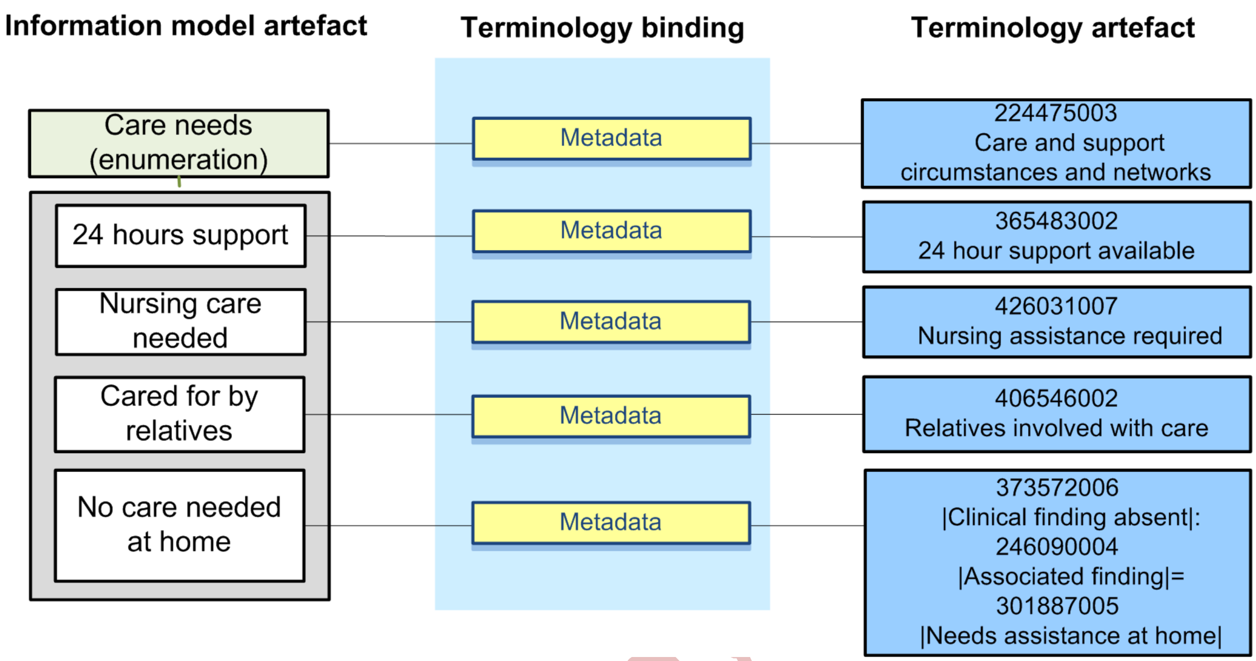

Figure 26 – Example of binding an enumeration type to concepts and expressions

10.2.3 Binding data elements to expression constraints

Intentionally defined value set bindings can are defined using an expression constraint. In Figure 27, the bindings for the Observation model are expressed using expression constraints. For example, the interpretation of the first binding is that the values of the 'Observation Code' are descendants of |observable entity|, and the interpretation of the second binding is that the values of 'Reason' are either descendants of |clinical finding| or |procedure|. In the last binding, we show an expression constraint which express that the units of the Observation Result must be members of the |units reference set|. This is just a different way of referring to a value set, which uses an expression constraint syntax.

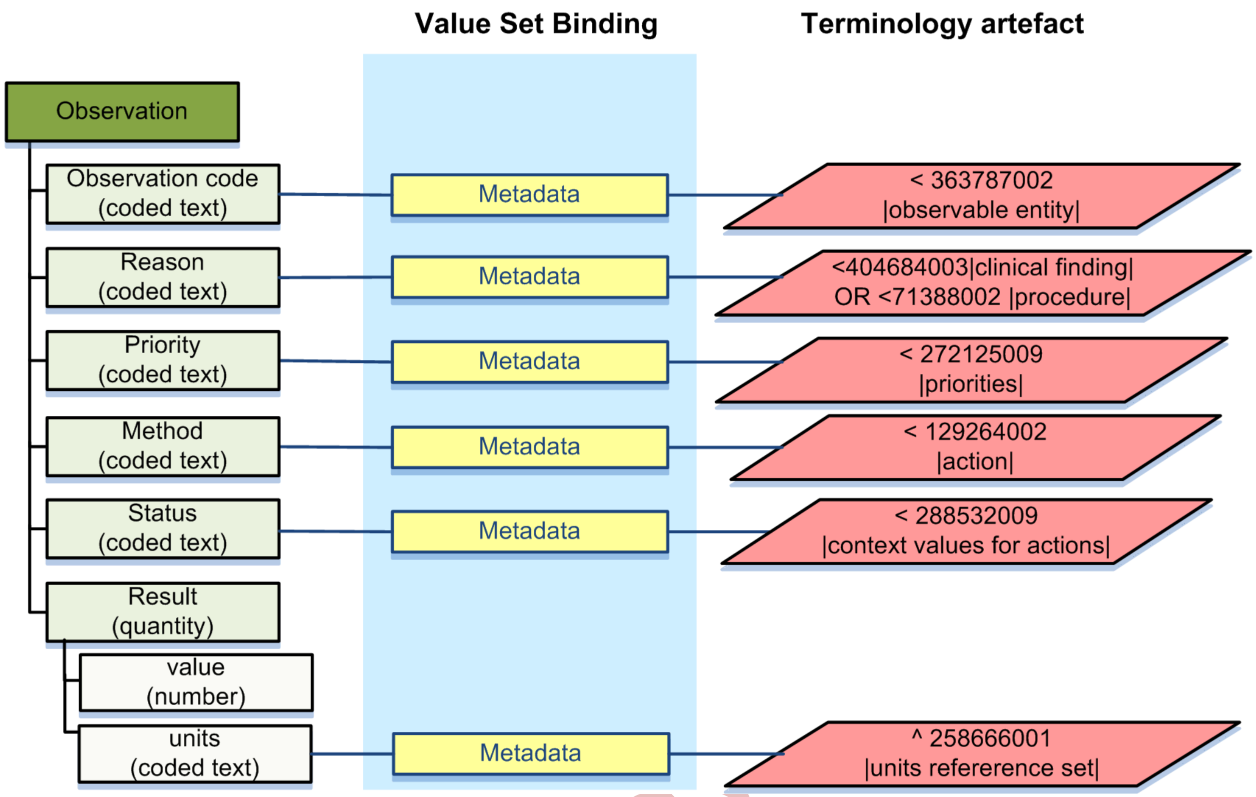

Figure 27 - Example of simple value set binding where data elements are bound to expression constraints

10.3 Conditional value set binding

Conditional value set bindings binds a data element to its valid values using a conditional statement. The condition is based on the value of another data element. The conditional value set bindings can be both extensional and intentional i.e. valid values can be expressed using either reference sets or expression constraints. Representation of conditional statements do not have an agreed upon language.
In Figure 28, a Discharge Summary model is shown, in which the value set used to populate the 'Procedure' data element is dependent on the sex of the patient. In this case, if the patient is male, then the valid values of procedure are expressed in the 'Male procedure reference set', which includes some procedures that are only applicable to males. If the patient is female, then the valid values of procedure are expressed in the 'Female procedure reference set'. If it is not known whether the patient is male or female, or they are neither, then the valid values of procedure are expressed in 'All procedures reference set'. At the data entry level this binding ensures that male procedures are not entered for female patients and vice versa. In messaging, it can be automatically detected by a sending or receiving system if a message conforms to this rule. In Figure 28, the conditional statement is situated in the terminology binding lane because it is considered a part of the binding itself.

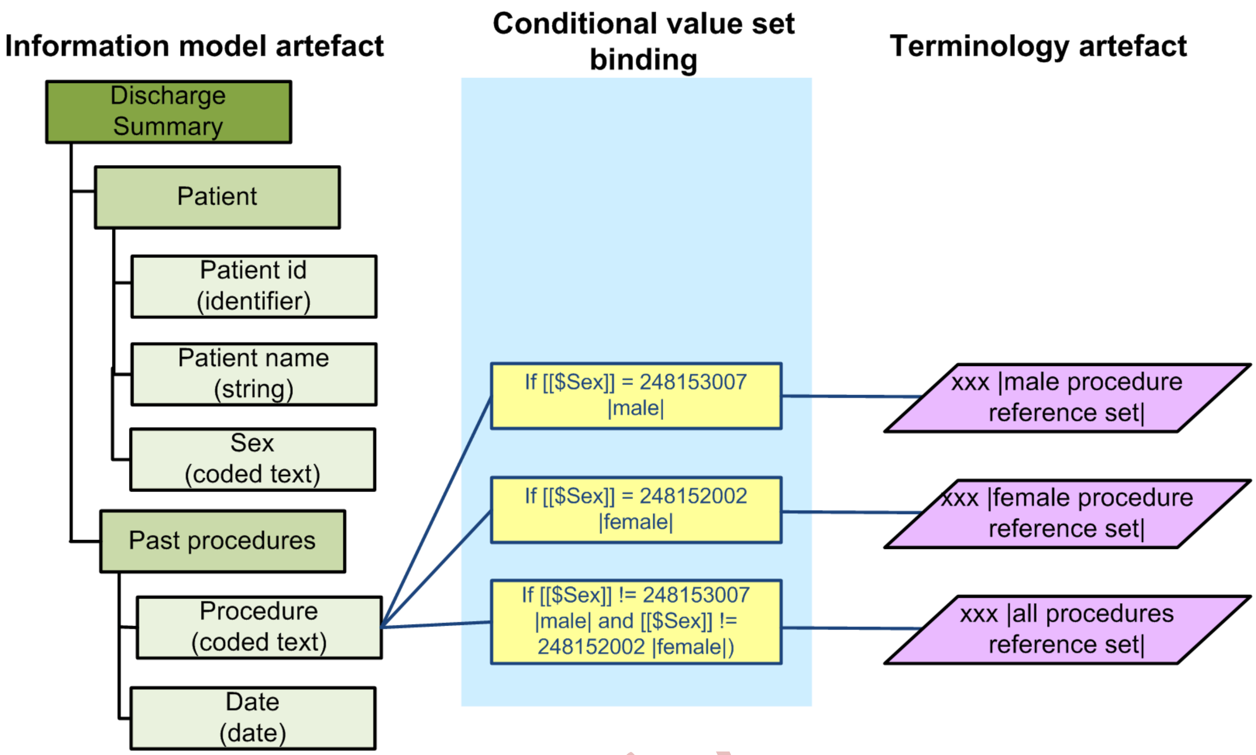

Figure 28 - Example of conditional value set binding

Conditional value set binding can be based partially on other data types than coded text. For example, in a Body temperature model, the constraint on the temperature value could depend on the units of measure. If the units are 'degrees Celsius', then the value must be between 20 and 50, and if the units are 'degrees Fahrenheit' then the value must be between 68 and 122.

10.4 Dependent value set binding

A dependent value set binding binds a data element to its valid values dependent on a value entered in another data element. Both data elements should be of type coded text. Figure 29 a dependent value set binding is shown, which takes its point of departure in a Discharge Summary model. The purpose of the binding is to define a constraint on the 'Procedure Type' data element to say that its value must be a descendant of the value of Procedure Category. So if, for example, the Procedure Category is 'Surgical Procedure', then the Procedure Type must be a descendant of 'Surgical Procedure' - for example, 'Appendectomy'.

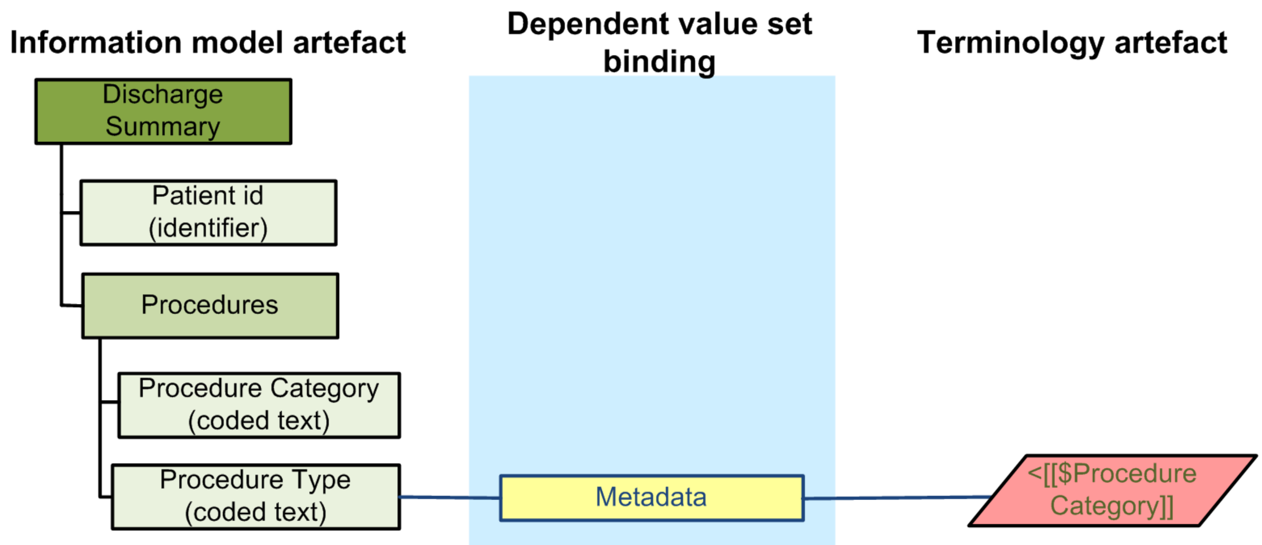

Figure 29 - Example of a dependent value set binding

In the above example, we express that an IS-A relationship defines the link between the procedure category and the procedure type. However, other relationship types can be used in dependent value set bindings. For example in a medication model the link between two data elements prescribed medication and dispensed medication could be is 'is pack of'. In this case, the dependency constraint attached to dispensed medication would be expressed as |is pack of|=[[$Prescribed Medication]]. In other words, the medication dispensed must be a valid packaging of the medication that was prescribed[1].
To express dependency value set bindings, an expression constraint template language is needed because a reference back to the information model is a prerequisite.

[1] |is pack of| is not currently a valid SNOMED CT attribute in the international version.

10.5 Compositional Value Set binding

In Figure 30, a compositional value set binding of an observable model is illustrated. The terminology artefact is an expression template that references the information model data elements and structures the referred values in accordance with the concept model. The expression template could, for example, be used to transform the data from one model, which uses 6 separate data elements, into another which combines the Reason, Priority, Method and Status together with the Observation Code into a single value.

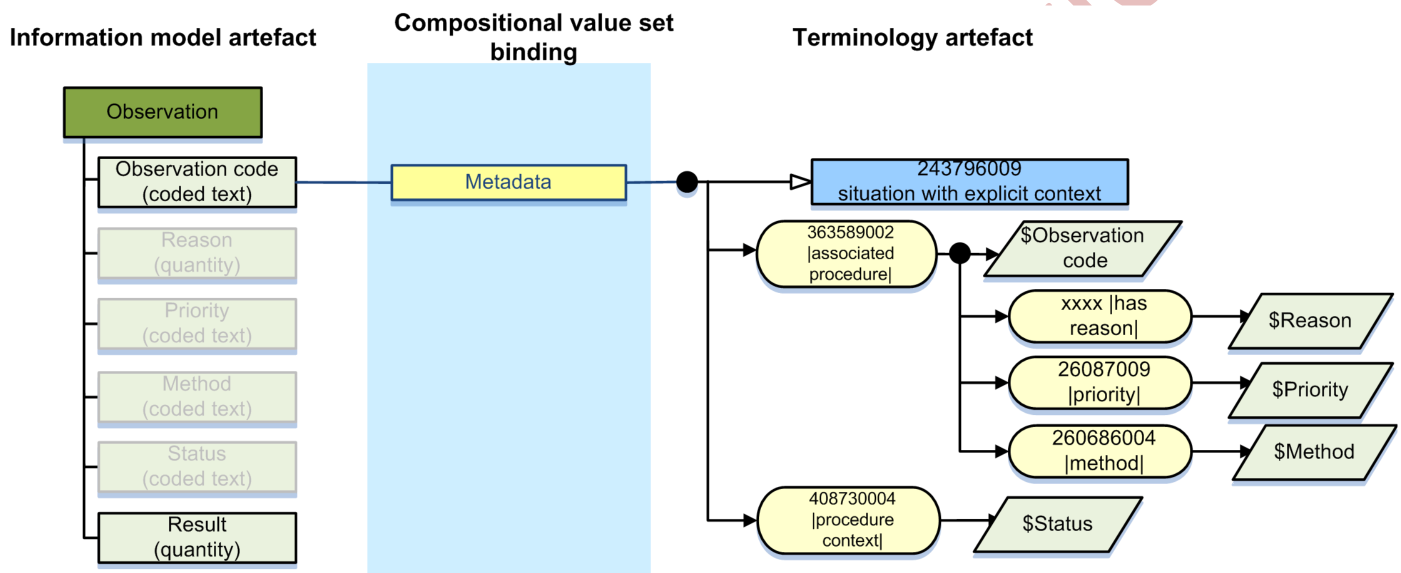

Figure 30 - Example of a compositional value set binding

The concept model consistent expression template may be difficult to develop because of the limitations in the SNOMED CT content model attribute hierarchy compared to the attributes needed to combine all the data elements that may be structures together in information models, in Figure 30 this is shown by the attribute value xxx|has reason|, which is not in the SNOMED CT concept model. When developing expression templates it is most important to think about the use case because some data may always be obtained in an information model structure, and consequently this data element may be transformed between models without representing it in an expression. In Figure 30, the result of the observation is represented in the information model without being part of the expression template. See chapter ?? about how to represent data with information models or terminology respectively.

It is important to handle context and negation when composing an expression template. Context and negation is important to handle if any of the attributes have the ability to alter the focus of the recorded meaning. In Figure 30, the focus concept (in the blue box) is NOT 363787002|observable entity| because the status attribute have the ability to alter the meaning of the expression. The default context is 385658003|done|, but if the status is set to e.g. 397943006|planned| or 385644000|requested|, the focus is no longer a 363787002|observable entity|, because that implies that something has actually been observed. Instead it is expressed as 2437960009|procedure with explicit context|.

11 Model meaning binding

11.1 Overview

Model meaning bindings express the meaning of information model artefacts whereas value set bindings are used to express valid values. This means that whenever the meaning of the model is of specific interest there will be use cases for model meaning bindings. Use cases are within the fields of querying and retrieval, information model library management and semantic interoperability:

- *Querying.* All model meaning bindings can be used to query a record, it is important to note that where the value set bindings will make it possible to add conditions to a query e.g. procedures which are appendectomies. Model meaning bindings will make it possible to identify all the procedure data elements in a record where this condition is relevant. Model meaning bindings, as they can be attached to other data elements than coded text, can also make it possible to retrieve all information of a certain type e.g. all data that are registered in models where the heart is in focus. This could (dependent on EHR design) return the result of cardiac auscultations, cardiac ultra sound results, MR images of the heart, ECGs etc, if each of these is tagged with a model meaning related to the heart.
- *Information model library management*. Simple model meaning bindings and concept domain model meaning bindings identify the semantic content of a model, and is therefore relevant in library model management. The model meaning binding make it possible to search model libraries e.g. which observation models or physical examination models do we have? Or which models include heart rate related findings? The model meaning bindings make it possible to identify semantic overlap between models, which can be used to identify when models cover the same area but with variations that would be better to avoid within a certain organisational setting. Validation between model dependencies is another use case that model meaning bindings support e.g. if we want to make sure that our heart rate model is a true specialisation of our observable model.
- *Semantic interoperability.* The combination of compositional model meaning bindings and object and range bindings is a way of expressing semantic patterns that makes it possible to compose and decompose meanings. These semantic patterns are important when we want to represent equivalent meanings in different models which are a prerequisite for semantic interoperability.

There are several types of model meaning bindings as described in section 9.3. These are exemplified in the following sections.

11.2 Simple model meaning binding

Simple model meaning bindings are used to express the meaning of the model and when nessesary identify its context. In Figure 31, a heart rate model is illustrated which is bound to the observable entity 364075005|heart rate| i.e. one concept that approximate the meaning of the entire model. By binding all models in a model library to their simple meaning it will e.g. be possible to search for groups of models e.g. all cardiovascular observation models by searching for simple model meaning bindings that are << 364066008|cardiovascular observable|.

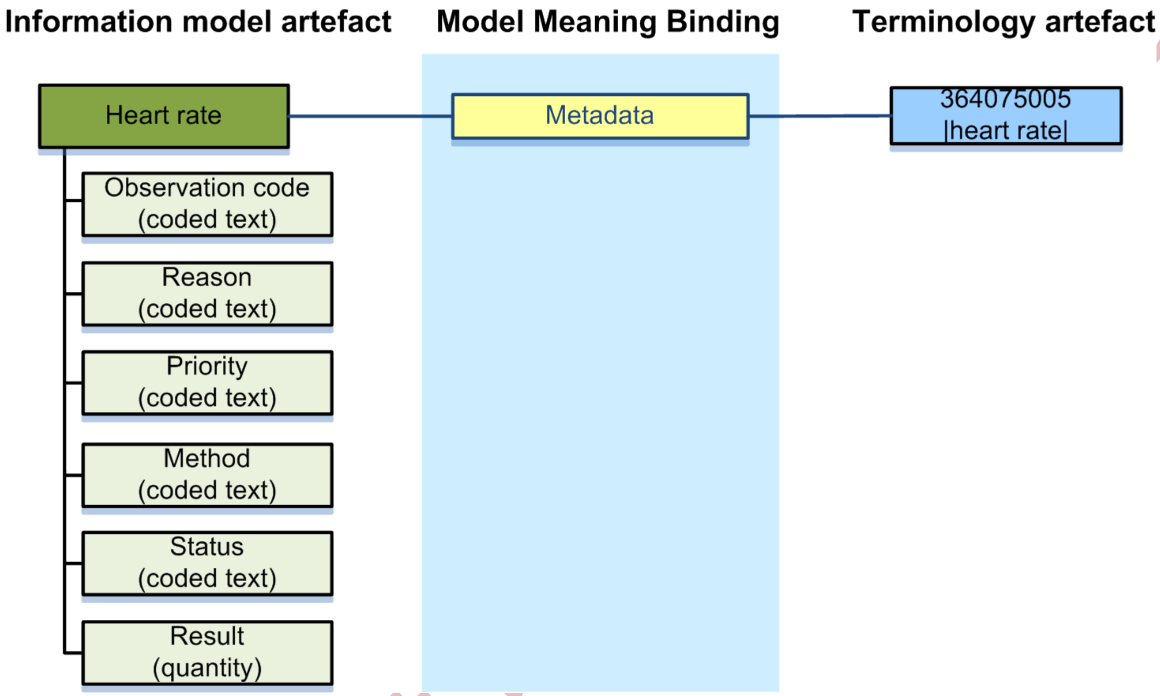

Figure 31 –Example of simple model meaning binding without context

In medical documentation, one of the challenges is to keep track on context that fundamentally alters the meaning of specific findings, observations or procedures e.g. a specification that a procedure is planned rather than performed or that a finding is absent rather than present. Simple model meaning bindings with context can support that by using a context wrapper that points to the context dependent parts of your model.

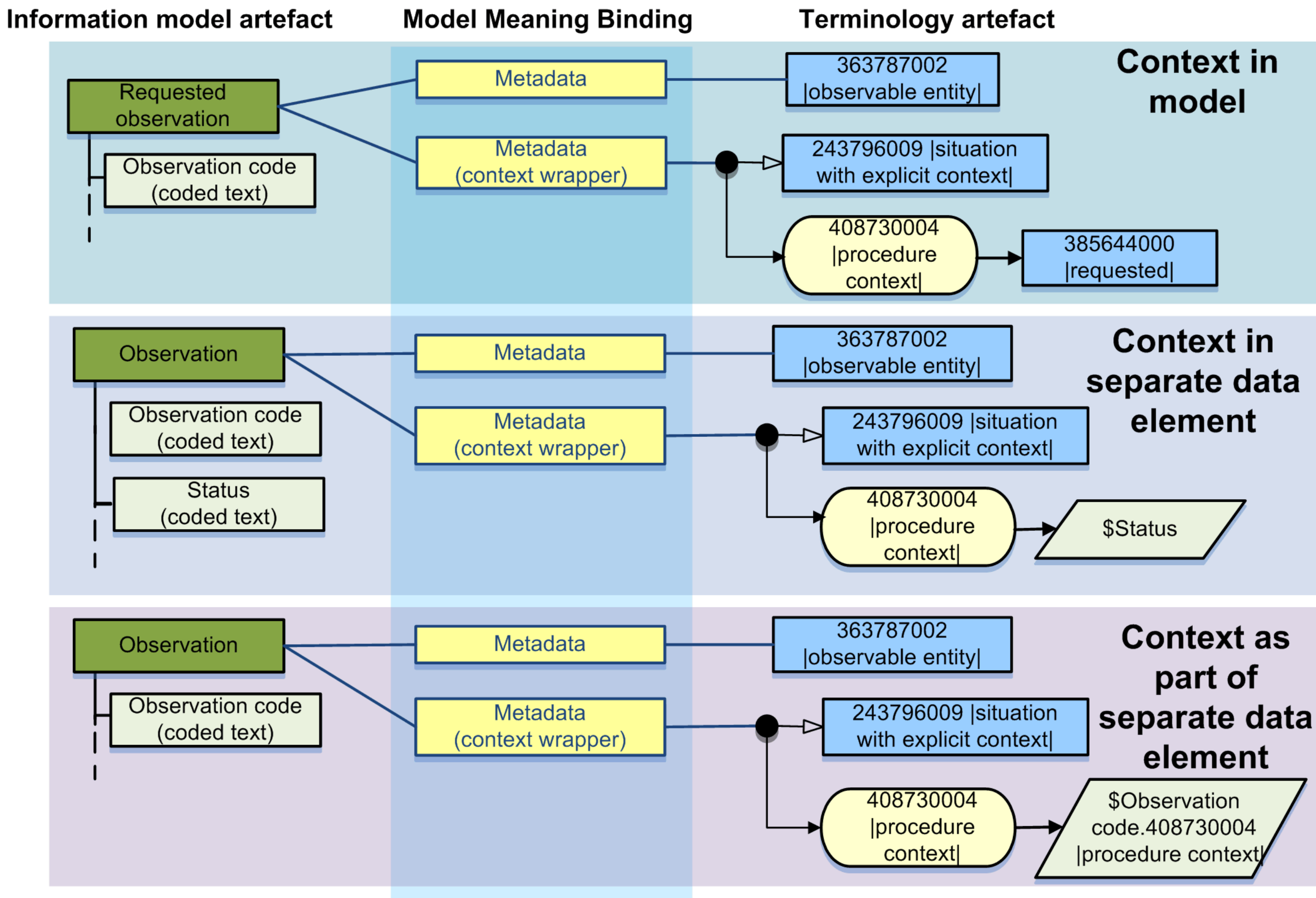

Figure 32 - Examples of simple model meaning bindings with context

As illustrated in Figure 32, there are three places where context can be represented. In the first example, the context is part of the model i.e. the whole model is a request. In the second example, context is specified in a separate data element i.e. the context wrapper point to the data element where the context is specified. In the third example, content is specified as the context part of another SNOMED CT encoded data element i.e. the context wrapper points to the 408730004|procedure context| of the observation code. Expressing context consistently support correct identification of content of interest, e.g. if we want to identify all requested observation, we can first identify all observations and then go to the context wrapper to see whether "requested" is specified as model context or context in a data element.

11.3 Concept domain model meaning binding

Concept domain model meaning bindings conceptually express the meaning of a model, data group or data element. In Figure 33, Concept domains are specified for the observation model and its data elements. The concept domain for the model itself is equivalent with the simple model meaning binding. Concept domains are helpful for various query purposes. They are also beneficial for model management purposes when trying to identify semantic overlap or inconsistencies between models e.g.

an organisation could test whether their heart rate model was a true specialisation of their observation model by testing if each data element in the heart rate model subsumed a data elements in the observation model. Concept domains can also be seen as a first step in generating value sets because concept domains can be interpreted as conceptual categories of concepts.

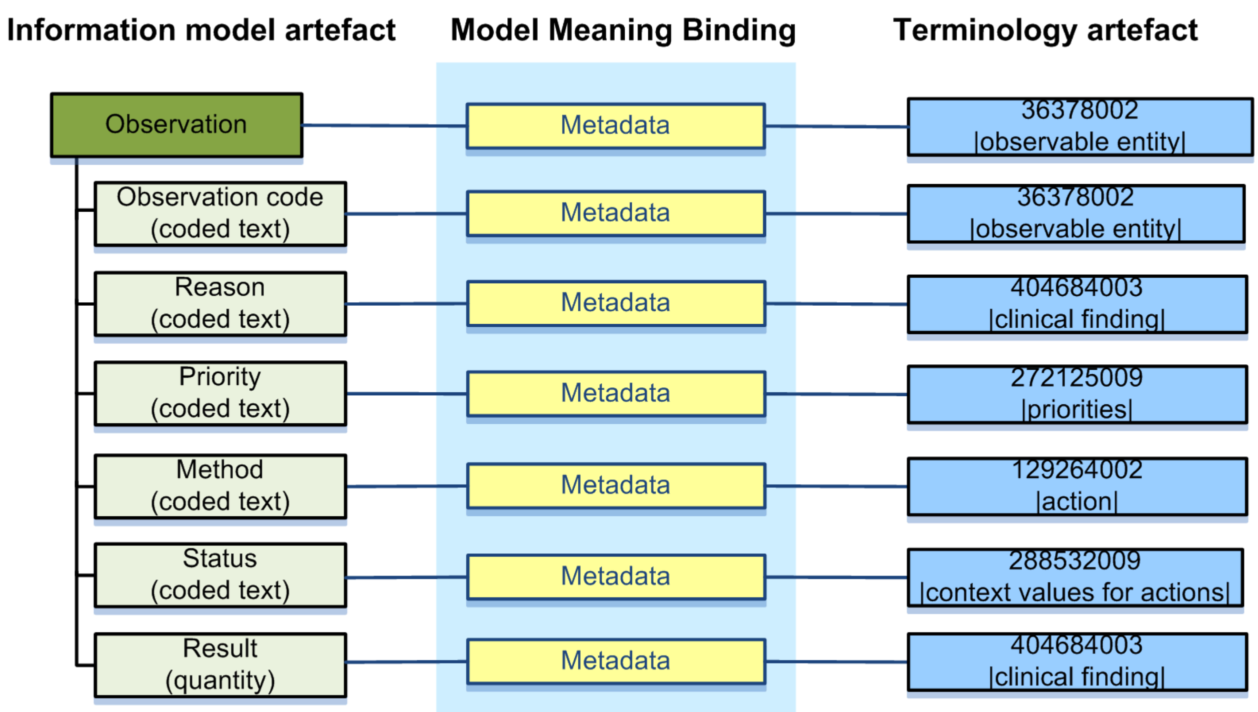

Figure 33 - Example of concept domain model meaning bindings

11.4 Compositional model meaning binding

A compositional model meaning binding expresses the precise compositional meaning of a data group or a model. In Figure 34, the meaning of the observation model is expressed as a composite meaning of its data elements. In this example the composition is in a close-to-user form which requires transformation to be concept model valid.

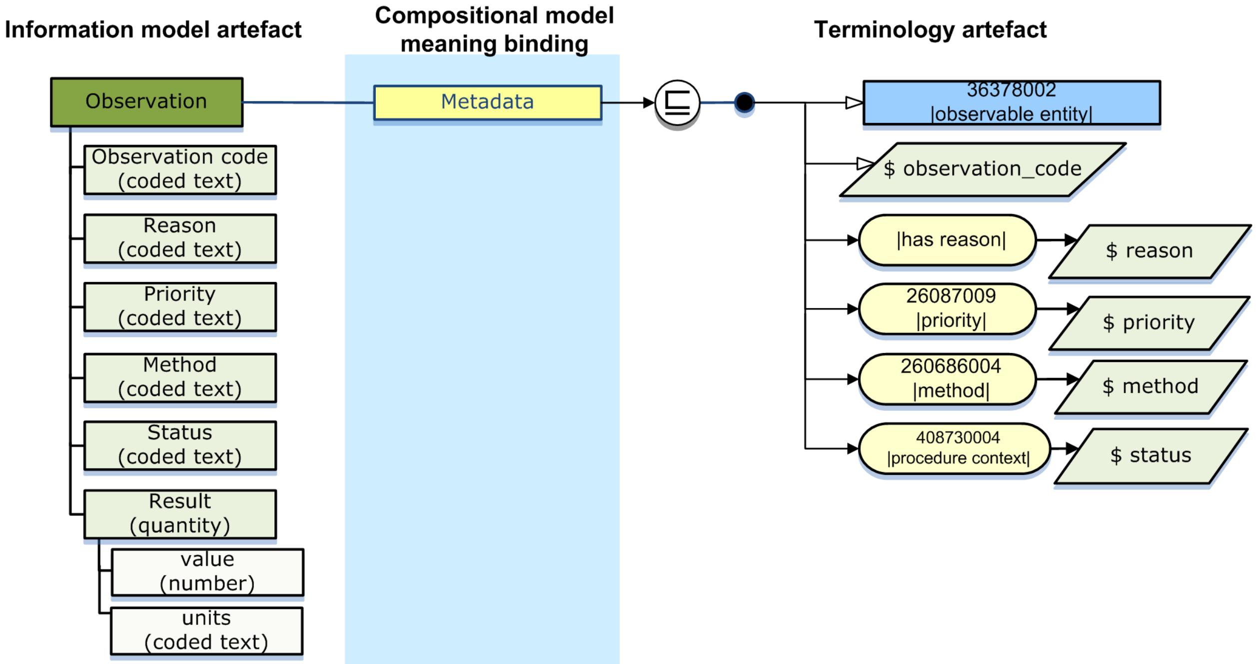

Figure 34 - Example of compositional model meaning binding

When comparing to simple model meaning bindings with context it is clear that compositional model meaning bindings are a more sophisticated way of expressing model meaning as well as context using a single expression. In Figure 35, compositional model meaning and simple model meaning bindings as a way to express context is compared.

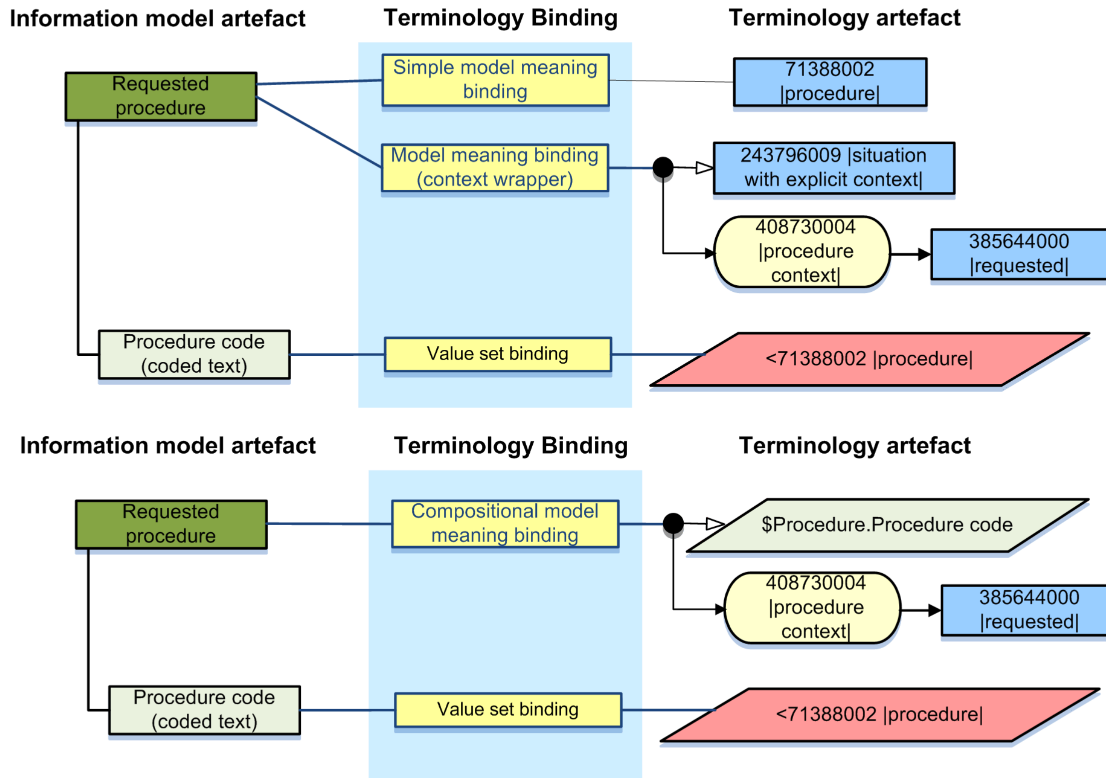

Figure 35 - Simple model meaning bindings vs. compositional model meaning bindings as a mean to express context

11.5 Attribute and range model meaning binding

The purpose of attribute and range model meaning bindings are to express the meaning of a data element and its link to the nearest data group or model.

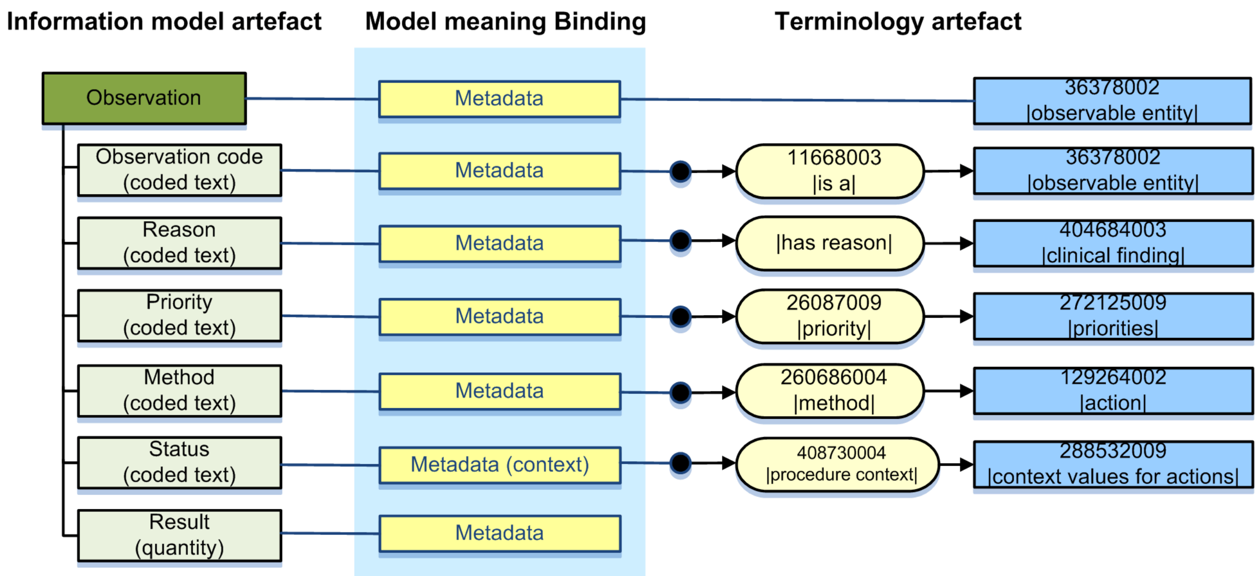

Figure 36 - Example of attribute and range model meaning bindings

In Figure 36, each data element has a range (blue boxes) that express the category from which valid values may be drawn. The attributes (yellow boxes) link each data element to the meaning of the model. Consequently, the bindings should be interpreted as follows: The data element $Method can contain types of 129264002|action|and these actions should be understood as a 260686004|method|of the 36378002|observable entity| that defines the meaning of the model $Observation. If the model had included the data group level – the attributes would link to the nearest data group.

12 Metadata

12.1 Overview

For every terminology binding, there is a set of metadata associated with it, which helps to define how the binding should be interpreted. In many cases, this metadata is not explicitly defined, but instead implicitly understood from the use case in which the binding was created. It is, however, important to be aware of what these types of metadata are, and what the default interpretation of this metadata is when it is not explicitly stated. If there is a risk that the default metadata values are not consistently understood or the values differ from the default, then the metadata should be explicitly defined with the binding. In the following sections metadata for terminology bindings will be presented.

12.2 Metadata for all types of terminology bindings

The metadata associated with all terminology bindings is used to exactly from which code system the codes originate including version and where and when the binding applies.

Code system:

- *Versioned edition:* The versioned edition is a bundle of SNOMED CT modules defined through the most dependent module and the version. This specifies the exact set of codes that are necessary to be able to interpret the binding. Consequently, the versioned edition is what a system using the binding needs to have access to. . By default, the versioned edition will usually include SNOMED CT plus all extensions maintained in the realm over which the defining organisation operates, or assumed based on the context of the information model. The versioned edition may either be associated with each individual binding, or there may be a global statement or assumption for all

Binding scope and version:

- *Binding realm or scope of the binding.* This defines the realm or region in which the binding applies. For example, a binding may be applicable internationally, nationally, or to a specific hospital or healthcare region. In general, the default realm is defined by the realm or region over which the defining organisation operates. If this is not clear from the organisation publishing the binding, then the realm may need to be specified.
- *Terminology binding version.* Versioning is usually dependent on where the binding is stored. If the binding lives with the information model, then it tends to be versioned together with the information model; similarly, if the binding lives with the terminology, then it tends to be versioned with that. Alternatively if the binding is stored independently of both, then it can use its own versioning. See more about storing and versioning in Chapter 0.

In addition, the type of the terminology binding can be described for all terminology bindings, but the "Type" metadata is populated differently dependent on whether it is a value set binding or a model meaning binding. Therefore, they are described in the following sections.

12.3 Metadata for value set bindings

The metadata associated with value set bindings is related to the type of binding, the degree of freedom in the binding, how the binding works together with the information model or information system, and additional information about the code system.

Type:

- *Type:* This defines the type of value set binding. A value set binding can have the following types: "simple, extensional", "simple, intentional", "simple, enumeration item", "conditional, extensional", "conditional, intentional", "dependent", "compositional" see chapter 10 for details. Note that a value set can only be bound to a data element of type <coded text> or <enumeration> see chapter 0.

Degrees of freedom are:

- *The degree of flexibility allowed by the value set binding.* Possible values are 'extensible', 'specialisation only',' not extensible' and 'example'. 'Extensible' means that the value set may be extended with additional values. 'Specialisation only' means that only specialisations of the existing values are allowed. 'Not extensible' means that the value set used is fixed to a specific version, which may not be extended. 'Example' is used when the value set merely provides an example to demonstrate the kind of values that may be possible. Value set bindings tend to be treated as 'not extensible' when the binding are between a data value in an enumeration type and a concept. On the other hand, value set bindings are typically 'extensible', when the binding is between a data element and a reference set or an expression constraint.
- *'Literal' or 'semantic' bindings.* In a 'literal' binding, the values used to populate the data element must be exactly those included in the value set. Most implementation models use 'literal' bindings, as this is required for the validation of data entry, storage and message exchange. Many models of meaning, particularly national or international clinical models, use 'semantic' bindings, where the values used to populate the data element may be any code that is equivalent in meaning to the ones included in the binding. In many cases a 'semantic' binding implies that as long as there is a mapping from the locally-used codes to the common value set, then that is acceptable and valid.
- *'Precoodination only' or 'Post coordination allowed'.* This metadata defines whether only precoordinated concepts may be used as values in a data element, or whether post coordinated expression which match the binding constraint are also acceptable.

Information model/information system:

- *Population of the Coded Text value pattern.* This defines how each attribute in the Coded Text datatype (e.g. code, term, codeSystem etc) is populated with values. Most specifications leave this to the implementer to decide exactly how this is done, but if automated population or validation is required, this needs to be explicitly defined. For example, the concept id is used to populate 'code', the SNOMED OID is used to populate 'code system', and perhaps the Great Britain Preferred Term is used to populate the 'term' attribute.
- *'Fixed', 'default' or 'assumed' values*. A fixed value is the value that is always used to populate the given data element. A default value is the value used to populate the data element initially, until the user decides whether or not the value needs to be changed. An assumed value is the value that is assumed intended whenever the data element is left blank. In other words, default values are relevant for data entry, whereas assumed values are relevant for data retrieval or interpretation.

Code system details for bindings using expression constraints:

- *The substrate of an expression constraint used to bind the terminology to the information model.* The substrate is the specific set of concepts that the expression constraint needs to function as intended. Should only be specified if the *versioned edition* metadata does not give adequate information or if the *versioned edition* is not populated.

In the context of this guide, specifying whether or not coding of a given data element is mandatory or not is a feature of the data element itself, rather than the binding, and can be specified by using a datatype flavour (or constraint) like 'CNE, for Coded No Exceptions, and CWE, for Coded With Exceptions.

12.4 Metadata for model meaning bindings

Metadata for model meaning bindings can describe the type and the degrees of freedom:

Type:

- *Type:* This defines the type of model meaning binding. A model meaning binding can have the following types: "model meaning, simple", "context wrapper", "concept domain", "attribute", "compositional". Attribute and range model meaning bindings are specified by combining a "concept domain" and an "attribute", see chapter 11 for additional details.

Degrees of freedom:

- *The degree of flexibility allowed by the model meaning binding.* Possible values are "explicit" and "example". 'Explicit' is used when the model meaning binding is exact and expressed in the only possible code system. 'Example' is used when the model meaning binding merely provides an example to demonstrate a code that can possible define the model meaning.

13 Principles

13.1 Overview

Throughout this document, a diversity of possible approaches to terminology binding has been presented. The focus has been firstly to explain, and secondly to guide. In this section, the guiding principles are collected, to provide an easy-to-use check list to use when designing terminology binding. The principles are grouped so that they cover:

Principles aimed at ensuring correlation between use cases and terminology binding
Because terminology binding should be purposeful
Principles aimed at designing terminology bindings in accordance with SNOMED CT
Because the terminology has to influence information modelling to get a meaning based representation of clinical information
Principles aimed at ensuring good modelling practice for terminology bindings
Because a good modelling approach is key for validation and retrieval functions to work as intended
Principles aimed at resolving issues in the boundary between information model and terminology
Because the overlap between information model artefacts and terminology artefacts have to be handled consistently
Principles guiding how to practically bind terminology to information model artefacts
Because terminology bindings should be represented correctly together with all the information necessary to interpret the binding
Principles for quality assurance
Because terminology bindings should be validated and up to date

13.2 Principles aimed at ensuring correlation between use cases and terminology binding

1. **The intended purpose of the terminology binding should drive the type of binding used.** The purpose of terminology bindings can be related to data capture, analytics, information model library management and semantic interoperability. (Details in chapter 4 and section 9.4)

2. **Binding types supporting data entry activities**:
    - *Data entry using value sets*: Simple value set binding (Section 10.2)
    - *Data entry using clinical decision support:* Conditional and dependent value set binding (Section 10.3 and 10.4)
    - *Data entry handling exceptions:* Conditional and dependent value set binding (Section 10.3 and 10.4)

  - *Data entry combining pre- and post-coordinated expressions for flexibility:* Compositional value set binding (Section 10.5)

3. **Binding types supporting analytics activities**:
    - *Analytics over valid values:* Simple value set binding Section 10.2
    - *Analytics over model meaning:* Simple model meaning binding OR compositional model meaning bindings OR Object-Range model meaning bindings OR a combination (Section 10.2, 11.4 and 11.5)
    - *Analytics over context specific data:* Simple value set binding AND compositional model meaning bindings AND Object-Range model meaning bindings(Section 10.2, 11.4 and 11.5)
    - *Analytics over data with different amount of pre coordination*: Simple value set binding AND compositional model meaning bindings AND Object-Range model meaning bindings (Section 10.2, 11.4 and 11.5)

4. **Binding types supporting information model library management activities**
    - *Search information models based on meaning:* Simple model meaning bindings (Section 11.2)
    - *Identify semantic overlap between models:* Simple model meaning bindings (Section 11.3)
    - *Validation between model dependencies:* Simple value set bindings AND Simple model meaning bindings (Section 10.2 and 11.3)

5. **Binding types supporting semantic interoperability activities**
    - *Message conformance testing:* Simple value set binding AND/OR conditional value set bindings AND/OR dependent value set bindings (Section 10.2, 10.3 and 10.4)
    - *Sharing data with similar models and value sets:* Simple value set binding AND concept domain model meaning binding (Section 10.2 and 11.3)
    - *Sharing data with different models or value sets:* Simple value set binding (Section 10) AND Compositional value set bindings (Section 10.5) OR compositional model meaning bindings AND Object-Range model meaning bindings (Section 10.2, 10.5 ,11.4 and 11.5)

13.3 Principles aimed at designing terminology bindings in accordance with SNOMED CT

1. **When creating value sets for binding, it is usually good practice to choose concepts from a common root node** (Section 5.2)

2. **The value set bindings should be consistent with the semantic bindings** e.g. be descendants of a SNOMED CT domain, or in the valid range of a SNOMED CT attribute as defined by the concept model (Section 5.2)

3. **Semantic bindings should be consistent with the SNOMED CT concept model** wherever possible (Section 5.3)

4. **There must be collaborative development in order for effective implementation of SNOMED CT in information models.** Siloed design processes cannot accommodate the need for accordance between an information model and concept model (Chapter 6).

5. **Make design choices that preserves meaning**
    - Terminology binding should wherever possible avoid coded values with the meaning equivalent to 'Other'. Instead, the information model should use a sufficiently general code, which captures all meanings not included in the list.(Section 3.4)
    - Information models should wherever possible avoid the use of Boolean attributes, and instead use terminology values to elucidate the meaning of the positive and negative. (Section 3.4)

13.4 Principles aimed at ensuring good modelling practice for terminology bindings

1. **Within a model, it should be explicit where to find attribute information – in the information model or in the terminology.**
    - Over specification should be avoided e.g. including attribute information in a code and having a data element that repeats the attribute information. (Section 3.2)
    - Under specification should be avoided e.g. when the information model artefact presumes that some attribute is specified within the terminology, and it is not. (Section 3.2)

2. **Follow the generally accepted definitions of specialization and generalization, also for the terminology artefacts**
    - The value set associated with an attribute of a specialized model must be subsumed by the value set of the matching attribute in the more general model (Chapter 18)
    - The semantic meaning of an attribute of a specialized model should be subsumed by the meaning of the matching attribute in the more general model (Chapter 18)

13.5 Principles aimed at resolving issues in the boundary between information model and terminology

1. **It is important for the author of the terminology binding to have both an understanding of the information model and an understanding of SNOMED CT** to ensure that bindings are consistent

with both the information model and SNOMED CT's hierarchy and concept model. (Chapter 3 and 5)

2. **Be aware of when to use terminology artefacts and when to use information model artefacts to represent clinical information**
    - Standardised representation of clinical domain knowledge such as diseases, symptoms, procedures is preferable to express in terminology artefacts. (Chapter 3)
    - Standardised representation of data elements such as dates, times and quantities is preferable to express in information model artefacts. (Chapter 3)
    - Standardised representations of most of the information in clinical information systems can be expressed either as information model artefacts or as terminology artefacts. The information types that can be expressed in either information model or information model artefact are (Chapter 3):
        - Standardised representation of semantic constraints such as adding a laterality to a finding
        - Context such as the absence or presence of a finding.
        - Relationship between record entries

3. **Make consistent design choices when information types can be expressed in either information model or information model artefact**
    - Within an organisation, Ensure that context is represented clearly and consistently to ensure that information retrieval can be both accurate and reliable. Section 4.3 especially 4.3.3.
    - Within an organization, information models should be consistent in terms of the amount of information model structure versus terminology precoordination is used. It is noted that this may not be possible where the information model closely mirrors the clinical user interface, where clinical workflow is the primary consideration.(Section 4.2)
    - In a cross-organizational setting, semantically equivalent models may occur as a result of different design choices. Consistent interpretation of such semantically equivalent models across organizational boundaries requires that the information pattern is expressed in the model as well as in a concept model valid composition (section 3.3, section 4.5.3 and section 9.4.5).

13.6 Principles guiding how to practically bind terminology to information model artefacts

1. **Data elements of type <Coded text> should have a value set binding associated with it, whenever possible. This is because coding is the purpose of the data type. (Chapter 0)**

2. **Terminology binding to all other information model artefacts is optional. (Chapter 0)**

3. **Terminology bindings should be represented consistently, using a standardized approach when possible.**
    - Use SNOMED URIs or expression constraints language to provide the link to a terminology artefact (Chapter 7)
    - Decide where to store the terminology binding i.e. information model, the terminology, or in an independent file and do not mix approaches (Chapter 7)
    - Terminology bindings should be clearly versioned. This may be done in the information model, the terminology, or in an independent file (Chapter 7)
    - Metadata should be defined on each binding (or set of bindings) whenever the assumed value of a metadata item is not commonly understood or clearly defined (Chapter 12)
    - When SNOMED CT is bound to a coded text data element, the 'code system' should always be defined as a URI or OID representing SNOMED CT, the 'code' should be the component identifier, and the text should be a synonym of the given concept (Chapter 0)

13.7 Principles for quality assurance

1. **Information model, terminology and clinical specialists are required in terminology binding design and validation to ensure the quality of the bindings (Section 6.3)**
    - Use cases with clinical relevance and ownership is important
    - Information model and terminology specialist have to work together to ensure consistent terminology binding which is in accordance with the concept model.
    - All terminology bindings should undergo technical validation
    - All terminology bindings should undergo clinical verification

2. **Terminology bindings should be re-validated whenever the terminology or the information model artefact is changed (Section 6.4)**
    - Terminology bindings should be retired or updated if the SNOMED CT components that they are dependent on is retired or updated. Consequently, with every bi-yearly release of SNOMED CT, it should be analysed whether changes affect existing terminology bindings
    - Terminology bindings should be retired or updated if the information model artefacts that that they are dependent on are retired or updated.
    - If changes are non-trival technical validation and clinical verification should be repeated

14 Use case examples

14.1 Overview

Information models can be used for many use cases in clinical information systems e.g. data entry, querying, aggregation and communication without a link to standardised terminologies. However, some use cases specifically needs this link i.e. the terminology binding to make sense. Four different sets of use cases are presented in the following within the areas of:

- Data capture
- Retrieval and querying
- Information model library management
- Semantic interoperability.

Various different stakeholders are considered e.g. clinical user such as physicians and nurses, regional information managers and e-health authorities. Focusing on these areas and stakeholders allow a view on what kind of functionality adding terminology bindings to information models might add, but it is not a complete list. Notice that the use cases are the same as those introduced in Chapter 3 where the focus was to introduce and motivate the importance of terminology binding. In this chapter the focus is on terminology binding techniques that would solve the use cases and concrete examples.

14.2 Data capture

In Figure 37, the data capture use cases are presented at the left hand side of the figure. At the right hand side, the binding functions and binding types are presented. Elaborations on how the different use cases may be solved by mastering different terminology binding techniques are presented in the next subsections.

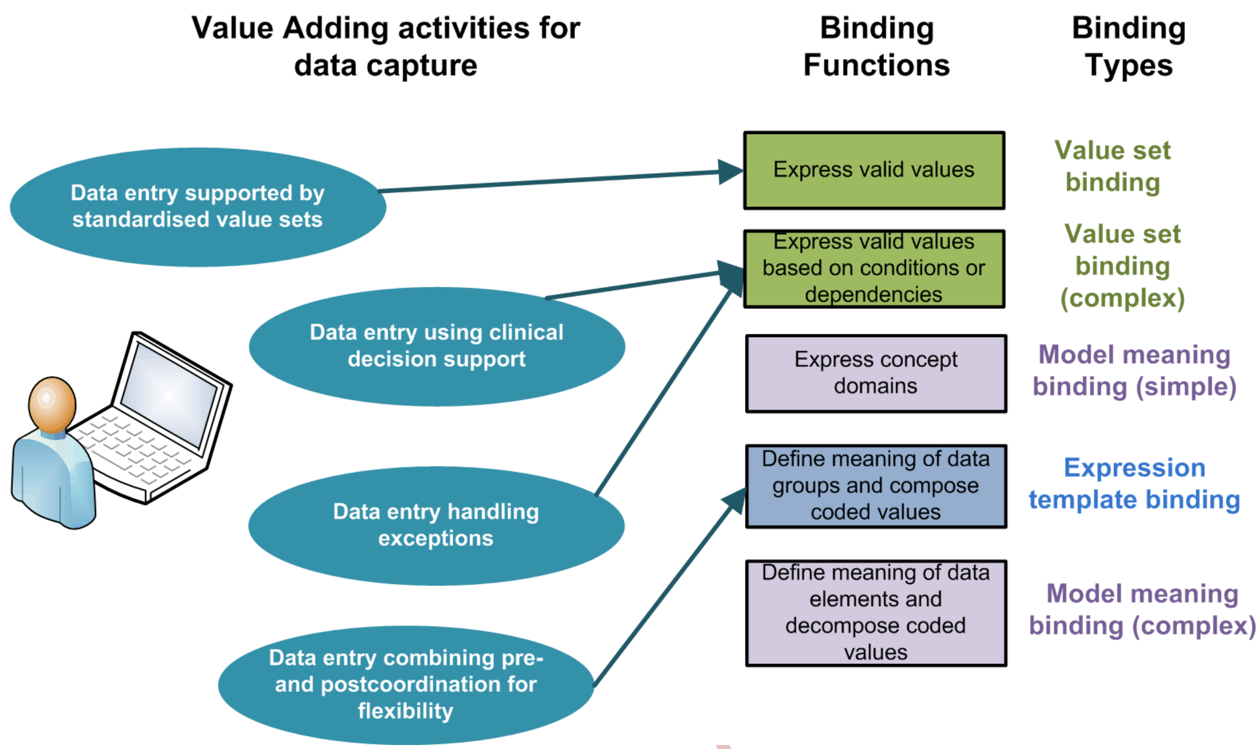

Figure 37 - Data capture use cases including binding functions and binding types

14.2.1 Data entry supported by standardized value sets

The goal of this use case is that when capturing data to populate an information model, the user is presented with valid value set, from which relevant recording can be chosen e.g. reflecting a patient's signs, symptoms, planned or performed procedures etc. To obtain this, value set bindings are used.

The value set binding support that the user does not need to know the whole SNOMED CT and its content model to be able to input a valid expression. In the following is a fracture entry example as oppose to the diagnosis list example presented in Chapter 3. The diagnosis list example would only have included one value set binding. A value set binding supporting the data entry use case for the fracture example is presented in Figure 38. In this Figure, intentional value set bindings are used which require the use of the expression constraint syntax.

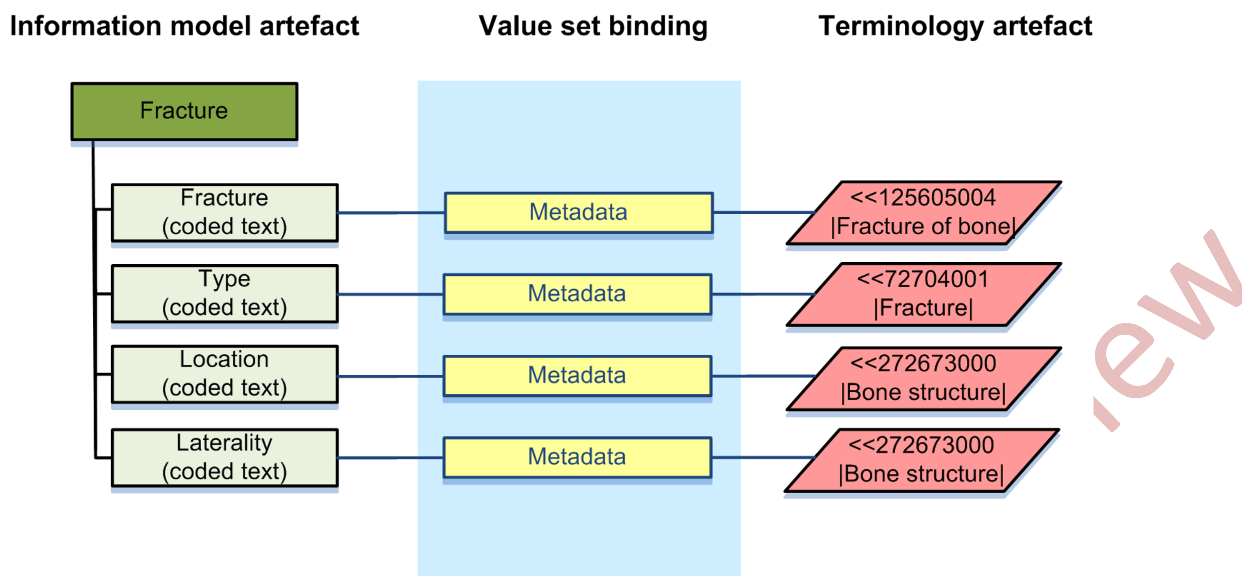

Figure 38 - Binding a fracture information model to relevant value sets using expression constraints. <<125605044|Fracture of bone| means that the valid values are fracture of bone or any subtypes as defined by SNOMED CT. The same goes for the rest of the interpretation of the rest of the expression constraints.

One or all of the bindings in the fracture example could be expressed extensionally using a link to a reference set. This is done in Figure 39. In Figure 40, a user interface using the bindings is illustrated.

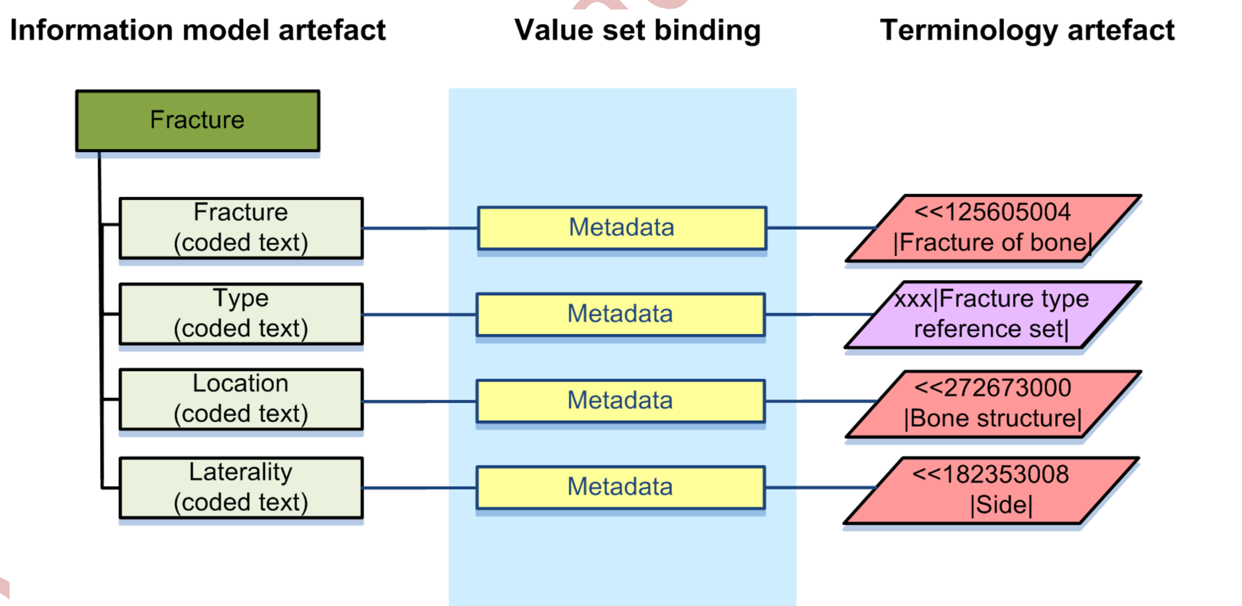

Figure 39 - Binding the fracture information model to relevant value sets using expression constraints and a link to a reference set (purple box).

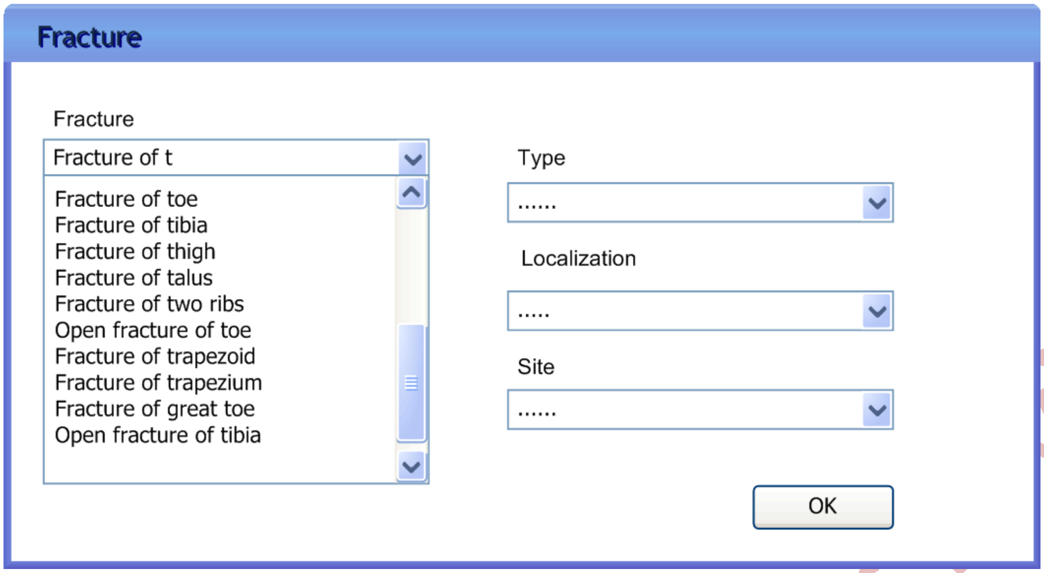

Figure 40 - Simple user interface that uses the fracture information model and its terminology bindings to support user entry.

The search and data entry guide gives additional examples of user interfaces that support this use case, see e.g. Figure 32 and 33.

One problem in value set binding is that the relationship between the fields is not defined. This can lead to inconsistent information overlap between fields. For example, if the user inputs a |open fracture of tibia| in the fracture-field, a |closed fracture| in the type-field, and |bone structure of ankle| there are obvious invalid overlaps of information. There are several ways of handling these overlaps. One solution is to leave it to the user to make consistent input without overlap. Another solution is to avoid fields with overlapping information e.g. only having a “fracture” field. A third option is to include information about the way SNOMED CT interprets the relationship between the fields. This is the solution presented in section 14.2.4.

14.2.2 Data entry using clinical decision support

14.2.3 Data entry handling exceptions

14.2.4 Data entry allowing for combinations of pre- and post coordinated expressions

One of the reasons that structured data entry is not always successful, is that filling out numerous fields is time consuming compared to writing a short narrative. One solution is reducing data entry to the most common entries by reducing the number of fields and inputting more complex expressions in each field. However, reducing data entry will result in that rare entries cannot be expressed in a standardized fashion. Expression template bindings allow for the use of complex pre-coordinated expressions that ease data entry for common entries and structured data entry for rare entries. The following example is a continuation of the fracture-example where an expression template binding is added to the existing value-bindings, see Figure 41.

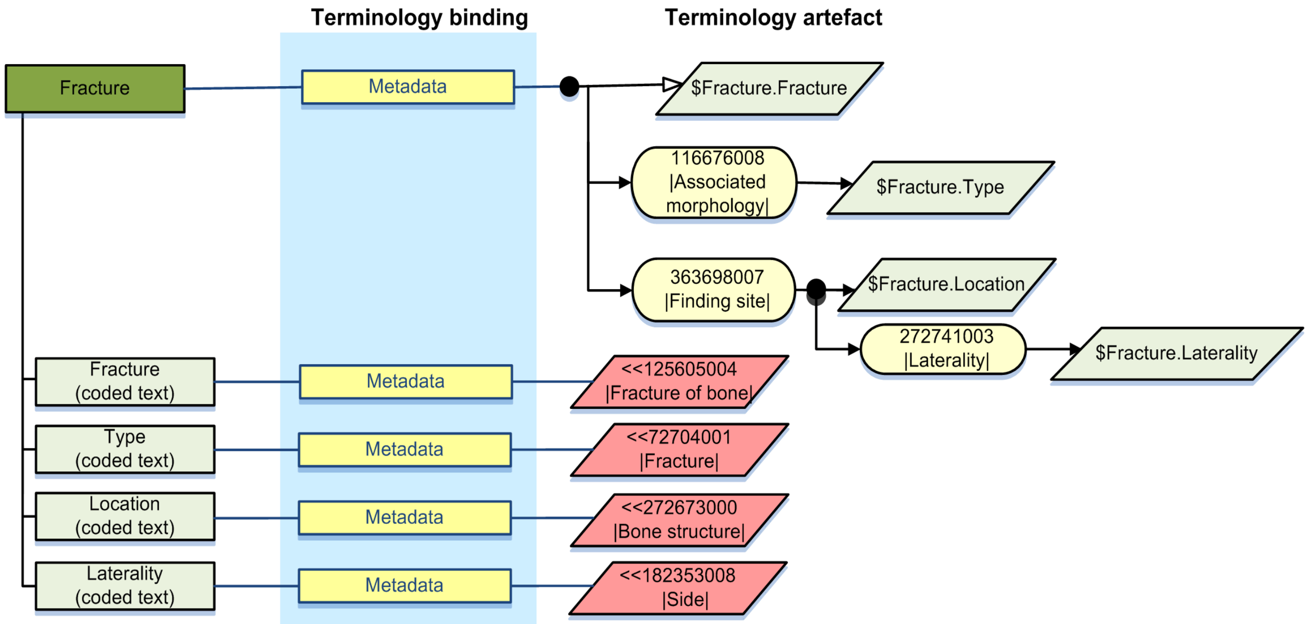

Figure 41 - Template expression binding added to the value set bindings. $Fracture.Type means that the content that can populate this place in the SNOMED CT expression can be drawn from the Type-field in the Fracture model. The relationship can also be interpreted in reverse meaning that the expression/concept in "fracture" can have an associated morphology, if so this value can be put into the Type-field in the Fracture model.

The expression template binding in combination with the value set bindings means that overlap of information can be avoided, and the system will have a way to ensure that only valid expressions are entered. See the example of user interfaces in Figure 42.

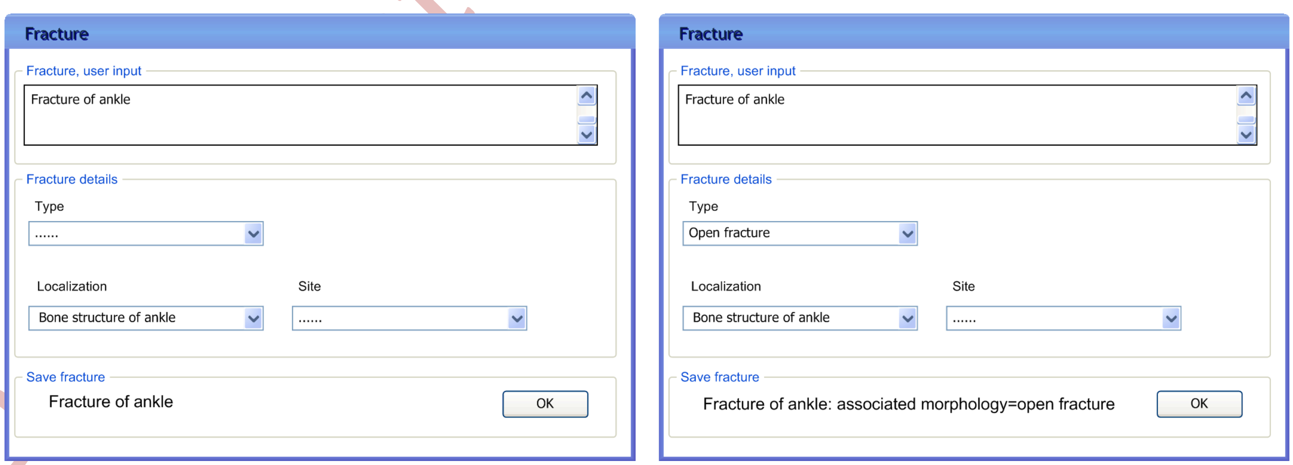

Figure 42 – User interfaces using value set bindings and expression template bindings. In the "Fracture, user input" field any valid subtype of Fracture of bone can be inputted including post-coordinated expressions. Secondly, the content of that user input is interpreted and if there is any mentioning of |associated morphology|, |Finding Site| or |Laterality| in the defining relationships,

the values are inserted in the proper fields in the "fracture details" section. This is the example shown to the left in the above figure where |fracture of ankle| is inputted by the user leading to the Localization-field being populated. In the bottom, the expression that can be saved in the database is shown. In the right user interface, the user has added a type to the fracture, so the expression saved in the database will be changed in accordance with the expression template binding

To fully get the benefit of the expression template binding a few things have to be kept in mind. In the fracture example, If a defining relationship is changed e.g. |bone structure of ankle| is changed to |bone structure of distal fibula| it will be a refinement because |bone structure of distal fibula| is a subtype of |bone structure of ankle|, so you would just add a "finding site=bone structure of distal fibula" to the saved fracture. However, if the |bone structure of ankle| is changed to |bone structure of hand| it would no longer be a |fracture of ankle|. One way to handle this is to only allow valid refinements of the focus concept i.e. it would not be possible to choose |bone structure of hand|. Another option would be to change the focus concept to the top-level of the allowed values if the defining relationships are violated i.e. the saved expression would be |fracture of bone|:|Finding site|=|bone structure of hand|. Using the top-level concept as default focus concept further allows using the three fracture-detail fields to describe various fractures using structured data-entry whether or not they have a pre-coordinated equivalence.
One shortcoming of the presented approach is that it is not obvious how concepts with several role-groups can be represented e.g. |Fracture of tibia AND fibula|. Including an extra fracture detail section when it is necessary would be the easiest solution.

14.3 Retrieval and querying

14.4 Information model library management

14.5 Semantic interoperability

(including message conformance testing)

15 Learning more

15.1 Overview

16 Glossary

The following table contains the definition of any terms used within this document.

Term	Definition

17 References

1. IHTSDO. *SNOMED CT Starter Guide*, IHTSDO, February 2014, Available at: General URL: www.snomed.org/sg.pdf Versioned URL:http://ihtsdo.org/fileadmin/user_upload/doc/download/doc_StarterGuide_Current-en-US_INT_20140222.pdf. Accessed 7/9, 2014.
2. HL7. FHIR Specification Home Page - FHIR v0.0.81. Available at: http://www.hl7.org/implement/standards/fhir/index.html. Accessed 7/3/2014, 2014.
3. SNOMED CT Search and Data Entry Guide, IHTSDO, April 2014, www.ihstdo.org/collab/doc10211
4. Allones JLI, Taboada M, Martinez D, Lozano R, Sobrido MJ. SNOMED CT module-driven clinical archetype management. Journal of Biomedical Informatics 2013;46(3):388-400
5. European Communities. Semantic Interoperability for Better Health and Safer Healthcare. 2009
6. Rector AL. Clinical terminology: why is it so hard? Methods of information in medicine 1999;38:239-52.
7. Garde S, Knaup P. Requirements engineering in health care: the example of chemotherapy planning in paediatric oncology. Requirements Engineering 2006;11(4):265-278.
8. Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. Proceedings of the SIGCHI conference on Human factors in computing systems: ACM; 1990.

18 Appendix A – Information modelling techniques

18.1 Overview

A number of information modelling techniques are used for clinical modelling, which are interesting to consider in terms of their potential impact on terminology binding. These techniques include:

- Modular modelling: Developing separate model fragments, which each represent the data captured about a specific clinical topic, and which can be reused in different situations;
- Composable modelling: Combining a number of small model fragments into a larger model to meet more complex requirements. The models are combined by replacing a reference to a set of possible models (also called a 'slot') with the contents of a specific model. This process is sometimes referred to as 'filling the slots';
- Model specialization: Adding constraints to a broader model to meet more specific requirements;
- Model extension: Extending an existing model with additional
- Constraint-based modelling: Designing models by applying constraints to a base reference model;

For example, FHIR resources use a modular approach to resource development, which are then specialized by the resource profiles, and composed together via the predefined references to other resources. openEHR and ISO13606 use constraint-based modelling to constrain an underlying reference model, thereby defining clinical models and their specialisations. Both openEHR and ISO13606 also use slots to define rules as to how the model fragments (known as archetypes) are composed together to form use-case specific templates. The HL7 RIM is an example of a clinical reference model, which may be constrained by a CDA template to meet specific clinical needs.

Consequently, the modeling domain is complex, and it is typically a requirement that the models of meaning belong to the same family of models. If the complexity of information models was transferred directly to the domain of terminology binding, it would mean that there could not be defined a common approach for terminology binding. However, terminology binding tends to be defined at the international level, using a reference terminology with fewer restrictions on the permitted values, while the implementation-specific model of use is defined with very specific value sets. In other words, terminology binding is a much simpler construct. In this guide, it is assumed that a common approach is possible and we base this common approach on a simplified hierarchical representation of the information model. In the future, it should be discussed with information model standardization organizations whether the terminology binding approach should be re-formulated to fit each individual information model or whether the terminology binding can be an artefact in its own right.

Illustration of puzzle pieces – David slideshow