ISAAC's KOMET and Solor

A Treatise on Symbolic Data Systems

Solor

Table of Contents

List of Figures

List of Tables

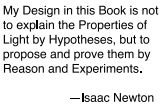
List of Examples

List of Editorial Rules

Part I. Motivation and foundation

Table of Contents

1. Solor Intro



1.1. Preface

Symbolic information uses symbols to represent perception, interpretation, communication, knowledge, facts, data, and planning. Symbolic information is specifically concerned with symbolic representation and interpretational infrastructure.[1]

An interpretational infrastructure establishes meaning, value, and usefulness for the symbols, and can generate and decode the symbols. Without consistent meaning of the symbols, there can be no stable knowledge, facts, or data. After the initial assignment or development of meaning, the interpretation of symbols must remain consistent if the symbols are to be used for perception, memory, communication, or planning.

Symbols have no meaning or usefulness without an interpretational infrastructure. Because the symbols and the interpretational infrastructure are both essential, they must develop or evolve together.

1.2. Motivation and Foundation

The essential challenge of informatics practice within the healthcare enterprise is to quickly deliver a high fidelity reasoned interpretation of principles and facts to the point of care—and then to quickly aggregate these point of care experiences for analytic analysis so that new principles and facts can be formulated and validated as part of a continuous optimization of healthcare knowledge and delivery. To effectively answer this challenge, we must focus on simplification and integration of knowledge assets, and on build, test, deploy, and release processes for delivering these assets to the points of care and analysis. This focus on perhaps mundane topics is not because we think that novelty has no place in our work; rather, that without a focus on aspects of our delivery challenge that are often treated as peripheral to the overall problem, we cannot achieve reliable, rapid, low-risk knowledge-asset development and delivery in an efficient manner.

The Veteran's Administration ISAAC (InformaticS Architecture ACceleration) effort seeks a holistic approach to architecture that supports novelty within a rigorous—and vertically integrated—deployment pipeline that enables knowledge engineers, developers, testers, build managers, and operations personnel to work together effectively to deliver assets to the points of care and analysis. This pipeline must support integrated delivery of iterative revisions of specifications, services, and content which are today delivered by isolated silo organizations who place the implementation burden upon their consumers. This pipeline will be built from existing software-based best practices, and will embrace DevOps culture and practice by emphasizing collaboration and communication while automating the process of product delivery. ISAAC's KOMET (KnOwledge Management EnvironmenT) realizes ISAAC's architecture within a DevOps environment that integrates development, testing, publication, and delivery of specifications, content, and services into a vertically integrated environment that supports continuous delivery.

[1] http://science.jeksite.org/info1/pages/page2.htm

A 2018 whitepaper [interoperabilityreport2018] cited that great strides have been made in healthcare data interoperability in the past decade…the vast majority of clinicians and patients have access to some portion of their health data in electronic format, thanks to the proliferation of electronic health record (EHR) systems installed in clinical care environments. The data in these EHRs usually follow HL7's Consolidated Clinical Document Architecture (C-CDA) as it has become the generally accepted primary data standard for structured clinical data exchange.

However, the whitepaper also found that significant gaps exist in the accurate encoding of the data contained in those C-CDA documents – in an analysis conducted of C-CDA documents produced by various EHR vendors and clinical organizations, the four most frequent problems identified as part of this analysis were that medications should be encoded in RxNorm (frequency of medication appearing in 13.7% of sampled test case documents), vital signs and results should use LOINC (9.2% of sampled documents), vital signs, and results should use unified code of units of measure (UCUM) for physical values (8.7% of sampled documents) and the inclusion of conflicting status information for medications (6.7% of sampled documents)[interoperabilityreport2018]. In short - even though standards and value sets exist for the encoding of data in EHRs, in the vast majority of instances this is simply not being done.

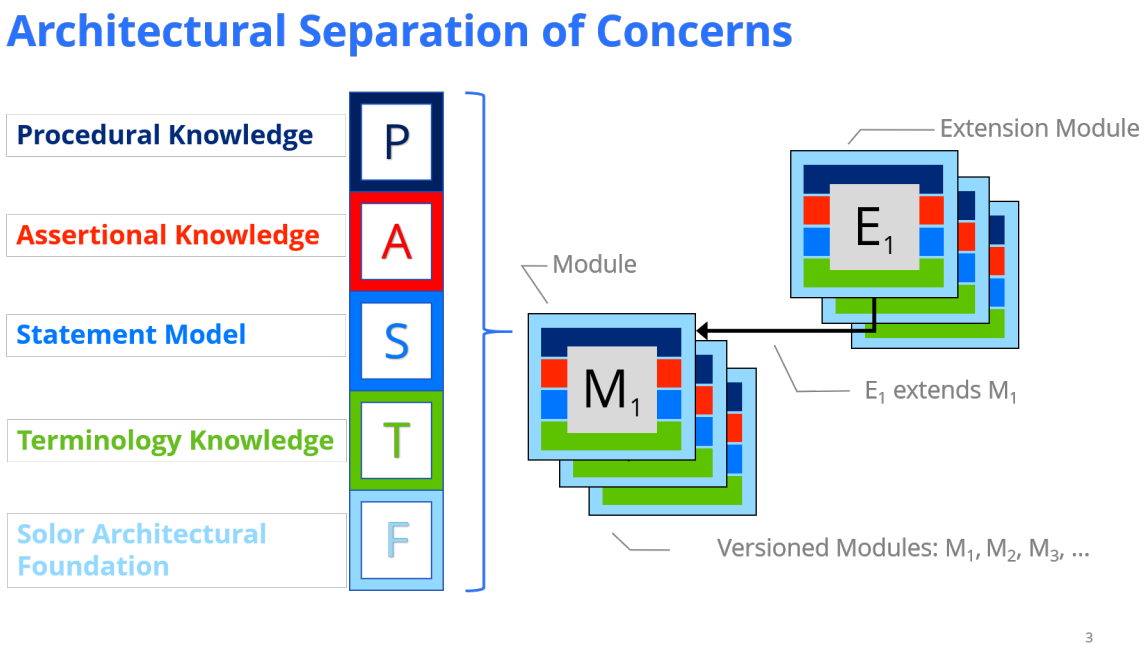
These issues can have a direct impact on patient safety and point to the need to be able to consistently represent and encode clinical data and observations. This is the next great challenge to conquer for health data interoperability to positively influence patient outcomes nationwide through clinical decision support.

When considering these challenges, it can be daunting to consider from where to start. There are hundreds of thousands of clinicians around the world, and up to the current century each was documenting their observations from seeing patients in their own way. Granted there have been efforts over the years to standardize medical terminology in order to encode it properly into systems, but even then there are instances where nuances of medical observations cannot be captured consistently, from something as basic as nuances of language (e.g.. English versus Chinese) to the specifics of how a measurement was actually taken procedurally - for example, in arriving at a quantitative measurement such as 90 beats per minute for a patient's pulse, one clinician may have used a pulseometer, while another may have arrived at that measurement using fingers to the patient's left wrist and a stopwatch. While the quantitative measurement is the same, the procedural information should also be documented and the differences noted.

1.3. Separation of Concerns

A systematic way to think about this (borrowed from the software development industry) is something called Separation of Concerns. Separation of Concerns is an architectural design principle that allows a complete system to be subdivided into several distinct sections. If concerns are well separated from each other, individual sections may be able to be reused, as well as worked on and updated independently to address new requirements and use cases. This is especially useful and important in a medical context given how many different health information and clinical terminology projects are ongoing at any given time, efforts that are often uncoordinated and led by disparate and unrelated standards development organizations.

The following diagram shows how the concept of Separation of Concerns can be applied to the problem of systematically and consistently representing data from clinical observations:



Starting from the bottom to the top of the Separation of Concerns diagram, the layers of the informatics architectural separation of concerns are described as:

Solor Architectural Foundation – Provides an interoperable, integrated common terminology model which concerns (a) the foundation and building blocks of the common model; (b) how the repeatable transformation process of disparate standards into the common model promotes interoperability with other environments; and, (c) how the modules of the architecture are tightly version controlled over time.

Terminology Knowledge – Structured sets of medical terms and codes that define concepts of interest, including descriptions, dialects, language, and semantic hierarchy. This layer also incorporates logical operators and description logic such as ‘representation of absence’.

Statement Model – Packaging of the terminology content in standardized data structures so that they can be readily consumed by the information retrieval process for analysis. Within the data structures, additional detail to describe subject, numerical, and categorical information related to concepts can be added in this layer.

Assertional Knowledge – Translation of guidelines to assist clinical decision making. This includes facts and knowledge upon which concepts and combinations of concepts can assimilate into protocols.

Procedural Knowledge – Information about standard ways to carry out specific procedures as well as other procedural guidelines, e.g. treatment protocols for diseases and order sets focused on particular patient situations. Procedural knowledge, together with assertional knowledge, enables clinical decision support, quality measurement, and supports patient safety. This layer integrates the architectural and terminology layers, incorporates the statement model for information retrieval, and uses the assertional layer to apply rules.

Examining a clinical procedure for controlling hypertension illustrates each of the layers of the informatics architectural separation of concerns. At the Terminology layer, there may be various codes and terms from disparate source terminologies to define the hypertension concept. For example, the concept “essential hypertension” is defined by the ICD-10 CM code I10 and 59621000 in SNOMED CT. Ideally, these

overlapping codes and terms would be oriented to the same parent concept during the transformation and integration process at the Solor Architectural Foundation layer. Furthermore, any updates over time to code sets or value sets that define hypertension (in the NLM's Value Set Authority Center for instance) would be maintained by continuous integration at the Solor Architectural Foundation layer. Moving to the statement layer, blood pressure measurement values may be packaged as a numerical measurement (e.g., systolic BP = 140 mmHg) or the categorial data (e.g., pregnancy induced hypertension vs. renal hypertension) within a standard data structure to facilitate information exchange or retrieval, such as within a FHIR Observation Resource. At the Assertional layer, guidelines such as the recommendation to control hypertension to under 140/90 mmHg might be translated into a clinical workflow facilitated by Health Information Technology (HIT). If HIT is involved with programmed Clinical Decision Support, there may be additional rules to suggest hypertension medications (e.g., beta-blockers, ACE inhibitors) while also including rules to avoid medication contraindications. Finally, at the Procedural-level, there may be a treatment protocol for different kinds of hypertension, including the considerations of, e.g. patient age, co-morbidities etc., which can be generated by an electronic clinical decision support system (Statement + Assertional layers).

Part II. Foundational

Table of Contents

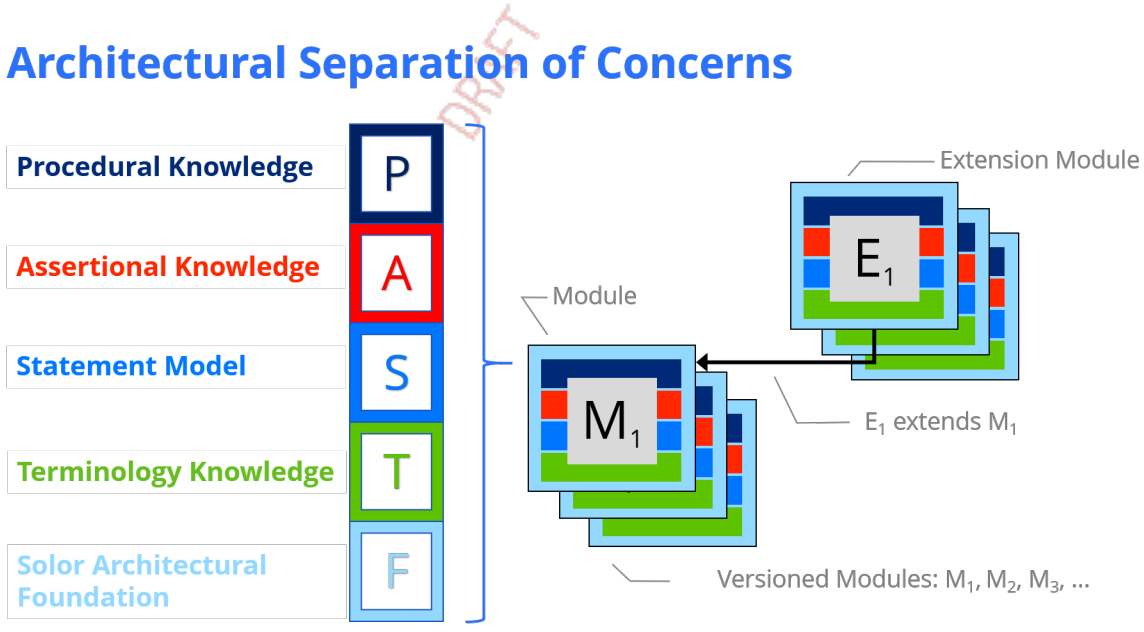
2. Solor Architecture

2.1. Versioning Overview

When dealing with the complexities of the various architectural layers of the informatics architectural separation of concerns, one of the most important things to note is that any one of Solor's architectural layers will be undergoing modifications at any given point in time, as various Standards Development Organizations go through each of their various drafting, balloting, and approval lifecycles. Therefore it is important to establish as a foundation for Solor a versioning and modularity architecture that allows changes and subchanges to be referenced uniquely so that all parties can be on the same page as to a particular version.

For example, the following diagram shows how each module could be given a unique version number and contain all layers of the architectural stack. In the instance that a particular versioned module needs to be extended, an extension module could be added to that main versioned module without the need to go to a completely new full module version. This arrangement accounts for the constant change in the healthcare interoperability space while still allowing two organizations to baseline on the same version for testing or exchange purposes (i.e. Module M13, Extension E25)

Figure 2.1. Versioning; Modules and Extensions



2.2. Architecture

An interoperable, integrated terminology model concerns (a) the foundation and building blocks of the common model; (b) how the transformation process of disparate standards into the common model is made repeatable and interoperable with other environments; and, (c) how the modules of the common model are tightly versioned controlled over time.

In this chapter we are concerned with detailing Solor's architectural foundation that will support semantic interoperability. We achieve this foundation in two ways:

1. Use of SNOMED CT, RxNorm, and LOINC as the primary building blocks for the foundational architecture (more on this in chapter 3).

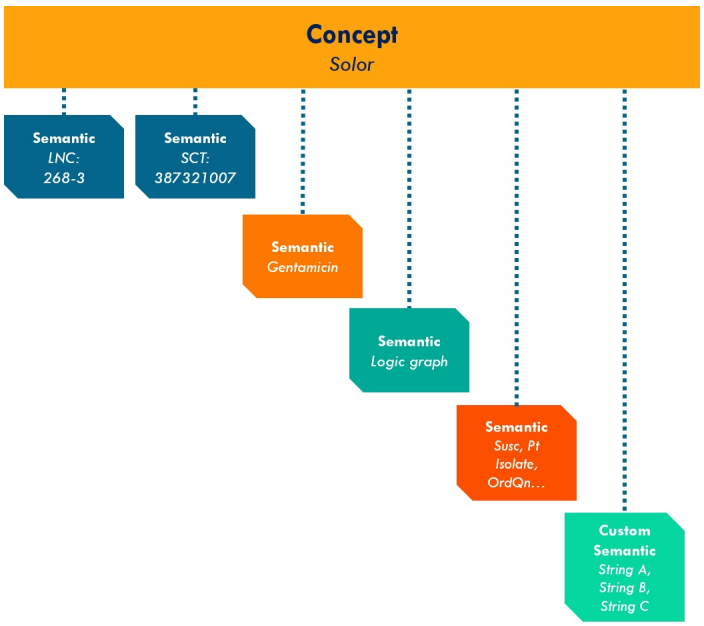
2. Enablement of semantic operability within the foundational architecture through normalization of representation and achieving coherence within and among the primary building blocks of the architecture.

Solor is an ecosystem that allows users to import, transform, and view content from disparate medical terminologies, all in one common model. Users can navigate and search Solor content, view details of the data elements, and select specific concepts to view more information. As Solor is open-source, developers are encouraged to build on top of existing functionalities.

We adopted contemporary software principles to create a multilayered architecture for integrating standard medical terminologies. We sought to adhere to three main principles in our architectural design: (a) to store concepts from medical terminologies so that one could apply classifiers and identifiers; (b) to allow for versioning and updates over time in a way that preserved concept orientation; and, (c) to promote collaborative, distributive workflows for developers.

2.2.1. Building Blocks

Solor has two fundamental building blocks: concepts and semantics. Concept is defined as an idea or a general notion. When abstracted out, it can be used to represent any idea, whether that is a medically related idea (e.g., heart attack) or an idea to represent metadata (e.g., a synonym or a fully specified name). A semantic enables addition of semantic data to the underlying concept's content, in a standardized way that provides for the same means of identifying, modularizing, and versioning. In other words, a semantic is attached to a concept to provide contextual meaning to the concept. Semantics can be grouped together in a collection to form an 'assemblage'. An assemblage consists of semantics that reference a component and provide additional data to that member for some purpose. Solor also has 'description semantics' with additional metadata specifying details including but not limited to 'language' and 'description type'.



Concept: represents any idea, whether that is medically related (e.g., heart attack) or metadata (e.g., author)

•Fields: Universally Unique ID

String Semantic: provides identifier to the concept in a standardized way

•Fields: Source Terminology Code

Description Semantic: provides a human-readable description

•Fields: Fully-Specified Name, Long Common Name, Short Name, and Display Name

Logic Graph Semantic: provides description logic for traversing hierarchies and for specifiying the view of the relationships between and amongst data elements

•:Fields: Parent-child-sibling relationship

Assemblage: Grouping set (i.e., “assemblage”) specifically created to store all of the data elements, data types, and metadata for a particular use case

•Fields: Variable to accommodate a variety of use cases. For instance, [StringA, StringB, StringC, StringD], [ID, String, Integer, Concept]

2.2.2. Transformation Overview

After a standards developer releases its content, a process will need to occur to transform data from its native format into Solor components. This programmatic process is tailored to each incoming data stream, where it will account for data represented in its original format. Other than transforming and applying versioning coordinates, the underlying process will also address the notion of dependency. For example, SNOMED US Extension will have a dependency on SNOMED International, and relationships from the LOINC-SNOMED collaboration effort will have a dependency on SNOMED and LOINC.

Once the content is in Solor, there is a step where equivalency is determined through various methods where concepts of the same idea are aggregated. For example, Gentamycin from SNOMED is the same as Gentamycin from LOINC, and is also the same Gentamycin from RxNorm. The end result from this process is the creation of a Solor concept that is devoid of any source information (but will have traceability). This end result is what will be exposed to the user to view and use. In the Gentamycin example, a user will find this concept that is devoid of any source information and will not need to know if this is the SNOMED/LOINC/RxNorm Gentamycin that needs to be selected. If the Gentamycin concept was used in the context of identifying what medication the patient is currently taking, then the underlying process will be able to transmit the RxNorm code if the receiving system is expecting RxNorm codes. Conversely, if Solor were adopted more universally, the transmitted information could be isolated to the Solor Gentamycin rather than a distinct code from a specific terminology.

2.2.3. Identifiable Components

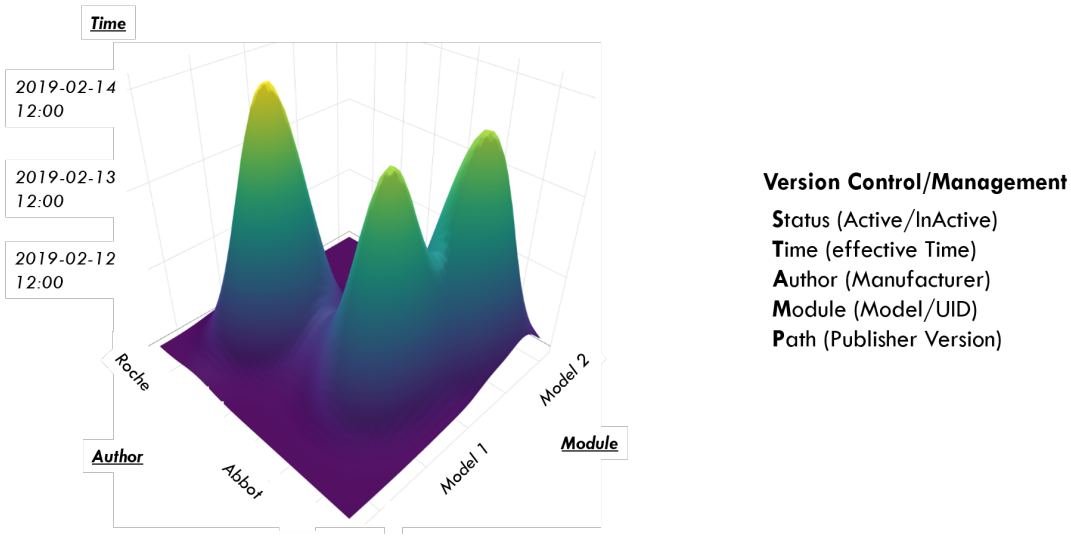
A universally unique identifier (UUID) is a 128-bit number used to identify information in computer systems.(ref]) The identifiable component layer of Solor manages the reproducible assignment of UUIDs to all imported components as well as the assignment of primordial UUIDs to all internally generated components. If imported components already provide UUIDs to identify components, those UUIDs will be used. The identifiable component layer must allow components to have more than one UUID identifier, and if previously independent components are given each other’s identifiers as alternate identifiers, the identifiable component layer must dynamically merge the parts of these previously distinct components into a single integrated component. This merging of components by merging identifiers is a simple means for managing duplicated content as it is identified.

2.2.4. Chronology

2.2.4.1. STAMP Coordinate

The chronology component of the architecture layer provides a means to generically represent the revisions to a component over time, and to index those revisions by status (e.g., active, inactive), effective time of change, author of change, module within which the change occurred (international edition, US extension, etc.), and the development path of the change (development, release candidate, etc.). Taken together, these fields can be referred to as a version's STAMP (status, time, author, module, and path). STAMP provides a foundation for version control and configuration management of all the components of the information architecture. The STAMP will provide a means to modularize content so that modules can be turned on and off depending on specific use cases, and that modular content can be developed independently from unrelated modules. This modularity will enable simplified development and quality assurance processes for each module. The following figure shows the UML representation of Solor's chronology layer on the left, and that of STAMP on the right.

Figure 2.2. Chronology and STAMP



There are also more nuanced components within Status, Time, Author, Module, and Path that can be configured. These include: 'Allowed States' (related to Status), precedence, and the ability to specify groups of modules in a 'Module Set'. Precedence can be set to stratify the mathematical constructs surrounding the components (e.g., path, time) so that one component can be prioritized over the other.

In summary, STAMP provides a high degree of configuration for navigating versions of content and how that content may be interacted within the Solor ecosystem.

2.2.4.2. Language Coordinate

The language coordinate provides the ability to configure details around what language of content to provide, and to select a particular dialect, and/or the order of dialects available in the Solor ecosystem. This also provides the ability for users to get the exact level of granularity of content they desire.

2.2.4.3. Logic Coordinate

The logic coordinate allows configuration of description logics and formal knowledge representation of Solor content. The fundamental modeling concept is an axiom—a logical statement relating roles and/or

concepts. Within the logic coordinate, users can specify which classifier to use (e.g., Snorocket), and which concepts they want to classify in their given use case (e.g., Solor content vs. Health content). Users can also specify how they want to configure the stated and inferred parent-child (supertype-subtype) relationships that are either available in the source terminology native logic, or through additional integration provided by the Solor common model.

2.2.4.4. Manifold Coordinate

In order to easily exchange the complex configurations of facets of the Solor common model, we need a unifying object to do that. The Manifold Coordinate restricts the instantiation of the configurations of the STAMP coordinate, Language Coordinate, and Logic Coordinate to one object, and provides a global access point to it (i.e., Singleton design pattern). In other words, it acts as an abstraction layer between the nuanced configurability of the other coordinates, and how it is ultimately executed with the Solor ecosystem is used.

2.2.4.5. FLWOR

FLWOR is an acronym for "For, Let, Where, Order by, Return". The programming language XQuery defines FLWOR as an expression that supports iteration and binding of variables to intermediate results. FLWOR is loosely analogous to SQLs SELECT-FROM-WHERE and can be used to provide join-like functionality to navigating content.

- For

  - selects a sequence of nodes

- Let

  - binds a sequence to a variable

- Where

  - filters the nodes

- Order by

  - sorts the nodes

- Return

  - what to return (gets evaluated once for every node)

The advanced version control and modularity provided by the Solor Architecture is embedded within a FLWOR framework. This allows for complex querying capabilities to navigate and search for concepts, data elements, metadata, the relationships between and amongst these data elements, and how they change over time and/or differ between and amongst the modules. The complexities of these queries are abstracted into a user-friendly graphical user-interface, and users are provided precise options for configuring their queries and use cases.

2.3. Challenges

Solor is an integrated clinical transformation process to represent and bring together disparate terminology standards by using a single model that can encompass any customized content. In our experience with building the Solor semantic architecture and transformation process, we have come to understand that health IT systems must address the following antipatterns:

2.3.1. Accidental Complexity

Accidental (or incidental) complexity is complexity that arises in computer programs or their development process that is non-essential to the problem to be solved. While essential complexity is inherent and unavoidable, accidental complexity is caused by the approach chosen to solve the problem. Some examples of accidental complexity as they relate to informatics are described in the following sections.

2.3.1.1. Semantic-laden Identifiers

Solving a distributed identifier allocation problem by using namespaces that are assigned to organizations (or committees in the case of HL7), semantics are often introduced into the identifier, which some developers use to identify what organization created the components that were associated with those identifiers. Exposing derivable semantics in the identifier can lead to complexity when users/developers demand that the semantics be maintained, which may result in unnecessary retirement as described in the next section. Reliance on UUIDs rather than on identifiers with derivable semantics would eliminate this complexity.

2.3.1.2. Unnecessary Retirement

An unintended side effect of using identified namespaces as part of distributed identifier assignment is an increase in the complexity of transferring responsibility for a component from one organization to another. This complexity includes an elaborate sequence of marking a component for retirement in one release, actually retiring it in a subsequent release, and creating an essentially identical component with an identifier derived from the new organization's namespace. Furthermore, there is the need for the creation of mapping solutions to keep historical relationships between components retired for these reasons to the current concepts that replace them. Again, reliance on UUIDs rather than on identifiers with derivable semantics would eliminate this complexity.

2.3.1.3. Post-coordination

Terminology models sometimes make it necessary to require post-coordination to provide domain coverage at the point of care, however, the information models we use in healthcare typically can't handle post-coordination well. Reliance on the information model to represent post-coordination has introduced complexity that might be avoided if we used a dynamic means to assign unique identifiers to post-coordinated expressions.

2.3.1.4. Accidental Complexity Solutions

Accidental complexity must be minimized in any good architecture, design, and implementation. Working in short iterations with ongoing design reviews may help reduce accidental complexity. We must also develop an example implementation in parallel with the architecture, so that complexity can be identified early, and evaluated critically with respect to the essential or accidental nature of that complexity.

2.3.2. Design by Committee

A project that has many designers involved but no unifying plan or vision.

2.3.2.1. No Unifying Vision

Design by committee is the result of having many contributors to a project, but no unifying vision. A complex software design is the product of a committee process. The design has so many features and variations that it is infeasible for any group of developers to realize the specifications in a reasonable time frame.

2.3.2.2. Interoperability at the Expense of Operability

Interoperability provides an illusion of operability between disparate systems, and therefore there is no need to standardize.

2.3.2.3. Design by Committee Solutions

A solution to design by committee is to articulate a set of architectural principles to which architectural components will be evaluated against, and to have the committee be advisory to an architect that provides the unifying vision.

2.3.3. Stovepipe

The Stovepipe Enterprise antipattern is characterized by a lack of coordination and planning across a set of systems. If every subsystem has a unique interface, then the system is overly complex. Absence of common multisystem conventions is a key problem for systems. For example, currently, essentially no terminology systems are the same with regard to their representation and semantics, despite the requirement that they must work together.

2.3.3.1. Overlapping and unreconciled models

SNOMED CT and LOINC are classic examples of two terminologies that are proposed for common use in health IT, but that are not well coordinated, and have unreconciled content (content that is not made consistent or compatible). As an example of unreconciled content, SNOMED CT and LOINC both have representations for Amoxicillin. In LOINC, Amoxicillin is a textual value in the has-component field of the concept:

AMOXICILLIN [MASS/VOLUME] IN SERUM OR PLASMA

HAS-COMPONENT: AMOXICILLIN

While SNOMED CT has the concept:

AMOXICILLIN MEASUREMENT (PROCEDURE)

COMPONENT: AMOXICILLIN (SUBSTANCE)

In SNOMED CT, Amoxicillin is also a concept, rather than just a text value.

From an end-user's perspective, the artificial separation and uncoordinated development of these important systems has been a burden. RxNorm may help bridge the medication components of the overlap, but there are other overlapping domains (method, type of scale, system, time aspect, and non-pharmaceutical components) that RxNorm does not cover. The UMLS may help us formally reconcile some of these other domains, but if coordination and reconciliation can be part of the development processes for these sources, rather than a cleanup exercise for implementers, we can allocate resources to solving more compelling problems.

We hope that the newly announced cooperative agreement between IHTSDO (owners of SNOMED CT) and the Regenstrief Institute (owners of LOINC), and the NLM (owners of RxNorm) will change the coordination of these systems in a significantly helpful way. Although SNOMED CT and LOINC are classic

examples of overlapping and unreconciled models, there are many other examples. The UMLS identifies over 150 sources, most of which are uncoordinated, and have independent models. These overlapping and unreconciled models create an unnecessary burden for the implementer.

2.3.3.2. Uncoordinated development

Today, related components from different organizations do not share their work prior to a release. The result of this lack of sharing is that dependent components are always out of date with the latest release of the underlying standard. For example, how can you keep a mapping of SNOMED CT to ICD-10-CM components up to date, when it takes 6 months after the release of SNOMED CT to update and quality assure the map? As an implementer, does that mean you should wait 6 months for the map to be updated before deploying the latest SNOMED CT release? What if the new SNOMED CT release contains new content that may improve the diagnosis, treatment, or prevention of disease? Is it really acceptable to delay implementation of the latest SNOMED CT release by 6 months while waiting for dependent system components to be updated after the fact?

2.3.3.3. Stovepipe solutions

The primary solution for the stovepipe systems we are working with is to break down the barriers that prevent collaborative development of content, tools, processes, and ultimately architecture. Today, deployment delay is not a significant issue because clinical decision support is relatively nascent, and pharmacy, laboratory, and clinical systems are poorly integrated. However, if we successfully create compelling decision support on an integrated and shareable platform, coordination of development and release cycles among clinical terminologies, logical representation, clinical facts, and clinical knowledge bases will become increasingly important. We must prepare for success and work to better coordinate development among dependent components.

Here we propose leveraging opportunities that are helping to break down these barriers. Those opportunities include acquisition and development of open-source tooling. Improvements in open-source tooling will help break down collaborative barriers significantly. Such improvement is a fundamental focus of our architecture effort. The solution to the stovepipe antipattern is effective collaboration without barriers of proprietary concern.

2.4. Summary

Currently, medical terminologies come from different sources and are represented by disparate models. However, by using a common model that integrates these terminologies seamlessly, Solor's architectural layer can display content from different sources after the Solor transformation process. Users will consequently not need to burden themselves with unnecessary complexities, and can instead focus on the meaning of medical content. Built upon an architecture intended to facilitate semantic interoperability, Solor stores concepts with UUIDs and classifiers, is maintained by robust version control, and promotes modular, collaborative development. Next steps include developing a list of agency-specific and industry-specific use cases for Solor upon which a formative evaluation approach and data collection and analysis can be conducted.

Part III. Terminology

Table of Contents

3. Concepts and Codes

3.1. Introduction

Terminology systems are increasingly critical components for achieving interoperability across applications in the healthcare domain. A standard terminology is one that has wide industry acceptance or use.The role of standard terminologies in achieving interoperability for the purposes of advancing patient care is well documented. Ideally, these clinical terminology standards intend to provide rules to allow for the exchange, integration, and management of electronic clinical information. The federal government recognizes the benefit of standard terminologies and promotes their development and use. The *Federal Health IT Strategic Plan 2015-2020* set a strategy to encourage consistent terminology standards implementation in Electronic Health Records (EHRs) and encourage use through federal payment policies.

Standards are obtained from a variety of efforts, cover different domains of clinical and nonclinical content relevant to the EHR, and serve various purposes. Currently, no single terminology or classification system contains everything that is needed for the medical record.

The scope of content covered in terminologies varies from focusing on a very specific domain to covering multiple domains. The main terminologies supported in Solor (i.e., SNOMED CT, LOINC, and RxNorm) all follow very structured and persistent code practices. They never delete codes but in some cases (i.e., RxNorm) they are moved to a separate table.

3.2. SNOMED CT Concepts

Systematized Nomenclature of Medicine-Clinical Terms (SNOMED CT®) is a comprehensive clinical terminology, maintained by the International Health Terminology Standards Development Organization (IHTSDO) representing over 300,000 concepts including disorders (22%), procedures (17%), body structures (11%), clinical findings other than disorders (10%), and organisms (10%).

SNOMED CT concepts span multiple domains, from findings and disorders to procedures and pharmaceutical and biological products. SNOMED CT concepts exist at multiple levels of granularity, and mechanisms exist to add missing content to ensure that information is coded at the most granular level possible. SNOMED CT concepts are identified with a 6- to 18-digit integer that has some structure that describes where the identifier originated (the original creator of the identifier) and what type of identifier it is.

3.3. LOINC Codes

Logical Observation Identifiers, Names, and Codes (LOINC®) is a terminology representing about 50,000 clinical and laboratory observations, health measurements, and documents, developed and maintained by the Regenstrief Institute.

LOINC codes are used to identify laboratory and clinical measurements, observations and documents. Each named concept in LOINC is given a LOINC code which has no structure other than a check digit appended to the end. In addition, all of the components that make up the LOINC name are assigned LOINC Part Codes (LP) to them in the LOINC Database. LOINC Parts do not follow the same policies and maintenance practices as LOINC terms and are not intended to be used as a standalone terminology. Some LOINC codes are also associated with answers (LA) and answer lists (LL) in the LOINC Database.

3.4. RxNorm

RxNorm is a terminology for human clinical drugs in the U.S., representing drug properties such as ingredient, strength, and dose form, maintained by the National Library of Medicine (NLM) and distributed via the Unified Medical Language System (UMLS).

RxNorm provides normalized clinical drug names and relationships between those names and common drug vocabularies for the purpose of easing the exchange of clinical drug information between systems that use different drug vocabularies. Every concept in RxNorm is given a unique RXCUI. The names and codes from the common drug vocabularies are then linked to that RXCUI.

3.5. UMLS

Terminology systems typically consist of the following elements:

- Coded Concepts – the discrete units of knowledge managed within the terminology. They typically consist of numeric codes and textual preferred names, synonyms, and descriptions.
- Concept Hierarchies – the logical organization of concepts into parent-child and ancestor-descendant relationships that express the semantics of generalization and specialization. The hierarchical organization of a terminology may be explicitly expressed through stored parent-child and ancestor-descendant links, or it may be implicitly expressed through the logical definitions of individual concepts that a computer can use to infer parent-child and ancestor-descendant relationships.
- Value Sets – named lists of individual concepts that represent more abstract categories useful in decision-support logic.

New applications and new medical knowledge constantly call for expansion and enhancement of existing terminologies. However, since terminology systems are often non-static, incomplete and under specified, inconsistencies may be introduced.

While many of these challenges are related to terminology evolution, others may be related to the design of the standard clinical terminologies themselves. Cimino notably described the challenges of concept orientation, completeness, correctness, currency, granularity, and redundancy when designing re-usable medical terminologies. Today, 20 years later, a menagerie of inconsistent and overlapping terminology models hinders efforts that try to store and analyze encoded clinical data. Several efforts aim to assist. The National Library of Medicine (NLM) integrates terms and codes from over 150 source vocabularies by concept, attribute, and meaning in the Unified Medical Language System ® (UMLS) Metathesaurus. The NLM, also, in collaboration with the Office of the National Coordinator for Health Information Technology and Centers for Medicare & Medicaid Services, hosts the Value Set Authority Center (VSAC). The VSAC aims to provide lists of values, codes, and names (i.e., value sets) from standard clinical terminologies to represent clinical concepts.

These tools, while helpful, have gaps. Raje et al. highlighted issues with completeness, correctness, and redundancy when they found gaps in the UMLS Metathesaurus' coverage of disease concepts. Similarly, Winnenburg et al. highlighted duplicate value sets in the VSAC, and showed that 19% of value sets in 2011 contained invalid codes. In subsequent work, they highlighted issues related to granularity by evaluating over 1,000 value sets and found that value sets varied vastly in size with some only containing one code, while other value sets included over 20,000 codes. Similarly, Bahr et al. showed issues with concept orientation by analyzing medication value sets and found extraneous and missing ingredients in both the value sets and drug classes.

These issues related to integrating clinical content have a direct impact on patient safety and point to the need to be able to consistently represent and encode clinical data and observations. Therefore, quality as-

surance is an indispensable part of the terminology management lifecycle. A central limitation of integrating controlled medical terminologies is that they often lack any formal model to denote the relationships among constituent data elements.

Recently, however, development teams for SNOMED CT, LOINC, and RxNorm have partnered to promote interoperability. Developers can now leverage SNOMED CT's representation model for the building blocks of LOINC, and a new drug model in SNOMED CT facilitates extensions and consistency to RxNorm. Bodenreider et al. wrote about the recent collaboration: "while this evolution leads to greater compatibility and interoperability, integration of SNOMED CT, LOINC, and RxNorm still requires mappings among the three terminologies. Moreover, these three terminologies use different formalisms and tools for their representation, have their own release cycles and versioning mechanisms, which makes their seamless integration non trivial, if at all possible."

3.6. Solor

Solor integrates the content from its native format into a common Solor format. Once the content is in Solor, there is a step where equivalency is determined through various methods where concepts of the same idea are aggregated. For example, Gentamycin from SNOMED is the same as Gentamycin from LOINC, and is also the same Gentamycin from RxNorm. The end result from this process is the creation of a Solor concept that is devoid of any source information (but will have traceability). This end result is what will be exposed to the user to view and use. In the Gentamycin example, a user will find this concept that is devoid of any source information and will not need to know if this is the SNOMED/LOINC/RxNorm Gentamycin that needs to be selected. Solor concepts are identified using a Univerally Unique Identifier (UUID).

3.7. Solor Integration - Integrating LOINC Method Attributes and SNOMED CT Concepts

The collaborative agreement between LOINC and SNOMED CT developers has enabled informaticists to leverage SNOMED CT for the representation of the building blocks of LOINC (e.g., method) and for a more consistent representation of clinical and laboratory observations in SNOMED CT. We utilized the derivative works from this collaborative effort to represent LOINC and SNOMED CT in Solor, an open-source ecosystem for integrating disparate medical terminologies in a common model.

Seamless integration of LOINC and SNOMED CT is non-trivial because LOINC and SNOMED CT have different semantics models, and use different formalisms and tools for their representation, have separate release cycles, and different versioning mechanisms. Furthermore, the initial collaboration between LOINC and SNOMED CT provided equivalent concepts between LOINC laboratory concepts and SNOMED CT concepts but did not include many clinical concepts from LOINC. In this work, our objectives are (1) to assess the extent to which LOINC method attributes can be represented by concepts in SNOMED CT, (2) to describe how to integrate equivalent LOINC method attributes and SNOMED CT concepts in the Solor common model, and (3) to explore the benefits and challenges of integrating LOINC and SNOMED CT in the Solor common model.

Methods: We sought to identify the overlaps and gaps between method attributes in LOINC and concepts in SNOMED CT, and integrated the equivalent concepts in Solor – an integrated common model for medical terminologies. First, we gathered the list of method attributes from LOINC version 2.63 and mapped each to a concept unique identifier (CUI) from the UMLS(using the UMLS API). For CUIs representing a LOINC method attribute, we retrieved associated atoms from SNOMED CT in the UMLS.2 Next, we imported each LOINC identifier and attached description logic defining whether there was equivalency of the method attribute to SNOMED CT concepts. When there was an overlap, a Solor-navigation concept

was created which facilitated an inferred taxonomy representation with semantic context from SNOMED CT attached to the right LOINC method attribute. Finally, we evaluated the alignments obtained between LOINC method attributes and SNOMED CT concepts to determine what method attributes in LOINC were not covered by concepts in SNOMED CT.

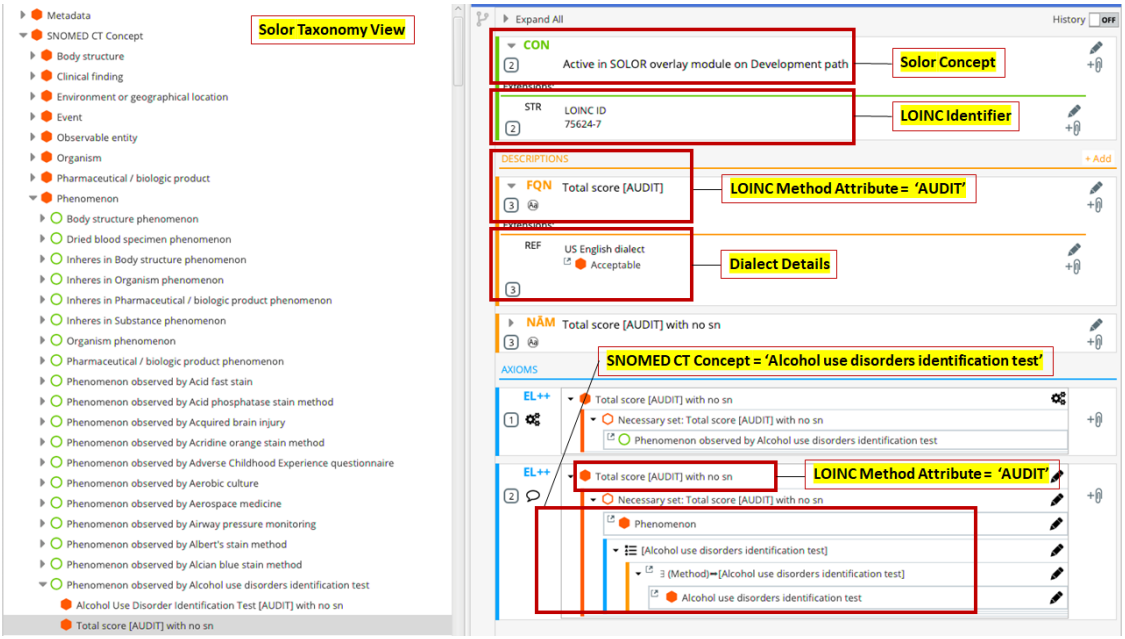
Results: *Semantic profile of LOINC method attributes* – The method axis of LOINC is used to specify methods used for particular clinical observations and measurements. The distribution of the most prevalent semantic groups found in LOINC method attributes include procedures (58%), concepts/ideas (14%), living beings (13%), occupations (8%), and disorders (2%). Whereas, SNOMED CT concepts represent disorders (22%), procedures (17%), body structures (11%), clinical findings other than disorders (10%), and organisms (10%).1

Coverage of LOINC method attributes by the UMLS and SNOMED CT – Of the 1702 LOINC methods, 1688 (99%) were mapped to a UMLS CUI. Sampling the 1% not mapped implies provisional codes added to a LOINC version update that were not yet added to the UMLS version release. We computed the coverage of LOINC method attributes by SNOMED CT by analyzing the count of LOINC method attributes that shared at least 1 UMLS CUI with a corresponding SNOMED CT concept. Of the 1688 CUIs that represented LOINC method attributes, 383 (23%) were associated with a SNOMED CT concept.

Solor Transformation –Solor is an ecosystem that allows users to import, transform, view, and export content from disparate medical terminologies, all in one common model. Users can navigate and search Solor content, view details of the data elements, and select specific concepts to view more information. Solor has two fundamental building blocks: concepts and semantics. A concept is defined as an idea or a general notion and is represented by a universally unique identifier (UUID). A semantic is attached to a concept to provide contextual meaning and semantic data to the concept's content.

Integrating LOINC method attributes into Solor required a transformation process in which the LOINC data was transformed into Solor components using these defining relationships to create OWL EL++ description logic definitions. The LOINC identifier was used to create a Type 5 UUID for a Solor concept, so that the identifiers used in Solor are idempotent, and are derivable directly from the LOINC data. Additionally, the original LOINC identifier for this concept is properly represented, as are the other data elements required by the LOINC license. These Solor concepts integrate SNOMED CT and LOINC semantics; the LOINC Method semantic and SNOMED CT semantic are grouped under the same Solor concept. Figure 1 exemplifies the integration: the LOINC method attribute is "AUDIT", and the SNOMED CT concept is "Alcohol use disorders identification test".

Figure 3.1. Solor Editor: Representation of LOINC and SNOMED CT in a common model



Conclusion: LOINC method attributes include concepts ranging from procedures, administrative concepts, occupations, and a small number of disorders and phenomena. Only about 23% of LOINC method attributes can be directly represented by a SNOMED CT concept. The overall interoperability between LOINC method attributes and SNOMED CT concepts was limited at best. Interestingly, there were a number of procedures and specimen source details in LOINC method attributes with little or no coverage by SNOMED CT. One suggestion is for developers of LOINC and SNOMED CT to review these gaps and include better coverage in future versions, if appropriate.

Solor may assist in providing a collaborative ecosystem to host local extensions for SNOMED CT to represent LOINC method attributes. Currently, implementers of LOINC and SNOMED CT must traverse the distinct hierarchies of each source terminology and version. Integrating the terminology content of LOINC and SNOMED CT into the Solor common model may have a beneficial impact on the usability (i.e., reduced burden) for implementers in both traversing distinct formalisms and maintaining version control. It may be helpful to communicate more specific details about LOINC method attributes by leveraging the right SNOMED CT concepts for additional details about the method attributes. Next steps include conducting a formative evaluation with a purposive sample of experts in standard clinical terminologies to assess the benefits and challenges of integrating and representing overlapping LOINC and SNOMED CT content in Solor's common model. We aim to have this evaluation completed by Summer 2019 and will include these updated results if accepted. Solor's integration of disparate medical terminology content may help implementers and authors of medical terminologies with orienting concepts and traversing relationships between disparate standards.

3.8. Evaluating the impact of implementing Solor

Electronic clinical quality measures (eCQMs) and clinical decision support (CDS) alerts are triggered by clinical data that is encoded by standards based clinical terminologies. Because these measures and alerts intend to promote evidence-based clinical processes, variations in data caused by having inaccurate or

antiquated implementations of underlying terminology standards may impact the ability of clinicians to assess care and improve quality. Jean-Jacques et al. showed that health information technology-supported quality improvement (QI) initiatives can decrease disparities for some chronic disease management and preventive measures;(ref12_ms). Data-driven QI efforts rely heavily on patient-level data generated by eCQM reports or CDS alerts, which are dependent upon standards-based encoded EHR data. If clinicians rely on inaccurate implementations of eCQMs and CDS, then they may have lists/alerts with patients intended to be excluded from a measure/alert, and may therefore, target inappropriate patients for therapies, such as recommending aspirin use for someone at high-risk for a fatal bleeding event. Furthermore, their lists/alerts will not include the newly added patients who may need certain therapies to improve outcomes. Having accurate eCQMs/CDS may translate into potential lives saved, and avoidable harms. Furthermore, the comparability of clinical quality performance scores between healthcare organizations is negatively impacted by the vast variation in standards-based terminology implementations. Value-based payment programs rely on standardized implementations of standards-based data that generate eCQM data to be able to benchmark scores effectively, and administer value-based payments accordingly. In the current ecosystem, eCQM data and their underlying standards-based encoded clinical data may not be implemented in a standardized way, and therefore the ability to increase value, and enhance population health, may be hindered.

For official eCQMs endorsed by CMS, regular updates occur at least annually, and sometimes two to three times per year. These updates to eCQM definitions may result in changes to measure logic or to the official sets of included and excluded codes in the standards-based terminologies (i.e. value set vocabularies). In previous work, we found that clinics often lag behind in implementing the most updated, and accurate, versions of official eCQM as outlined by value set specifications. When older and newer versions of eCQMs were implemented against the same clinical data, we found changes in measurement of quality of up to 5% difference is overall performance score, and up to 28% difference in the number of patients included in a measure's denominator. (ref_ms10) Similarly, in other work, we showed that implementations of the same eCQM using distinct value set specifications also led to variations in the calculated prevalence of patients at risk for key conditions, and in some cases led to variations in CQM performance percentages.(ref_ms11)

Proposed Study to Evaluate the Impact of Solor

Purpose

Solor provides an easier way to verify that value sets are up to date and covered. Solor can also suggest and add additional codes based on Solor concepts to value set specifications. In this study, our objective is to use Solor to identify codes from eCQM value sets, to better understand the usage of these codes against clinical data, and to assess the impact of the pre and post Solor codes on eCQM performance.

Methods

First, we will identify differences in the coverage of vocabulary specifications – unique identifiers, concepts, code groups, and coding systems – between what is covered in VSAC value sets and what is intended to be covered in value sets according to Solor to define global concepts in measures. After this, we will query clinical data at xxx to determine the frequency of patients for whom the new Solor codes are used. Finally, we will implement the measures in a quality measure calculation registry and CDS environement to estimate the performance differences before and after Solor's mapping of non-covered value set codes.

Evaluation

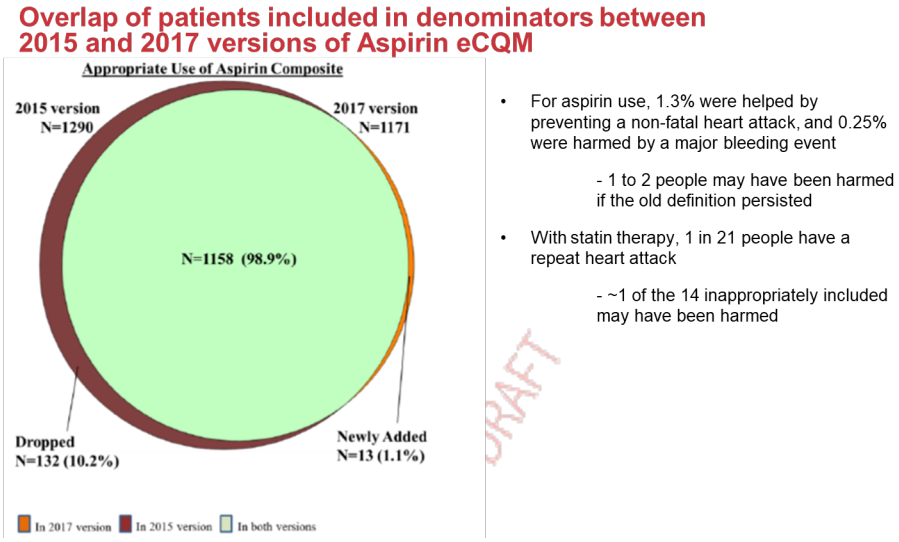
We will compute the frequency of patients who use any of the codes contained in CQM value sets, stratified by measure. We will compare the change in frequencies before and after Solor's addition of equivalebt codes. We will use Fisher's exact test to compare aggregate-CQM performance rates between the original versions of measures and the versions of measures after Solor value sets are implemented. We will use the Jaccard similarity index to assess the similarity between the patients included in the original versions

of the measures, and the versions including complete value set coverage. Number Needed to Treat and Number Needed to Harm Statistics can be used to calculate the potential harms avoided, and harms causes based on pre and post Solor encoded data.

Goals

1. Assess VSAC value sets before and after the use of Solor
2. Understand the frequency of patients that are impacted by newly added Solor value set concepts.
3. Understand how increasing value set code coverage impacts CQM performance estimates, and the patients included in measure populations, and implications on population health.

Example Results**s**



Assuming that the Solor value set specifications of a measure represent “perfect” inclusion, then every newly included patient can be thought of as needing the evidence-based therapy (such as aspirin for secondary prevention of heart attacks) in order to avoid bad outcomes. Under the same assumption, every dropped patient between value set versions of a measure can be thought of as avoiding potential harm caused by the promoted therapy. For aspirin use, Number-Needed-to- Treat (NNT) statistics show that of patients with known cardiovascular risk who took aspirin, 1.3% were helped by preventing a non-fatal heart attack, and 0.25% were harmed by a major bleeding event.26, 27 In xxx study, 121 (92%) of the patients dropped in the Solor version of the Aspirin measure were also taking an anticoagulant medication, so the Number-Needed-to-Harm (NNH) statistic for this subset of patients is likely much higher, and for these clinics, 1 to 2 people may have been harmed if the pre-Solor definition persisted, as Hansen et. al showed that patients with combinations of aspirin, warfarin, and clopidogrel are associated with up to a three-fold higher risk of bleeding for patients on dual therapy and triple therapy.28With another measure for statin therapy, 1 in 21 people have a repeat heart attack, stroke or death avoided, so even 10 missed people have significant risk of events. Similarly, 10% are harmed by muscle damage or pain, or ~1 of the 14 inappropriately included.29 Even in this small study, failure to include or exclude patients could have led to real harm. With eCQM implementation and QI infrastructure increasing, the problem of having, and using, antiquated CQM versions of value sets could have significant potential negative impact on population health by not avoiding events, and avoiding harms for patients.

4. Language

Language is used to describe identified components. While the initial focus of Solor will be to use the English language versions from the foundational coding systems, support for other languages will be included as a part of Solor.

4.1. Language Layer Concerns

4.1.1. Language

SNOMED CT, LOINC, and RxNorm as well as other coding systems have various ways of representing language.

SNOMED CT uses a combination of tables to represent language. The Description table in SNOMED CT includes one Fully Specified Name and at least one synonym for each language. The Preferred and Acceptable Synonyms per language are then specified in a Language Reference Set. Any additional synonyms, other than the preferred, would be identifed as acceptable.

RxNorm identifies language in the RXNCONSO file in the STR field. The language of the name is specified in the LAT field and the source the name comes from is represented in the STT field. The names in the RXNCONSO are not unique as the same name can come from multiple sources.

LOINC has names spread across multiple fields with the Fully Specified Name constructed as a concatenation of the six parts. It also contains a Long Common Name and a Short Name.

Descriptions within a coding system can span multiple languages. For example, "deja vu" exists in both the French and English languages as it is the description used to describe the memory finding.

4.1.2. Dialect

Merriam-Webster's Dictionary defines a dialect as "a regional variety of language distinguished by features of vocabulary, grammar, and pronunciation from other regional varieties and constituting together with them a single language". Two common differences between dialects deal with spelling variants and phrases that have alternate meanings. An example of a spelling variant would be "Anesthetic" in the US dialect versus "Anaesthetic" in the British dialect. The same word in one dialect can mean have a different meaning in another, for example "napkin" in the US is used to describe a piece of cloth or paper used to wipe the hands and mouth at a table while in the UK it is used to describe a diaper.

4.1.3. Interface Terminology

Structured, standardized terms that uniquely identify a concept need to be associated with more natural common phrases preferred by end-users.

4.2. Cross Cutting Concerns

4.2.1. Understandability, Reproducibility, and Utility

The language used to describe a component must be concordant with the underlying semantics of the object being identified. Therefore, there needs to be guidelines in place to ensure only correct terms are associated with an object in Solor.

Having a consistent naming convention defined will assist with textual queries to identify duplicates when concepts are primitive and not able to be fully defined using relationships within Solor. Having a consistent way of representing Fully Specified Names will alleviate the issue of users creating duplicate concepts like "Disorder of immune function" and "Immune function disorder".

Consistent naming is also important to support effective retrieval. For example, the SNOMED CT concept 386560004 |Glasgow coma score finding (finding)| has 13 children all with the string Glasgow coma scale instead of Glasgow coma score.

Another common issue is to add a synonym to a concept that is more specific than the concept itself. A concept should only have synonyms that accurately represent a concept and not any of its children. If a synonym has a more specific meaning, a new concept should be created.

4.2.2. Language Query Requirements

For a search engine to retrieve meaningful results, it must be able to understand common usages of everyday jargon, similar to how synonyms are used to help broaden the way to express the same word. This section will describe several strategies used to help with a query.

Word variants – Similar to synonyms, word variants are used to express the same word. While synonyms are explicitely created as a term to describe a concept (for example, SNOMED's "Heart attack" and "Myocardial infarction"), word variants are utilized during searching to assist in finding the correct concept, rather than explicitely creating the term. Hypothetical example - if "kidney failure" is a term created for a concept, a word variant of 'renal' could be created for 'kidney'. Instead of explicitely creating a separate term of "renal failure", this word variant could be utilized during searching to find all concepts that has the explicit term of "kidney failre" when a user enters "renal failure" by replacing "renal" with "kidney". This would create the burden of creating all possible variant termss for a given word.

Misspellings – Certain terms are more commonly misspelled when searching over healthcare descriptions. The ability for a search mechanism to recognize them and to search over both the correct and incorrect spellings will help to identify the correct concept. For example, perineal vs peroneal, aphagia vs aphasia

Word order – terms can be combined in different ways to mean the same synonym. The ability to search over a term in varying order of phrases is important. For example, Disorder of the eye vs Eye disorder.

Components query – importance of this searching strategy comes into play when a certain focus is desired for the search result. For example, in LOINC, there may be circumstances where a certain axis is desired for the search. Simlarly, certain hierarchy may be desired when searching in SNOMED. For example, "cold" is a synonym of common cold (a disorder) in SNOMED, and also exists as "cold sensation quality" (a qualifier value). By allowing users to limit the search criteria (disorder vs. qualifier), the most appropriate query result will be returned to the user.

Active and inactive – concepts and terms will come in and out of use over time. This is often indicated by an active/inactive designation. In order to properly return concepts/terms that are active, query parameter must contain a parameter to designate if the query result should/shot not return active and inactive concepts or terms.

Regular Expressions - regular express or regex is a sequence of characters that defines a search pattern. This pattern would allow a user to retrieve results based on a certain pattern. For example "a|b*" would return all "a", "b" and other b's that fits the pattern such as "bb", "bbb", "bab", etc. Since the depth and breadth of regular expression is beyond the scope of this document, various syntaxes, usage and explanation can be found in many resources such as https://regexr.com/.

Grouping Results by hierarchy - this search requirement can be thought of as a complement to "components query". After casting a wide net, results could be a bag of various terms (common cold vs feeling

cold) that may be cumbersome for users to sift through if it is not organized in an orderly fashion. Therefore, if results are placed together in a logical grouping, it would assist the user in finding the appropriate query result. For example, in SNOMED, it may be worthwhile to group results by hierarchy (disorder vs. procedures) to allow an user to look for a result in a desired domain or in RxNorm where all Ingredient results are grouped separately from Semantic Clincal Drug results.

5. Definitional

5.1. Introduction

We, as humans, are able to express individual concepts and expand on just the name of the concept, because we are able to associate certain characteristics to individual concepts to further describe itself, as well as establishing connections or relationships between concepts. Once concepts have a meaning beyond just a name, and have relationships to other meaningful concepts, then reasoning can occur.

While the purpose of Language is to provide necessary vocabulary to express various domains of medicine, the purpose of the Definitions are to expand on just a name to go with an idea - single concepts with a name by itself adds no value. In other words, a system must be capable of capturing not just the words of the concept, but also provide a mechanism to express characteristics of that concept (it has the color white [assuming color and white are already defined]) and relationships between concepts (it is a beta-blocker). Additionally, once relationships of meaningful concepts are established, reasoning can occur (beta-blocker is used to treat high blood pressure, therefore, it treats high blood pressure).

Once a system is able to 'digest' the meaning of concepts, it can begin to utilize logic to conduct reasoning. In the field of computer science, the study of Description Logic is to represent the domain, then using these concepts to specify properties of objects and individuals occuring in the domain. Additionally, another feature of Description Logic is the capability to conduct reasoning on represented knowledge[1]. The goal of this section is to first provide a primer to those who may not be fully immersed in the study of description logic, then an introduction to Solor designs to allow for a robust representation and relationships of concepts, followed by topics or concerns from Solor developers and contributors that necessitate a discussion on why these concerns could affect a system to conduct reasoning properly or successfully.

5.2. Description Logic Primer

5.2.1. Description Logic

Description Logics (DL) consist of a family of formal knowledge representation language that implements mathematical logic to support formal expressions, reasoning, and formal proof. It is typically more expressive than propositional logic, which only deals with fixed truth values, which may or may not be true (e.g. "it is raining"), and cannot have variables to represent 'things' (e.g. books or temperature). However, it is less expressive than first-order logic, which assumes the world contains Objects, Relations and Functions, allows variables and can quantify over non-logical objects. The main design principle of Description Logics is its balance between expressivity and computational complexity to suit different applications, with medical ontology modeling being one use case.

DLs provides a way to model the domain by providing three entities: concepts, roles and individual names:

- Concepts - represents sets of individuals
- Roles - relations between concepts
- Names - individual name to represent concepts

Instead of fully describing the state of a domain, as one would with a database, DLs contain axioms, or statements. These axioms capture partial knowledge about the situation that the ontology is describing, and there may be many different states of the world that are consistent with the ontology[2]. With a proper

[1] Badder, F., Nutt, W. (2003). The description logic handbook. Retrieved from https://www.inf.unibz.it/~franconi/dl/course/dlhb/dlhb-02.pdf
[2] Krötzsch, M., Siman#ík, F., Horrocks, I. (2013). A Description Logic Primer. Retrieved from https://arxiv.org/pdf/1201.4089.pdf

modeling of formal semantics, DLs allows humans and computer systems to unambiguously exchange ontologies without losing their meaning, and also provides the capability to infer (reason) additional information from given facts in order compute a conclusion.

5.2.1.1. Definitional Operators

Once statements or axioms are established, a set of syntax and properties are used for further expressivity. Below is an overview of common DL syntax and properties seen in the medical domain, and major design considerations for Solor.

5.2.1.1.1. Conjunction $\wedge$

Example: A $\wedge$ B (A and B)

"A and B" is true only if A is true and B is true. This syntax is also known as intersection.

5.2.1.1.2. Disjointness $\vee$

Example: A $\vee$ B (A or B)

"A or B" is true if A is true, or if B is true, or if both A and B are true. This syntax is also known as union.

5.2.1.1.3. Reflexive roles

Every element is related to itself. For example, X = X

5.2.1.1.4. Role inclusions $\subset$

Example: A $\subset$ B (all A are B)

Role inclusions allow expression of role hierarchies, transitive roles and right identities.

5.2.1.1.5. Necessary axioms

Condition A is said to be necessary for Condition B, if falsity of A guarantees the falsity of B.

In other words, if A then B: B is necessary for A because A cannot be true unless B is true.

Example:

- A = Human being is alive
- B = Air is necessary for human being to breathe
- If "Human being is alive", then "human being has air to breathe"

5.2.1.1.6. Sufficient axioms

Condition A is sufficient for Condition B, if and only if truth of A guarantees the truth of B.

Continuing with the 'necessary' example, air by itself does not guarantee a human being is alive since other factors are required, such as water. In other words, there are several conditions that are required for a human being to be alive, and a sufficient set of these conditions must be present in order for a human being to be alive.

5.2.1.1.7. Defining relationships

Role relationships are represented as existential restrictions. These are used to logically represent a concept by establishing a relationship with other concepts. This will be further elaborated in the Solor Definitional Knowledge chapter of the Definitional section.

5.2.1.1.8. Quantities

Concrete domains are a construct that can define new classes by specifying restrictions on attributes that have literal values (as opposed to relationships to other concepts). The binary operators, equal to, greater than, greater than or equal to, less than, and less than or equal to, can be used in concrete domain expressions, and literal values can be integers, floating point numbers, string literals, and dates. [3]

Concrete domains are used to model quantities in the definition of concepts, such as defining how much ibuprofen may be in a medication tablet. This is further examined in the Topics of Concern chapter of the Definitional Section.

5.2.1.1.9. EL++

Since Solor is based on SNOMED, and SNOMED utilizes a subset of EL++, a brief introduction to this topic appears to be necessary.

According to w3.org[4], EL++ is a lightweight description logic that admits sound and complete reasoning in polytime. It is a syntactic fragment of OWL 1.1 DL. In particular, it shares the semantics of OWL 1.1 DL. The design goals behind EL++ were two-fold:

- capture the expressive power that is used by large-scale ontologies from practical applications
- have polytime reasoning problems, in particular classification and instance checking

As of 2011[5], SNOMED CT content limits itself to a subset of the EL++ formalism, restricting itself to the following operators:

- Top, bottom
- Primitive roles and concepts with asserted parent(s) for each
- Concept definition and conjunction but NOT disjunction or representation of absence
- Role hierarchy but not role composition Domain and range constraints
- Existential but not universal restriction
- A restricted form of role inclusion axiom (xRy ^ ySz => xRz)
- The logic will be extended in the near future to include General Concept Inclusion Axioms

5.2.2. Terminology Layer Exclusions

While computation of language representation is an advanced area, there are certain scenarios that are highly complex and we humans either cannot consistently explain how a machine should interpret or there just is no way to consistently create the content in a manner which a machine can deduce its true meaning. This section describe such scenarios, which would create known undesired effects, or will be handled separately from normal description logic operations.

5.2.2.1. Logical negation

Logical Negation, or "Representation of absence" as it is described throughout this document, is the notion of how to describe something that is not present. The complexity of this topic is described in [refer-

[3] SNOROCKET 2.0 Concurrent Domains and Concurrent Classification
[4] https://www.w3.org/2007/OWL/wiki/EL
[5] https://en.wikipedia.org/wiki/SNOMED_CT

ence definitional_conceptAnalysis.xml, section 1.2.1] and illustrates why it is difficult to represent this notion of something that is not present. Furthermore, it describes how current content in terminologies is inconsistently represented. The topic of absence representation is described in greater detail in [reference absence_representation_requirement.xml].

Therefore, content deemed as "absence" content are identified, which is described in [reference definitional_conceptAnalysis.xml, section 1.2]. The identification of this set of content would allow a system to handle the "absence" computation, when available, in a manner that is separtate and more specialized from typical description logic operations.

5.2.2.2. Measurement

Measurement is a complex topic that can be both addressed through the statement model as well as the terminology knowledge. Since measurements can be much better represented in the statement model and to a much more granular level, measurement is a topic that was determined should not be handeled by normal description logic.

5.3. Solor definitional knowledge

In order to represent the various domains of healthcare, Solor "stands on the shoulder of giants" by building on existing content from SNOMED, LOINC, and RxNorm. The purpose of this section is to discuss the categories within Solor where concepts reside, as well as the relationships used to define the concepts. Ultimately, the goal of this representation of concepts and their relationships would support the description logic component of Solor.

5.3.1. Top level categories

In SNOMED, various hierarchies (e.g. Procedures, Clinical Findings, etc) are used to store and maintain various concepts. These hierarchies are typically modeled such that higher level concepts (i.e. Disorder of endocrine system) are more generic than lower level concepts, which are more granular (e.g. Type 1 diabetes mellitus). Through a series of Is a relationships, which will be explained in the next section, one could traverse down the hierarchy from a top level generic concept to a very specific concept. In an attempt to gather 'like' concepts, Solor created top level categories of concepts to 'house' concepts that belong to those categories. This section will describe the intent and purpose of these categories.

5.3.1.1. Body structure

Contains both normal and abnormal anatomical structures.

Example of Body structures include Structure of left lower limb (body structure), Anastamosis, Roux-en-y (morphologic abnormality) and Skin xenograft (body structure).

5.3.1.2. Environment or geographical location

This hierarchy contains the types of environments and named locations such as countries.

5.3.1.3. Event

Events are different from procedures in that they are occurrences that impact health or health care.

5.3.1.4. Medication

This hierarchy is comparable to the Pharmaceutical / biologic product in SNOMED CT and used to represent drug products.

5.3.1.5. Object

Natural and man-made objects encountered in the healthcare environment.

5.3.1.6. Organism

An individual entity that exhibits the properties of life.

5.3.1.7. Phenomenon

This Hierarchy contains Observable Entities, Clinical Findings, and Disorders.

5.3.1.8. Procedure

Procedures are actions performed in the provision of health care.

5.3.1.9. Qualifier value

Miscellaneous concepts used to represent the values for definitional relationships in SNOMED CT.

5.3.1.10. Record artifact

The Record artifact hierarchy is used to represent the names of a clinical document or parts of a clinical document.

5.3.1.11. Situation with explicit context

Clinical findings or Procedures that have contextual information applied to them. This can include concepts like Family history and Procedures performed in the past or planned in the future.

5.3.1.12. SNOMED CT Model Component

Concepts and attributes used to create and organize SNOMED CT.

5.3.1.13. Social context

Social conditions and circumstances, for example ethnic groups, life styles, occupations, and religions.

5.3.1.14. Special concept

Inactive and navigational concepts from SNOMED CT.

5.3.1.15. Specimen

Material collected for examination or analysis. Usually from a patient but can also be obtained from other sources, for example a catheter or the environment.

5.3.1.16. Stages and scales

SNOMED CT concepts used to represent stages, grades and scales.

5.3.1.17. Substance

Physical matter from which something is made or which has discrete existence. This hierarchy includes allergens, agents, substances, and materials used to define Medications, Phenomenon, and Procedures.

5.3.2. Relationship types

As alluded in earlier sections, each concept with a name by itself does not provide any additional knowledge to reason. If the concept "Disorder of endocrine system" is placed into a bucket of like concepts, such as "Type 1 diabetes mellitus", without any formal definition, one could not possibly deduce that "Type 1 diabetes mellitus" is a type of endocrine system disorder. Therefore, a series of relationships must be used to properly define the concept so that the concept could contain additional knowledge for reasoning to occur. With the simple example of diabetes, it may seem that one could simply create an "Is A" relationship to relate "Disorder of endocrine system" and "Type 1 diabetes mellitus". Although it may connect these two concepts, there needs to be additional relationship types to further define each concept to provide more context and knowledge such that a concept is further defined and could provide additional knowledge. This section describes the relationship types of Solor and provides example of its usage.

5.3.2.1. Accepted relationship types

5.3.2.1.1. Is a

Definition. Is a Relationships are used to represent a hierarchical parent/child relationship between two concepts. Is a relationships should only be used in the cases of a true parent child relationship between two concepts. Only proximal Is a relationships should be distributed in releases, however they may exist in modeling views. Concepts can have more than one Is a relationship but must have at least one Is a Relationship.

Utility. Is a Relationships give a hierarchical structure to Solor and provide a mechanism to query and retrieve subtype concepts.

Example. Bacterial pneumonia (disorder) has two Is a Relationships, one to Bacterial lower respiratory infection (disorder) and another to Infective pneumonia (disorder).

5.3.2.1.2. Phenomenon relationship types

Table 5.1. Phenomenon Relationship Types Sub-Domain and Range

Attribute	Parent Attribute	Domain	Range
Associated morphology	None	Clinical findings/ Disorder	Morphologically abnormal structure (morphologic abnormality)
Associated with	None	Clinical findings/ Disorder	Clinical finding (finding) OR Procedure (procedure) OR Event (event) OR Organism (organism) OR Substance (substance) OR Physical object (physical object) OR Physical force (physical force)

Attribute	Parent Attribute	Domain	Range
Causative agent	Associated with	Clinical findings/ Disorder	Organism (organism) OR Substance (substance) OR Physical object (physical object) OR Physical force (physical force)
Due to	Associated with	Clinical findings/ Disorder	Clinical finding (finding) OR Procedure (procedure) OR Event (event)
Temporally related to	Associated with	Clinical findings/ Disorder	Clinical finding (finding) OR Procedure (procedure)
Before	Temporally related to	Clinical findings/ Disorder	Procedure (procedure)
During	During AND/OR after	Clinical findings/ Disorder	Procedure (procedure)
After	During AND/OR after	Clinical findings/ Disorder	Clinical finding (finding) OR Procedure (procedure)
Clinical course	None	Clinical findings/ Disorder	Courses (qualifier value)
Characterizes	None	Observable entity	Process (qualifier value) OR Procedure (procedure)
Component	None	Observable entity	Body structure (body structure) OR Organism (organism) OR Substance (substance) OR Specimen (specimen) OR Physical object (physical object) OR Pharmaceutical / biologic product (product) OR Record artifact (record artifact)

Attribute	Parent Attribute	Domain	Range
Direct Site	None	Observable entity	Body structure (body structure) OR Organism (organism) OR Substance (substance) OR Specimen (specimen) OR Physical object (physical object) OR Pharmaceutical / biologic product (product) OR Record artifact (record artifact)
Episodicity	None	Clinical findings/ Disorder	Episodicities (qualifier value)
Finding informer	None	Clinical findings/ Disorder	Performer of method (person) OR Subject of record or other provider of history (person) OR Person with characteristic related to subject of record (person)
Finding method	None	Clinical findings/ Disorder	Procedure (procedure)
Finding site	None	Clinical findings/ Disorder	Anatomical or acquired body structure (body structure)
Has definitional manifestation	None	Clinical findings/ Disorder	Clinical finding (finding)
Has interpretation	None	Clinical findings/ Disorder	Finding values (qualifier value) OR Colors (qualifier value)
Has realization	None	Clinical findings/ Disorder, Observable entity	Process (qualifier value)
Inherent location	None	Observable entity	Body structure (body structure) OR Organism (organism) OR Substance (substance) OR

Attribute	Parent Attribute	Domain	Range
			Specimen (specimen) OR
			Physical object (physical object) OR
			Pharmaceutical / biologic product (product) OR
			Record artifact (record artifact)
Inheres in	None	Observable entity	Body structure (body structure) OR
			Organism (organism) OR
			Substance (substance) OR
			Specimen (specimen) OR
			Physical object (physical object) OR
			Pharmaceutical / biologic product (product) OR
			Record artifact (record artifact) OR
			Person (person)
Interprets	None	Clinical findings/ Disorder	Observable entity (observable entity) OR
			Laboratory procedure (procedure) OR
			Evaluation procedure (procedure)
Occurrence	None	Clinical findings/ Disorder	Periods of life (qualifier value)
Pathological process	None	Clinical findings/ Disorder	Autoimmune (qualifier value) OR
			Infectious process (qualifier value) OR
			Hypersensitivity process (qualifier value) OR
			Pathological developmental process (qualifier value)
Precondition	None	Observable entity	Clinical finding (finding) OR

Attribute	Parent Attribute	Domain	Range
			Precondition value (qualifier value) OR
			Procedure (procedure)
Procedure device	None	Observable entity	Device (physical object)
Property	None	Observable entity	Property of measurement (qualifier value)
Process agent	None	Observable entity	Body structure (body structure) OR
			Organism (organism) OR
			Physical object (physical object) OR
			Pharmaceutical / biologic product (product) OR
			Substance (substance)
Process duration	None	Observable entity	Time frame (qualifier value)
Process output	None	Observable entity	Substance (substance) OR
			Process (qualifier value)
Relative to	None	Observable entity	Body structure (body structure) OR
			Organism (organism) OR
			Substance (substance) OR
			Specimen (specimen) OR
			Physical object (physical object) OR
			Pharmaceutical / biologic product (product) OR
			Record artifact (record artifact)
Relative to part of	None	Observable entity	Body structure (body structure) OR
			Organism (organism) OR
			Substance (substance) OR

Attribute	Parent Attribute	Domain	Range
			Specimen (specimen) OR
			Physical object (physical object) OR
			Pharmaceutical / biologic product (product) OR
			Record artifact (record artifact)
Scale type	None	Observable entity	Quantitative (qualifier value) OR
			Qualitative (qualifier value) OR
			Ordinal value (qualifier value) OR
			Ordinal or quantitative value (qualifier value) OR
			Nominal value (qualifier value) OR
			Narrative value (qualifier value) OR
			Text value (qualifier value)
Severity	None	Clinical findings/ Disorder	Severities (qualifier value)
Technique	None	Observable entity	Technique (qualifier value)
Time Aspect	None	Observable entity	Time frame (qualifier value)
Towards	None	Observable entity	Body structure (body structure) OR
			Organism (organism) OR
			Substance (substance) OR
			Specimen (specimen) OR
			Physical object (physical object) OR
			Pharmaceutical / biologic product (product) OR
			Record artifact (record artifact)

Attribute	Parent Attribute	Domain	Range
Units	None	Observable entity	Unit of measure (qualifier value)
Using device	Procedure device	Observable entity	Device (physical object)

5.3.2.1.2.1. Associated morphology

Definition. Associated morphology is used to define Phenomenon by specifying the morphologic changes that are characteristic of a disease.

Utility. The Associated morphology is useful in identifying the morphologic change associated with a disease. This is usually grouped with a finding site to fully define a disease.

Example. [# ##### ########] has an Associated morphology of Fracture (morphologic abnormality)

5.3.2.1.2.2. Associated with

Definition. The Associated with attribute is used to define Phenomenon to specify an association between two concepts that doesn't explicitly state a causal relationship.

Utility. The Associated with attribute is useful to define higher level concepts that collect the various subtype attributes.

Example. [# #### ## ########## # ### # has an Associated with of Acquired immune deficiency syndrome (disorder)

5.3.2.1.2.2.1. Associated With Subtype Roles

Associated With has three Subtype Attributes: Causative agent, Due to, and Temporally related to.

5.3.2.1.2.2.1.1. Causative agent

Definition. The term disease causative agent usually refers to the biological pathogen that causes a disease, such as a virus, parasite, fungus, or bacterium, or can refer to a toxin or toxic chemical that causes illness. [6]

Utility. The Causative agent attribute is useful to identify the cause of a disease that correctly place the concept in the proper hierarchy and aid in search and retrieval of concepts.

Example. Welding of stainless steel is a well recognised cause of occupational asthma, the chrome in the fume has been shown to be the cause in some challenge tests. Non-stainless steel welding is more problematic as specific causative agents have not been demonstrated, but nevertheless occupational asthma occurs. Probably the best evidence comes from longitudinal studies of apprentice welders. Priapism caused by drug (disorder) has a Causative agent of Drug or medicament (substance).

Welders asthma (disorder) has Causative agent of Welding fume (substance)

5.3.2.1.2.2.1.2. Due to

Definition. Due to is used to relate a Phenomenon directly with its causal Phenomenon, Event or Procedure. If the Phenomenon, Event, or Procedure does not directly cause the disease then the parent attribute of Associated with should be used instead.

[6] https://en.wikipedia.org/wiki/Disease_causative_agent

Utility. Due to is used to define concepts where a Phenonmenon, Event or Procedure directly causes a Phenomenon.

Example. [# ######## ### ## ########] has a Due to relationship of Cerebrovascular accident (disorder)

5.3.2.1.2.2.1.3. Temporally related to

Definition. Temporally related to is used in the Phenomenon hierarchy to specify the clinical entity occurring either before, during or after a Phenomenon or Procedure.

Utility. Temporally related to is a parent attribute that can be used to describe a more general concept that will collect the subtypes of Before or During AND/OR after.

Example. This attribute is currently not used to define a concept in Solor. However the subtype attributes are used.

5.3.2.1.2.2.1.3.1. Temporally related to Subtype Roles

Temporally related to has two subtype attributes, Before and During AND/OR after.

5.3.2.1.2.2.1.3.1.1. Before

Definition. Before is used in the Phenomenon hierarchy to define complications that occur prior to a procedure.

Utility. To define Phenomenon that are complications that occur prior to a procedure the Before attribute is used to represent the procedure that the Phenomenon occurs prior to.

Example. This attribute has not been used to define a concept in Solor.

5.3.2.1.2.2.1.3.1.2. After

Definition. The After attribute is used to define Phenomenon that occur after another Phenomenon or Procedure.

Utility. After indicates a sequence of events and not necessarily a cause. If a cause is implied then a Due to relationship should be used instead.

Example. [# ########## ########## ##] has an After relationship of Testicular ablation (procedure)

5.3.2.1.2.2.1.3.1.3. During

Definition. The During attribute is used to define Phenomenon that occur during another Phenomenon or Procedure.

Utility. During indicates a sequence of events and not necessarily a cause. If a cause is implied then a Due to relationship should be used instead.

Example. [# ####### ###### ####### ##] has a During relationship of Surgical procedure (procedure)

5.3.2.1.2.3. Clinical course

Definition. The Clinical course attribute is used in the Phenomenon hierarchy to represent both the course and onset of a disease.

Utility. Course and onset are two categorizations that are typically used in conjunction with each other though sometimes are considered separately. For example, sudden onset and short-term courses

Example. [# ###### ##### # ##### ### has a Clinical course of Chronic (qualifier value)

5.3.2.1.2.4. Characterizes

Definition. The Characterizes attribute is used in the Phenomenon hierarchy to specify the process the property describes and depends.

Utility. Describe why the role is useful here.

Example. [# ##### ###] has a Characterizes of Cardiac process (qualifier value)

5.3.2.1.2.5. Component

Definition. The Component attribute is used in the Phenomenon hierarchy to specify the numerator of a relational property type.

Utility. Describe why the role is useful here.

Example. Hepatitis antibody radioimmunoassay (procedure) has a Component of Hepatitis antibody (substance)

5.3.2.1.2.6. Direct site

Definition. The Direct site attribute is used in the Phenomenon hierarchy to define the direct and sometimes the indirect entity on which and observation is made. An indirect site is allowed to be specified when a direct observation cannot be made.

Utility. Describe why the role is useful here.

Example. Heart rate measured at systemic artery (observable entity) has a Direct site of Systemic arterial structure (body structure)

5.3.2.1.2.7. Episodicity

Definition. The Episodicity attribute is used in the Phenomenon hierarchy to define the episode of care provided by a healthcare provider.

Utility. Episodicity is not used define the episode of disease experienced by the patient. Episodicity is not currently used to define concepts in Solor, but can be used to define new concepts or post-coordinated expressions as needed.

Example. Arthritis (disorder) with Episodicity = First episode (qualifier value) represents the first time the patient presents to their healthcare provider with arthritis.

5.3.2.1.2.8. Finding informer

Definition. The Finding informer attribute is used to define the entity that informs about the clinical finding.

Utility. Finding informer is used to differentiate patient vs provider determined findings. Finding informer is frequently grouped with Finding method.

Example. Complaining of cough (finding) has a Finding informer of Subject of record or other provider of history (person)

5.3.2.1.2.9. Finding method

Definition. The Finding method attribute is used in the Phenomenon hierarchy to define the way a finding was determined.

Utility. Finding method is usually used in conjunction with the Finding informer attribute.

Example. Finding of pulse taking by auscultation (finding) has a Finding method of Auscultation (procedure)

5.3.2.1.2.10. Finding site

Definition. The Finding site attribute is used in the Phenomenon hierarchy to define the body site affected.

Utility. Describe why the role is useful here.

Example. Cervical lymph node abscess (disorder) has a Finding Site of Cervical lymph node structure (body structure)

5.3.2.1.2.11. Has definitional manifestation

Definition. Retired

5.3.2.1.2.12. Has interpretation

Definition. The Has interpretation attribute is used in the Phenomenon hierarchy to define the judgement of the thing being evaluated or interpreted in the Interprets attribute.

Utility. Has interpretation is grouped together to with Interprets to represent what is being evaluated with its interpretation.

Example. Electrocardiogram normal (finding) has a Has interpretation of Normal (qualifier value)

5.3.2.1.2.13. Has realization

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.14. Inherent location

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.15. Inheres in

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.16. Interprets

Definition. The Interprets attribute is used in the Phenomenon hierarchy to define the entity being evaluated.

Utility. Interprets is grouped together to with Has Interpretation to represent what is being evaluated with its interpretation.

Example. Electrocardiogram normal (finding) has an Interprets of Electrocardiographic procedure (procedure)

5.3.2.1.2.17. Occurrence

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.18. Pathological process

Definition. Pathological processes is used in the Phenomenon hierarchy to define the underlying pathological process of a disorder that is not structural and cannot be represented using the Associated morphology relationship type.

Utility. Describe why the role is useful here.

Example. Lupus hepatitis (disorder) has a Pathological Process of Autoimmune process (qualifier value)

5.3.2.1.2.19. Precondition

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.20. Procedure device

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.21. Property

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.22. Process agent

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.23. Process duration

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.24. Process output

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.25. Relative to

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.26. Relative to part of

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.27. Scale type

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.28. Severity

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.29. Technique

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.30. Time aspect

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.31. Towards

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.32. Units

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.2.33. Using device

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3. Procedure relationship types

Table 5.2. Procedure Relationship Types Sub-Domain and Range

Attribute	Parent Attribute	Sub-Domain	Range
Access	None	Procedure	Surgical access values (qualifier value)
Component	None	Evaluation procedure	Body structure (body structure) OR
			Organism (organism) OR
			Substance (substance) OR
			Specimen (specimen) OR
			Physical object (physical object) OR
			Pharmaceutical / biologic product (product) OR
			Record artifact (record artifact)
Direct substance	None	Procedure	Substance (substance) OR
			Pharmaceutical / biologic product (product)
Has focus	None	Procedure	Clinical finding (finding) OR
			Procedure (procedure)
Has intent	None	Procedure	Intents (nature of procedure values) (qualifier value)
Has specimen	None	Evaluation procedure	Specimen (specimen)
Measurement method	None	Evaluation procedure	Laboratory procedure categorized by method (procedure)
Method	None	Procedure	Action (qualifier value)

Attribute	Parent Attribute	Sub-Domain	Range
Priority	None	Procedure	Priorities (qualifier value)
Procedure device	None	Procedure	Device (physical object)
Direct device	Procedure device	Procedure	Device (physical object)
Indirect device	Procedure device	Procedure	Device (physical object)
Using device	Procedure device	Procedure	Device (physical object)
Using access device	Using device	Procedure	Device (physical object)
Procedure morphology	None	Procedure	Morphologically abnormal structure (morphologic abnormality)
Direct morphology	Procedure morphology	Procedure	Morphologically abnormal structure (morphologic abnormality)
Indirect morphology	Procedure morphology	Procedure	Morphologically abnormal structure (morphologic abnormality)
Procedure site	None	Procedure	Anatomical or acquired body structure (body structure)
Procedure site - Direct	Procedure site	Procedure	Anatomical or acquired body structure (body structure)
Procedure site - Indirect	Procedure site	Procedure	Anatomical or acquired body structure (body structure)
Property	None	Evaluation procedure	Property of measurement (qualifier value)
Recipient category	None	Procedure	Person (person) OR
			Family (social concept) OR
			Community (social concept) OR
			Donor for medical or surgical procedure (social concept) OR
			Group (social concept)
Revision status	None	Procedure	Primary operation (qualifier value)
			Revision - value (qualifier value)

Attribute	Parent Attribute	Sub-Domain	Range
			Part of multistage procedure (qualifier value)
Route of administration	None	Administration of substance via specific route	Route of administration value (qualifier value)
Scale type	None	Evaluation procedure	Quantitative (qualifier value) OR
			Qualitative (qualifier value) OR
			Ordinal value (qualifier value) OR
			Ordinal or quantitative value (qualifier value) OR
			Nominal value (qualifier value) OR
			Narrative (qualifier value) OR
			Text value (qualifier value)
Surgical approach	None	Surgical procedure	Procedural approach (qualifier value)
Time aspect	None	Evaluation procedure	Time frame (qualifier value)
Using energy	None	Procedure	Physical force (physical force)
Using substance	None	Procedure	Substance (substance)

5.3.2.1.3.1. Access

Definition. This attribute describes the route used to access the site of a procedure and used to distinguish open, closed or percutaneous procedures.

Utility. Describe why the role is useful here.

Example. Open removal of foreign body from colon (procedure) has an Access of Open approach - access (qualifier value)

5.3.2.1.3.2. Component

Definition. The Component attribute is used in the Procedure hierarchy to represent what is being observed or measured.

Utility. The Component attribute is used specifically to define Evaluation procedures.

Example. Fluorescent antibody measurement (procedure) has a Component of Antibody (substance)

5.3.2.1.3.3. Direct substance

Definition. The Direct substance attribute is used in the Procedure hierarchy to represent the Substance or Medicine on which the procedure's method directly acts.

Utility. Medications are currently not used to define Procedures, but can be used in Extensions and Post-coordinated expressions.

Example. Intra-amniotic prostaglandin instillation (procedure) has a Direct substance of Prostaglandin (substance)

5.3.2.1.3.4. Has focus

Definition. Has focus is used in the Procedure hierarchy to define the focus of a procedure.

Utility. Describe why the role is useful here.

Example. Viral screening (procedure) has a Has focus of Viral disease (disorder)

5.3.2.1.3.5. Has intent

Definition. The Has intent attribute is used in the Procedure hierarchy to define the intent of a procedure.

Utility. Describe why the role is useful here.

Example. Diagnostic procedure (procedure) has a Has intent of Diagnostic intent (qualifier value)

5.3.2.1.3.6. Has specimen

Definition. Has specimen is used in the Procedure hierarchy to define the type of specimen a measurement or observation is performed.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.7. Measurement method

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.8. Method

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.9. Priority

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.10. Procedure device

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.10.1. Subtype Roles

Procedure device has three subtype roles.

5.3.2.1.3.10.1.1. Direct device

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.10.1.2. Indirect device

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.10.1.3. Using device

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.10.1.3.1. Subtype Roles

Using device has a single subtype role.

5.3.2.1.3.10.1.3.1.1. Using access device

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.11. Procedure morphology

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.11.1. Subtype Roles

Procedure morphology has two subtype roles.

5.3.2.1.3.11.1.1. Direct morphology

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.11.1.2. Indirect morphology

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.12. Procedure Site

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.12.1. Subtype Roles

Procedure Site has two subtype roles.

5.3.2.1.3.12.1.1. Procedure site - Direct

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.12.1.2. Procedure site - Indirect

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.13. Property

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.14. Recipient category

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.15. Revision status

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.16. Route of administration

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.17. Scale type

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.18. Surgical approach

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.19. Time aspect

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.20. Using energy

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.3.21. Using substance

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.4. Body structure relationship types

Table 5.3. Body structure Relationship Types Sub-Domain and Range

Attribute	Parent Attribute	Sub-Domain	Range
All or part of	None	Body structure	Body structure (body structure)

Attribute	Parent Attribute	Sub-Domain	Range
Proper part of	All or part of	Body structure	Body structure (body structure)
Constitutional part of	Proper part of	Body structure	Body structure (body structure)
Regional part of	Proper part of	Body structure	Body structure (body structure)
Lateral half of	Regional part of	Body structure	Body structure (body structure)
Systemic part of	Proper part of	Body structure	Body structure (body structure)
Laterality	None	Body structure	Side (qualifier value)

5.3.2.1.4.1. All or part of

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.4.1.1. All or part of Subtype Roles

All or part of currently has one subtype role, Part of.

5.3.2.1.4.1.1.1. Proper part of

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.4.1.1.1.1. Proper part of Subtype Roles

Part of has three subtype roles that can be further used to define Body structures.

5.3.2.1.4.1.1.1.1.1. Constitutional part of

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.4.1.1.1.1.2. Regional part of

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.4.1.1.1.1.2.1. Regional part of Subtype Roles

Regional part of has one subtype role that can be further used to define Body structures.

5.3.2.1.4.1.1.1.2.1.1. Lateral half of

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.4.1.1.1.3. Systemic part of

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.4.2. Laterality

Definition. Laterality is an attribute used to represent the side of the body to which the body structure belongs. It is only used for body structures that are symmetric to both sides of the body and is not used to represent sidedness.

Utility. Laterality is useful for defining body structures that are symmetric to both sides of the body only. There are currently no attributes used to represent sidedness that could define concepts like Structure of left side of heart (body structure).

Example. Structure of right knee region (body structure) has a Laterality of Right (qualifier value)

5.3.2.1.5. Situation with explicit context relationship types

Table 5.4. Situation with explicit context Relationship Types Sub-Domain and Range

Attribute	Parent Attribute	Sub-Domain	Range
Associated finding	None	Finding with explicit context	Clinical finding (finding) OR
			Event (event)
Associated procedure	None	Procedure with explicit context	Procedure (procedure)
Finding context	None	Finding with explicit context	Finding context value (qualifier value)
Procedure context	None	Procedure with explicit context	Context values for actions (qualifier value)
Subject relationship context	None	Situation with explicit context	Person (person)
Temporal context	None	Situation with explicit context	Temporal context value (qualifier value)

5.3.2.1.5.1. Associated finding

Definition. Associated finding links the Situation with explicit context concept to the related Phenomenon or Event

Utility. The Associated finding attribute is used to link a Phenomenon or Event to the contextual information contained in the Finding context, Subject relationship context and Temporal context attributes.

Example. Family history: Diabetes mellitus (situation) has an Associated finding of Diabetes mellitus (disorder)

5.3.2.1.5.2. Associated procedure

Definition. Associated procedure links the Situation with explicit context concept to the related Procedure

Utility.

The Associated procedure attribute is used to link a Procedure to the contextual information contained in the Procedure context, Subject relationship context and Temporal context attributes.

Example. 183985008 |Renal transplant planned (situation)| has an Associated procedure of Transplant of kidney (procedure)

5.3.2.1.5.3. Finding context

Definition. Finding context is used in the Situation with explicit context hierarchy to represent whether a Phenomenon or Event is known or unknown.

Utility. Finding context is used to define the contextual information about whether a Phenomenon or Event is known or unknown.

Example. Chvostek sign positive (situation) has a Finding context of Known present (qualifier value)

5.3.2.1.5.4. Procedure context

Definition. Procedure context is used in the Situation with explicit context hierarchy to represent the status of a Procedure

Utility. Procedure context is used to define the contextual information about the status of a Procedure.

Example. Hemodialysis procedure done (situation) has a Procedure context of Done (qualifier value)

5.3.2.1.5.5. Subject relationship context

Definition. Subject relationship context is used in the Situation with explicit context hierarchy to represent the relationship of the finding or procedure to the subject of record. This can be the subject of record or someone else.

Utility. Subject relationship context is useful for representing the contextual information regarding who the Procedure, Phenomenon, or Event is about.

Example. History of arthritis (situation) has a Subject relationship context of Subject of record (person). Family history: Alzheimer's disease (situation) has a Subject relationship context of Person in family of subject (person)

5.3.2.1.5.6. Temporal context

Definition. This attribute represents the time of a procedure or finding when used in the Situation with explicit context hierarchy

Utility. Temporal context is useful for representing the contextual information regarding when a Procedure, Phenomenon, or Event occurred.

Example. Hip replacement planned (situation) has a Temporal context of Current or specified time (qualifier value). The concept History of malignant neoplasm (situation) has a Temporal context of In the past (qualifier value)

5.3.2.1.6. Medication relationship types

Table 5.5. Medication Relationship Types Sub-Domain and Range

Attribute	Parent Attribute	Sub-Domain	Range
Has ingredient	None	Medication	Substance (substance)
Has active ingredient	Has ingredient	Medication	Substance (substance)
Has precise active ingredient	Has active ingredient	Medication	Substance (substance)
Has basis of strength substance	None	Medication	Substance (substance)
Has manufactured dose form	None	Medication	Pharmaceutical dose form (dose form)
Has presentation strength denominator unit	None	Medication	Unit of measure (qualifier value)
Has presentation strength denominator value	None	Medication	Number (qualifier value)
Has presentation strength numerator unit	None	Medication	Unit of measure (qualifier value)
Has presentation strength numerator value	None	Medication	Number (qualifier value)
Has concentration strength denominator unit	None	Medication	Unit of measure (qualifier value)
Has concentration strength denominator value	None	Medication	Number (qualifier value)
Has concentration strength numerator unit	None	Medication	Unit of measure (qualifier value)
Has concentration strength numerator value	None	Medication	Number (qualifier value)
Has unit of presentation	None	Medication	Unit of presentation (unit of presentation)
Plays role	None	Medication	Role (role)
Count of active ingredient	None	Medication	Number (qualifier value)
Count of base and modification pair	None	Medication	Number (qualifier value)
Count of base of active ingredient	None	Medication	Number (qualifier value)

5.3.2.1.6.1. Has ingredient

Definition. The Has ingredient attribute allows for the definition of a Substance that can be used as an ingredient for a Medicinal product.

Utility. This attribute is not used to define Medicinal product, but is used as a parent attribute for Has active ingredient. It is considered a grouper attribute for other ingredient attributes like Has active ingredient. It can also be used for querying other ingredient attributes to find any Medicinal product with a specific ingredient regardless of the subtype attribute used.

5.3.2.1.6.1.1. Has ingredient Subtype Roles

Has ingredient has one subtype role to further define Medications.

5.3.2.1.6.1.1.1. Has active ingredient

Definition. Has active ingredient represents the substance that has a therapeutic action

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.1.1.1.1. Has active ingredient Subtype Roles

Has active ingredient has one subtype role to further define Medications.

5.3.2.1.6.1.1.1.1.1. Has precise active ingredient

Definition. Has precise active ingredient represents the most specific substance present in the manufactured dose form

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.2. Has basis of strength substance

Definition. Has basis of strength substance is used to represent an active ingredient or part of the active ingredient that the strength of a product is based on.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.3. Has manufactured dose form

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.4. Has presentation strength denominator unit

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.5. Has presentation strength denominator value

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.6. Has presentation strength numerator unit

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.7. Has presentation strength numerator value

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.8. Has concentration strength denominator unit

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.9. Has concentration strength denominator value

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.10. Has concentration strength numerator unit

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.11. Has concentration strength numerator value

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.12. Has unit of presentation

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.13. Plays role

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.14. Count of base of active ingredient

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.15. Count of active ingredient

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.6.16. Count of base and modification pair

Definition. Insert definition here.

Utility. Describe why the role is useful here.

Example. Give an example of correct use here.

5.3.2.1.7. Substance relationship types

Table 5.6. Substance Relationship Types Sub-Domain and Range

Attribute	Parent Attribute	Sub-Domain	Range
Has disposition	None	Substance	Disposition (disposition)
Is modification of	None	Substance	Substance (substance)

5.3.2.1.7.1. Has disposition

Definition. The Has disposition attribute relates a Substance with the behavior that the substance will exhibit or participate in.

Utility. The Has disposition attribute allows for the definition of behaviors like Antimicrobial (disposition), Decarboxylase (disposition), and Chelating agent (disposition).

Example. Estradiol (substance) has a Has disposition of Estrogen (disposition)

5.3.2.1.7.2. Is modification of

Definition. The Is modification of attribute is used in the Substance hierarchy to define the structural modification of another concept

Utility. The Is modification attribute allows for the definition of Substances that are modifications of other Substances.

Example. Rilmenidine phosphate (substance) has an Is modification of Rilmenidine (substance)

5.3.2.1.8. Specimen relationship types

Table 5.7. Specimen Relationship Types Sub-Domain and Range

Attribute	Parent Attribute	Sub-Domain	Range
Specimen source morphology	None	Specimen	Morphologically abnormal structure (morphologic abnormality)
Specimen source topography	None	Specimen	Anatomical or acquired body structure (body structure)
Specimen source identity	None	Specimen	Person (person) OR
			Family (social concept) OR
			Community (social concept) OR
			Environment (environment) OR
			Physical object (physical object)
Specimen procedure	None	Specimen	Procedure (procedure)
Specimen substance	None	Specimen	Substance (substance)

5.3.2.1.8.1. Specimen source morphology

Definition. Specimen source morphology is used in the Specimen hierarchy to specify the morphologic abnormality from which the specimen was obtained.

Utility. Specimen source morphology is useful for defining the morphologic change the Specimen concept was obtained from.

Example. Swab from abscess of brain (specimen) has a Specimen source morphology of Abscess (morphologic abnormality).

5.3.2.1.8.2. Specimen source topography

Definition. Specimen source topography is used in the Specimen hierarchy to specify the anatomical or acquired body structure from which the specimen was obtained.

Utility. Specimen source morphology is useful for defining the body structure the Specimen concept was obtained from.

Example. Excised breast ectopic tissue sample (specimen) has a Specimen source topography of Breast structure (body structure).

5.3.2.1.8.3. Specimen source identity

Definition. Specimen source identity is used in the Specimen hierarchy to specify the person, group or location from which a specimen was collected.

Utility. Specimen source identity is useful for defining the entity from which a Specimen concepts was obtained.

Example. Environmental swab (specimen) has a Specimen source identity of Environment (environment).

5.3.2.1.8.4. Specimen procedure

Definition. Specimen procedure is used in the Specimen hierarchy to represent the procedure perfomed to obtain the specimen.

Utility. Specimen procedure is useful for defining Specimen concepts that are obtained by performing a procedure.

Example. Specimen from eye obtained by fine needle aspiration biopsy (specimen) has a Specimen procedure of Fine needle aspiration biopsy of eye (procedure).

5.3.2.1.8.5. Specimen substance

Definition. Specimen substance is used in the Specimen hierarchy to specify the type of substance a specimen is comprised.

Utility. Specimen substance is useful for defining the substance the Specimen concept is comprised of.

Example. Arterial blood specimen (specimen) has a Specimen substance of Arterial blood (substance).

5.4. Topics of Concerns

5.4.1. Introduction

In order for a computer system to perform reasoning properly, it must be instructed with very specific steps. However, there exist scenarios that would cause a reasoner to fail or improperly interpret the logic. Therefore, these different groups of concepts must be handled differently with a specific set of instructions. The purpose of this section is to introduce various topics that are of concern within a terminology system.

5.4.2. Content Requiring Special Handling

5.4.2.1. Purpose

The creation of groupings (Assemblage) containing SNOMED CT concepts that require special handling supports the maintenance of this content over time without the necessity of re-reviewing the entire content. Within Solor, these various Assemblages are imported and are properly grouped within the system. Subsequently, a set of rules could be developed and applied to handle each of the cases appropriately.

Concepts may require special handling for a number of reasons:

- Hierarchies may be incorrect and could affect retrieval
- Concepts may require retirement or movement to the “Situation” hierarchy
- Use of concepts may have to be limited

This section outlines the agreed upon rules, the reasoning for applying those rules and provides practical examples of how they are applied.

The concepts identified in this task as either meeting inclusion or exclusion criteria belong to the following categories:

- Concept includes absence
- Concept is not related to the subject of record
- Concept is a compound observations concept
- Concept includes laterality
- Concept is an inverse of a concept
- Concept is a primitive concept that should be fully defined
- Concept is symmetrically modeled

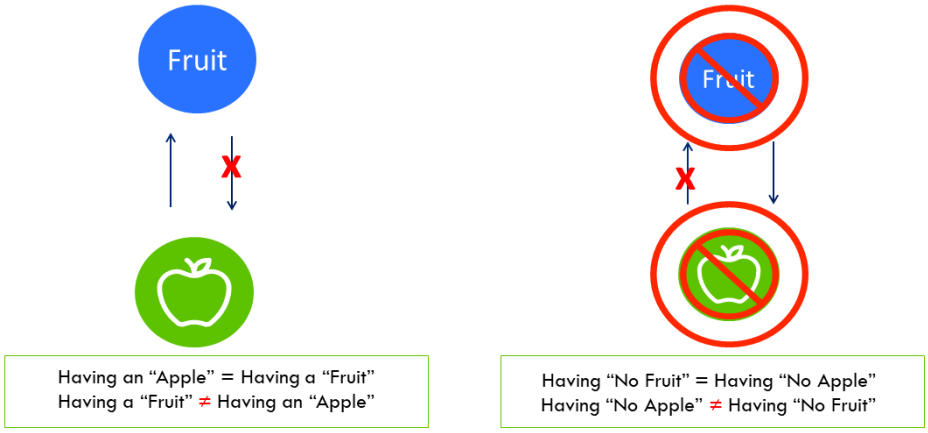
5.4.2.2. Special Handling Categories

5.4.2.2.1. Absence Representation

Absence, where in the strictest sense within the description logic realm, is "NOT" and it means "everything but". If one were to express "not diabetes", it equates to "everything but diabetes".

This is further complicated within SNOMED by the parent-child relationship "Is A". Take the following figure as an example:

Figure 5.1. Effect of Is A on absence



In a hierarchical structure, Is A is a one-way pointer. If B is a child, and A is a parent, that means B "Is A" A. However, one cannot flip that relationship. For example, one can express that one is "having an apple", and by the definition of "Is A", one can assume that one is "having a fruit" (apple is a fruit). However, this directionality cannot be flipped because "having a fruit" does not necessarily mean that one is "having an apple".

In a separate example, what if "No apple" is a child of "No fruit"? If one were having "no apple", it doesn't necessarily mean that one is having "no fruit" (one could very well have other fruits). However, in this

scenario, if one were to express having "no fruit", one could deduce that one is also having "no apple". Note the directionality of the "Is A" in this scenario, which is opposite of the previous example.

Not shown in the figure, but what if "No apple" is a child of "Fruit"?

Although simplistic, this example shows how an absence concept in a hierarchical structure significantly complicates any calculations. Without a way to properly identify if a concept is an 'absence' concept, computational methods could not be applied because the directionality as shown with the apple/fruit example would complicate any calculations.Therefore, it was deemed necessary that such "absent" concepts within SNOMED required identification such that they can be segregated for further special handling.

“Absence” vs. “Affirmation” are two polar opposite paradigms within the SNOMED CT Concept Model. Where “Affirmation” represents a statement that e.g. a finding or a disorder is present, absent states their absence.

However, in SNOMED, the expression of "no diabetes" is a positive assertion that something is not present. This is different than "everything but diabetes". As a result, these two potentially different semantics could lead to confusion and delay if one were to apply computational methods - does "No diabetes" mean "everything but diabetes" or "diabetes is not present"?

Example:

65124004 |Swelling (finding)| vs. 300890009 |Swelling absent (situation)|

“Absence” concepts are generally located in the 243796009 |Situation with explicit context (situation)| hierarchy, where the Context terminological model is consistently applied. Concepts including or implying absence, which are located outside this hierarchy pose challenges for the logical semantic hierarchies they reside in. This study focused only on identifying concepts that are currently not located within the “situation with explicit context” hierarchy. Some of these identified concepts may need to be relocated to the situation hierarchy as a result of this project.

Currently the logical hierarchy for absence concepts remains “upside-down”.

Example:

162298006 |No headache (situation)| is a subtype of 81765008 |No pain (situation)|, but “no headache” does not necessarily mean the patient has no pain.

5.4.2.2.1.1. Approach

The initial task was to evaluate 50,000 concepts and determine their potential membership in one or more of the Assemblages.

For each of the Assemblages for inclusion, word patterns that explicitly or implicitly identify a concept as a member of the Assemblage were developed. As a first automated step, queries using string matching of those patterns or keywords were applied to the following SNOMED CT hierarchies:

1. Clinical Findings
2. Procedures
3. Body Structures

Based on the keywords, terminologists developed a set of rules for each inclusion/exclusion to be applied to each Assemblage.

The sets of concepts that resulted from the initial automated query were then assigned to at least two independent reviewers to confirm or deny Assemblage membership for each concept based on the rule

sets. Disagreements between the reviewers were extracted and analyzed to determine if the rules needed to be adjusted in order to achieve maximum reproducibility. Adjustments included clarifying rules, adding rules or in some cases eliminating ambiguous rules.

Certain concepts such as “Dental referral - child (procedure)” or “Fetal distress affecting management of mother (disorder)”, which were identified as ambiguous to an extent, where inclusion or exclusion from Assemblage membership could not be determined were extracted and added to a separate Assemblage.

5.4.2.2.1.2. Rule Set Considerations

Besides clearly *stated* absence in the SNOMED CT (SCT) Fully Specified Names (FSN), implied absence had to be considered in a number of contexts.

Example: Symptom not changed (finding) vs. Late syphilis with clinical manifestations other than neurosyphilis (disorder)

The first concept clearly states the absence (“NOT changed”), the words “other than” in the second concept implies it.

Rules For Inclusion in “Absence” Assemblage

- FSN states that something about the Subject of Record is “absent”.

  Example: Ankle movement **absent** in *“No ankle movement (finding)”*

- FSN states that something about a procedure is “absent” (Assumption: Procedures are documented, when they are carried out on a Subject of Record).

  Example: Use of contrast media **absent** in *“Magnetic resonance imaging without contrast (procedure)”*

- FSN negates everything “**other**” than what it describes.

  Example: Perception of nothing **other** than light in *“Perceives light only (finding)”*

5.4.2.2.1.3. Queries to Identify Candidate Concepts for Absence Assemblage

Identify content that would need to be evaluated for absence concepts:

- All Situations with a Finding Context = Known Absent
- All Situations with a Procedure Context assigned
- Any concept in Clinical Findings, Procedures, Situation with Explicit Context, and Body Structures hierarchies with strings matching:
  - no
  - not
  - unilateral
  - none
  - without
  - only
  - unable

- inability

The query results were reviewed and either accepted or denied based on the development of a set of rules as described above.

5.4.2.2.1.4. Examples for Inclusion/Exclusion in Absence Assemblage

Keyword: "NO"

SCT ID	FSN	INCLUSION	COMMENT
276038000	No help available (finding)	✓	Something **about the subject of record** is "**absent**"
304327001	No ankle movement (finding)	✓	
164899008	Electrocardiogram: no heart block (finding)	✓	
405491001	Adverse incident resulting in no harmful effect (finding)	✓	
226238008	No beef diet (finding)	⊗	**"No beef" is not about the subject of record** – it is about the **diet**.

Keyword: "NONE" or "NON-X"

SCT ID	FSN	INCLUSION	COMMENT
398278002	Sensory nerve conduction block - none (finding)	✓	Something **about the subject of record** is "**absent**"
369984009	Immature white blood cells - none present (finding)	✓	
50874004	Nonerosive nonspecific gastritis (disorder)	✓	
34390007	Nonexcisional debridement of burn (procedure)	✓	Something **about the procedure** is **"absent"**
445303008	Compression of lymphedema using nonelastic compression device (procedure)	⊗	"**Nonelastic**" does **not** apply to the procedure itself

Keyword: "NOT"

SCT ID	FSN	INCLUSION	COMMENT
288887001	Does not eat (finding)	✓	Something **about the subject of record** is "**absent**"
401169009	Not yet walking (finding)	✓	
248256006	Not getting enough sleep (disorder)	✓	
303863001	Reduction of dislocated joint, not prosthetic (procedure)	✓	Something **about the procedure** is **"absent"**
183052003	Recommendation not to eat (procedure)	⊗	**"Not"** does **not** apply to the procedure itself

Keyword: "UNILATERAL"

SCT ID	FSN	INCLUSION	COMMENT
257840004	External fixation using static unilateral bar (procedure)	✓	Unilateral means that something about the concept is "**absent**" from one side and not the other
253662002	Double aortic arch with unilateral atresia (disorder)	✓	
715905006	Unilateral polymicrogyria (disorder)	✓	

Keyword: "WITHOUT"

SCT ID	FSN	INCLUSION	COMMENT
41119002	Akinetic seizure without atonia (finding)	✓	Something **about the subject of record** is "**absent**"
448521006	Incontinence without sensory awareness (finding)	✓	
400081000	Blister without infection (disorder)	✓	
90084008	Magnetic resonance imaging without contrast (procedure)	✓	Something **about the procedure** is **"absent"**

Keyword: "ONLY"

SCT ID	FSN	INCLUSION	COMMENT
260296003	Perceives light only (finding)	✓	Everything **other** than what the FSN states is negated
170745003	Diabetic on diet only (finding)	✓	
267728009	Blind or low vision - one eye only (disorder)	✓	
173209004	Mediastinoscopy - inspection only (procedure)	✓	
169471006	Progestogen-only pill failure (finding)	⊗	"Progestogen-only" is about the **pill**, not the **subject of record**.

Keyword: "UNABLE"

SCT ID	FSN	INCLUSION	COMMENT
282475008	Unable to run (finding)	✓	Something **about the subject of record** is "**absent**"
288885009	Unable to eat (finding)	✓	

Keyword: "INABILITY"

SCT ID	FSN	INCLUSION	COMMENT
47695004	Inability to cope (finding)	✓	Something **about the subject of record** is "**absent**"
249881006	Inability to imitate tongue movements (finding)	✓	

Keyword: "REJECTED"

Note: "Rejected" was not one of the original search strings but was identified while evaluating the concepts for inclusion.

SCT ID	FSN	INCLUSION	COMMENT
135839007	Sample rejected (finding)	✓	Something **about the subject of record** is "**absent**" Assumption: "Sample/Specimen" was taken from the subject of record.
373880007	Specimen rejected / not processed (finding)	✓	
284348003	Excision of rejected transplanted kidney (procedure)	⊗	"**rejected**" is **not about the procedure** – it is about the **kidney**.

5.4.2.2.2. Concepts Where Patient Is Not Subject of Record

The default context of SNOMED CT concepts as stated in the SNOMED CT Editorial Guide means that, unless stated otherwise within the description or the definition of the concept, clinical findings are occurring to the subject of record (the patient) and procedures are performed on the subject of record (the patient).

The only exceptions are concepts whose description actually contains a specific context (e.g. father smokes), and these are all grouped in the “situation with explicit context” hierarchy. Concepts, where the patient is not the subject of record outside this hierarchy do not adhere to the guidelines. This study did not focus on the concepts within the “situation with explicit context” hierarchy as they have their context already identified using the context attributes.

5.4.2.2.2.1. Approach

The initial task was to evaluate 50,000 concepts and determine their potential membership in one or more of the Assemblages.

For each of the Assemblages for inclusion, word patterns that explicitly or implicitly identify a concept as a member of the Assemblage were developed. As a first automated step, queries using string matching of those patterns or keywords were applied to the following SNOMED CT hierarchies:

1. Clinical Findings
2. Procedures
3. Body Structures

Based on the keywords, terminologists developed a set of rules for each inclusion/exclusion to be applied to each Assemblage.

The sets of concepts that resulted from the initial automated query were then assigned to at least two independent reviewers to confirm or deny Assemblage membership for each concept based on the rule sets. Disagreements between the reviewers were extracted and analyzed to determine if the rules needed to be adjusted in order to achieve maximum reproducibility. Adjustments included clarifying rules, adding rules or in some cases eliminating ambiguous rules.

Certain concepts such as “Dental referral - child (procedure)” or “Fetal distress affecting management of mother (disorder)”, which were identified as ambiguous to an extent, where inclusion or exclusion from Assemblage membership could not be determined were extracted and added to a separate Assemblage.

5.4.2.2.2.2. Rule Set Considerations

Definition for Inclusion: The SNOMED CT concept is about something / someone other than the patient.

Although it can be assumed that all SNOMED CT concepts which are included in this Assemblage are ultimately used to document something in a patient’s record, this particular concept for documentation is **NOT** about the patient.

Rule for Inclusion in “Patient Not Subject of Record” Assemblage:

The concept is about patient’s family, family members, friends or other social contacts, even if it is the patient’s family members, friends or other social contacts.

Examples:

- Findings of relatives surviving (finding)
- Family tension (finding)

5.4.2.2.2.3. Queries to Identify Candidate Concepts for Patient Not Subject of Record Assemblage

Identify content where the subject of record in NOT the patient:

- Subject Relationship Context is not equal to Subject of Record
- Any concept in Clinical Findings, Procedures, Situation with Explicit Context, and Body Structures hierarchies with strings matching:
  - lower(term) like '%father%'
  - or lower(term) like '%mother%'
  - or lower(term) like '%family%'
  - or lower(term) like '%caregiver%'
  - or lower(term) like '%paternal%'
  - or lower(term) like '%maternal%'
  - or lower(term) like '%child%'
  - or lower(term) like '%wife%'
  - or lower(term) like '%husband%'
  - or lower(term) like '%partner%'
  - or lower(term) like '%spouse%'

The query results were reviewed and either accepted or denied based on the development of a set of rules as described above.

5.4.2.2.2.4. Examples for Inclusion/Exclusion in "Patient Not Subject of Record" Assemblage

Examples: "Family", "Family Members", "Friends" or Other "Social Contacts"

SCT ID	FSN	INCLUSION	COMMENT
169944002	Mother has a social worker (finding)	✓	Although it is the patient's "Mother", "Father" or "Friend", the concepts are about the "***Mother***", "***Father***" or "***Friend***" not the patient
185412005	Father made appointment (finding)	✓	
224334009	Friend arrested (finding)	✓	
224139006	Lives with mother (finding)	⊗	Concept is about the **patient**, who lives with the mother – **not** about the **mother**
307101004	Deserted by father (finding)	⊗	Concept is about the **patient**, who was deserted by the father – **not** about the **father**
228302005	Drinks with friends (finding)	⊗	Concept is about the **patient**, who drinks with friends – **not** about the **friends**

5.4.2.2.3. Concepts Including Compound Observation

Compound Observations are the set of concepts within SNOMED CT that involve the combination of more than one observation. While these concepts do not necessarily have issues with them, the fact that they combine multiple concepts into one can cause modeling issues that affect retrieval.

5.4.2.2.3.1. Approach

The initial task was to evaluate 50,000 concepts and determine their potential membership in one or more of the Assemblages.

For each of the Assemblages for inclusion, word patterns that explicitly or implicitly identify a concept as a member of the Assemblage were developed. As a first automated step, queries using string matching of those patterns or keywords were applied to the following SNOMED CT hierarchies:

1. Clinical Findings
2. Procedures
3. Body Structures

Based on the keywords, terminologists developed a set of rules for each inclusion/exclusion to be applied to each Assemblage.

The sets of concepts that resulted from the initial automated query were then assigned to at least two independent reviewers to confirm or deny Assemblage membership for each concept based on the rule sets. Disagreements between the reviewers were extracted and analyzed to determine if the rules needed to be adjusted in order to achieve maximum reproducibility. Adjustments included clarifying rules, adding rules or in some cases eliminating ambiguous rules.

Certain concepts such as “Dental referral - child (procedure)” or “Fetal distress affecting management of mother (disorder)”, which were identified as ambiguous to an extent, where inclusion or exclusion from Assemblage membership could not be determined were extracted and added to a separate Assemblage.

5.4.2.2.3.2. Rule Set Considerations

Definition for Inclusion: The SNOMED CT concept describes more than one observation or procedure

Rules for Inclusion in “Compound Observation” Assemblage:

- Concept is about X ***and*** Y, e.g., *Malaise and fatigue (finding)*
- Concept is about X ***or*** Y, e.g., *Mass in head or neck (finding)*
- Concept is about X ***with*** Y, e.g., *Cough with fever (finding)*
- Concept is about X ***without*** Y, e.g., *Bee sting without reaction (disorder)*
- Concept is about X ***not*** Y, e.g., *Radiographic image not correlated with tumor pathology finding (finding)*
- Concept is about X ***due to*** Y, e.g., *Malnutrition due to child maltreatment (disorder)*
- Concept is about X ***associated with*** Y, e.g., *Limited duction associated with other condition of eye (disorder)*
- Concept is about X ***after*** Y, e.g., *Seizure after head injury (finding)*

5.4.2.2.3.3. Queries to Identify Candidate Concepts for Compound Observation Assemblage

Identify content that are compound observation concepts:

- Any concept in Clinical Findings, Procedures, Situation with Explicit Context, and Body Structures hierarchies with strings matching:
  - and
  - with

- without
- w/o
- due to
- and/or
- after
- resulting
- caused by
- causing
- prior

The query results were reviewed and either accepted or denied based on the development of a set of rules as described above.

5.4.2.2.3.4. Examples for Inclusion/Exclusion in “Compound” Assemblage

Examples “X and Y”

SCT ID	FSN	INCLUSION	COMMENT
417850002	Respiratory tract congestion and cough (disorder)	✓	Concepts describe more than one observation or procedure
247805009	Anxiety and fear (finding)	✓	
16932000	Nausea and vomiting (disorder)	✓	

Examples “X or Y”

SCT ID	FSN	INCLUSION	COMMENT
211506004	Contusion wrist or hand (disorder)	✓	Concepts describe more than one observation or procedure
248477007	Swelling or edema (finding)	✓	
287613009	Middle ear syringing or suction (procedure)	✓	

Examples “X with Y”

SCT ID	FSN	INCLUSION	COMMENT
271503005	Pleural empyema with fistula (disorder)	✓	Concepts describe more than one observation or procedure
120608000	Blister with infection (disorder)	✓	
29532006	Proctoscopy with biopsy (procedure)	✓	
408821002	Lives with partner (finding)	⊗	In these examples, the use of the word “***with***” does **not** constitute the description of more than one observation or procedure
223455001	Assisting with procedure (procedure)	⊗	

Examples “X without Y”

SCT ID	FSN	INCLUSION	COMMENT
448521006	Incontinence without sensory awareness (finding)	✔	Concepts describe more than one observation or procedure
41119002	Akinetic seizure without atonia (finding)	✔	
140921000119102	Ischemic stroke without coma (disorder)	✔	
609242005	Lives in apartment without elevator access (finding)	⊗	In these examples, the use of the word "***without***" does **not** constitute an observation about 2 or more subjects
262312009	Without floor of mouth depressed (finding)	⊗	

5.4.2.2.4. Laterality Concepts

The purpose of the Laterality Assemblages is to identify concepts that are not currently modelled with the correct body structure that utilizes laterality. This only pertains to laterality as currently represented in SNOMED CT, which is used to designate one or both of paired bilaterally symmetrical (or near symmetrical) body structures. It therefore does not apply to sidedness of specific body structures. For example 364006 |Acute left-sided heart failure (disorder)| is not a lateralized disorder since the heart is not a bilaterally symmetrical body structure. For more information on laterality vs sidedness, please see *Choosing Sides. Assigning Laterality as an Attribute in SNOMED® CT.*[7]

5.4.2.2.4.1. Approach

To identify content that would need to be evaluated for laterality concepts:

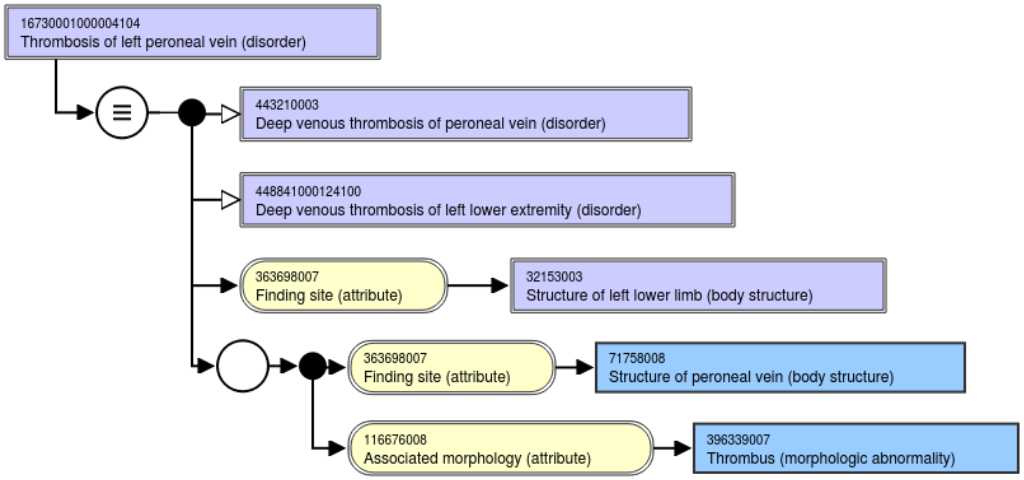
- Set 1: Find all concepts with "right", "left", or "bilateral" in an active term. This identifies all concepts that could potentially represent a lateralized concept based on a term.
- Find all concepts where Set 1 is used as a destinationId for a defining relationship in the Relationship table. This identifies concepts that use the concepts from Set 1 as a value for a defining relationship, which would include both children of concepts in Set 1 and those that use them for other defining relationships.
- Set 2: Find all concepts with a laterality defining attribute. This would identify all Body Structures that use a Laterality Attribute.
- Find all concepts where Set 2 is used as a destinationId for a defining relationship in the Relationship table. This query would identify those concepts that do not have a term with "right", "left", or "bilateral" but do use a Body Structure as a value for a defining relationship.
- Remove from all sets any concepts from the Body Structure hierarchy.

5.4.2.2.4.2. Rule Set Considerations

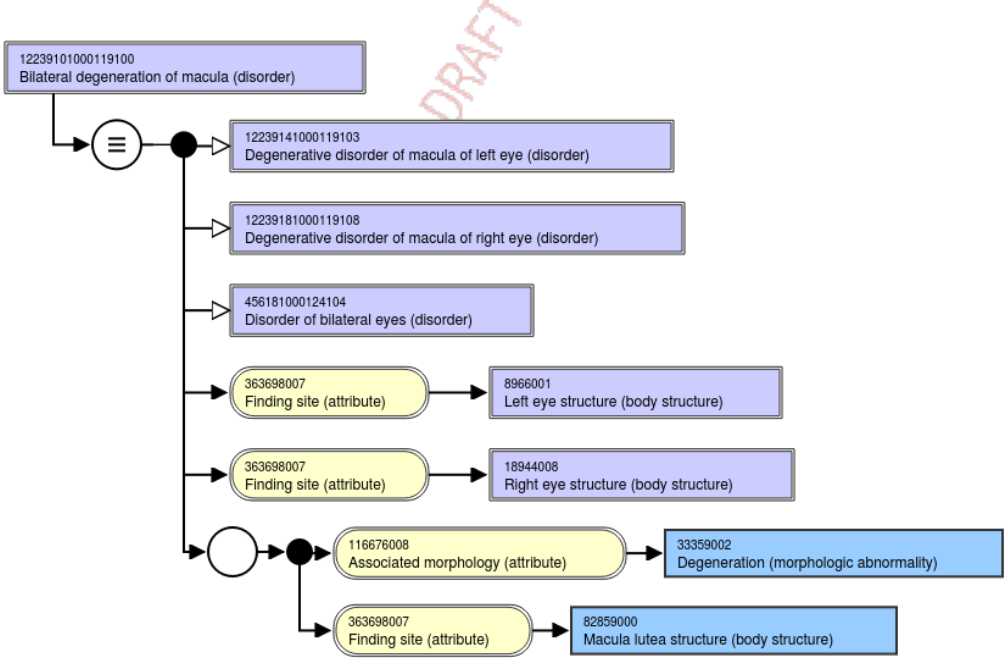
Rules for Inclusion in "Laterality" Assemblage:

1. If the concept being evaluated includes laterality in its FSN and it is not modelled using a Finding Site or Procedure Site, even in cases where there is no current SCT body structure concept with the correct laterality, it will be marked as incorrect. For example, 16730001000004104 |Thrombosis of left peroneal vein (disorder)| should be defined with Structure of left peroneal vein, which does not currently exist in SNOMED CT.

[7] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2244155/?page=1



2. All bilateral concepts were evaluated against the current SNOMED CT modelling guidance, which requires the use of two separate role groups with one representing the right body structure and one representing the left body structure. If the concept is modelled using the bilateral body structure (e.g. 40638003 |Structure of both eyes (body structure)|), it was added to the Laterality Incorrectly Modeled Assemblage. For example, 12239101000119100 |Bilateral degeneration of macula (disorder)| should have two different role groups, one with 721947001 |Structure of macula lutea of left eye (body structure)| and the other with 721945009 |Structure of macula lutea of right eye (body structure)|.



3. If the concept being evaluated included a plural form of a potentially lateralizable body structure in the FSN, the concept would be considered to represent bilaterality and evaluated it as such. For example, 248422003 |Warm hands (finding)| does not specify right, left, or bilateral in the FSN but since it uses the term “hands,” it would be considered as bilateral and evaluated against the current SNOMED CT modelling guidance as stated above.

4. If the concept being evaluated represented sidedness of a non-bilaterally symmetrical body structure, it was added to the Does Not Include Laterality Assemblage. For example, 111283005 |Chronic left-sided heart failure (disorder)| represents heart failure on the left side of the heart.

5. If the concept being evaluated was ambiguous as to whether it represented one side vs both sides, it was placed in the Ambiguous Laterality Assemblage. For example, “lacrimal canaliculi” concepts were considered to be ambiguous since their FSNs did not specify if it was the lacrimal canaliculi of both eyes or the right or left eye.

5.4.2.2.5. Inverse Concepts

The purpose of the Inverse Assemblages is to identify concepts that should have an opposing concept due to a description indicating an opposite or inverse concept, regardless of whether that opposing concept currently exists in SNOMED CT. It is not the purpose to identify and pair opposing concepts. In many cases, the opposing concept does not exist in SNOMED CT and the next iteration of this Assemblage should be to link the two inverse concepts together to identify missing content.

5.4.2.2.5.1. Approach

To identify content where there is an inverse concept:

- Any concept with strings matching a set of search term that would indicate the inverse of another concept

Table 5.8. Inverse Concepts Search Terms

Search Term	Opposing Term
Able to	Unable to
Normal	Abnormal
Present	Absent
Decrease	Increase
Acquired	Congenital
Localized	Generalized
Does	Does not
Benign	Malignant
Complete	Incomplete
Accidental	Intentional
Active	Inactive
Acute	Chronic
Adequate	Inadequate
Open	Closed
Attends	Does not attend
Can	Cannot
(Stable or Stability)	(Unstable or Instability)
Primary	Secondary
Positive	Negative
Major	Minor
Increased	Decreased

Search Term	Opposing Term
Direct	Indirect
Early	Late
Internal	External
Extrinsic	Intrinsic
High	Low
Legal	Illegal
Appropriate	Inappropriate
Increasing	Decreasing
Effective	Ineffective
Insufficient	Sufficient
Irregular	Regular
Loosening	Tightening
Success	(Unsuccess or not success)
Known	unknown
Narrow	Wide
Always	Never
Dependent	Nondependent
Hodgkin	nonhodgkin
Smoker	nonsmoker
Traum	nontraum
Urgent	nonurgent
Venomous	nonvenomous
Old	new
Satisfact	(unsatisfactory or not satisfac)
Use	does not use
Lengthening	Shortening
Near	Far
Infect	noninfect
Inflammatory	noninflammatory
Obstruct	unobstruct
(Loss or Lost)	Gain
Fit	(unfit or not fit)

- Additional keywords were identified during the review process that should be added to future review efforts:

Table 5.9. Inverse Concepts Search Terms

Search Term	Opposing Term
Anteversion	Retroversion

Search Term	Opposing Term
Soft	Firm
Recessive	Dominant
Mature	Immature
Functional	Non-functional

5.4.2.2.5.2. Rule Set Considerations

Rules for Inclusion in "Inverse" Assemblage:

1. Concepts were only considered inverse if a valid opposing concept should exist in SNOMED CT. For example, 56313000|Abnormal placenta affecting management of mother (disorder)| was not considered to be inverse since the opposing concept would be "Normal placenta affecting management of mother" which would not be a valid concept.

2. Anatomical positions and relative locations such as lateral, medial, distal, proximal, etc., were not considered to be inverse.

3. If there is an Open procedure and the only "closed" concept that would ever need to be created would be one that uses only one specific device, these two concepts would be considered as inverse. For example, 179820004 |Open excision of implanted ligament (procedure)| is inverse of 179891009 |Arthroscopic excision of implanted ligament (procedure)| and 265071006 |Open bilateral clipping of fallopian tubes (procedure)| is inverse of 176979002 |Endoscopic bilateral clipping of fallopian tubes (procedure)|.

4. Male and Female were not considered to be inverse.

5.4.2.2.6. Primitive Concepts

The purpose of the Primitive Assemblage is to identify concepts that could be easily fully defined under the current concept model of SNOMED CT. From the SNOMED CT Technical Implementation Guide, a fully defined concept is defined as:

"A Concept is considered to be fully defined if its defining characteristics are sufficient to define it relative to its immediate supertype(s). A concept which is not fully defined is Primitive and this is indicated by the value of the definitionStatusId field.

1. 233604007 |Pneumonia| defining characteristics are specified that effectively distinguish 233604007 |pneumonia| from other lung diseases then it is regarded as a primitive concept.

   If a concept is primitive then the defining characteristics for that concept are incomplete. It is not possible to automatically compute that a concept represented as a postcoordinated combination of several concepts is or is not a subtype of a particular primitive concept.

2. The Concept "lung disease" qualified by 246075003 |causative agent| = 41146007 |bacteria| may be 233604007 |pneumonia| but could also be "bronchitis."

   In contrast if a concept is fully defined it is possible to state that any concept represented as a combination of the same defining characteristics is equivalent to or a subtype of that concept.

3. **Example:** Assume that the Concept53084003 |bacterial pneumonia| is fully defined as 312342009 |infective pneumonia| with 246075003 |causative agent| = 41146007 |bacteria| and that 9861002 |pneumococcus| is a 41146007 |bacteria|. It then follows that the post coordinated representation of 233607000 |pneumococcal pneumonia| as 312342009 |infective pneumonia| with 246075003 |causative agent| = 9861002 |pneumococcus| is computably a subtype of |bacterial pneumonia.|

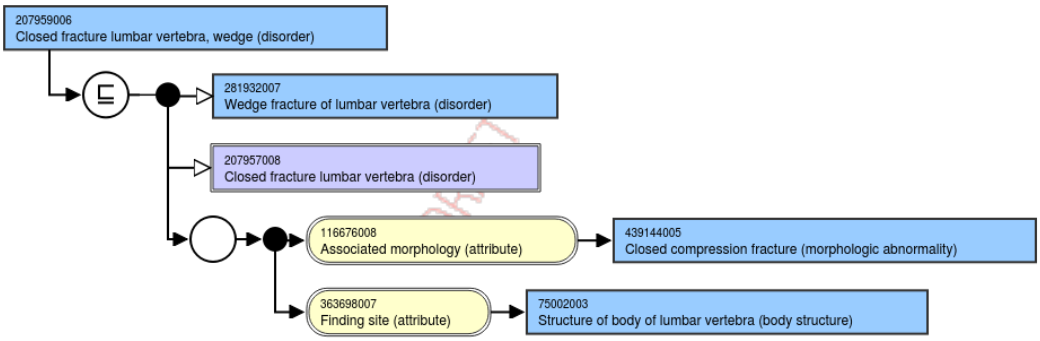
5.4.2.2.6.1. Approach

To identify content that are currently primitive concepts, but may be able to be fully defined:

- Select all concepts that are intermediate primitives, meaning they have both ancestors and descendants that are fully defined but they are primitive
- Select all concepts that are primitive leaf nodes but they have fully defined ancestors

5.4.2.2.6.2. Rule Set Considerations

1. If the evaluated concept can be fully defined within the current SNOMED CT concept model and no changes are required, then it will be placed in the Can Be Fully Defined Assemblage. For example, 201558003 |Reactive arthropathy of shoulder (disorder)| can be changed to fully defined today as there is nothing missing from its definition.
2. If a concept could be fully defined by the addition of a new concept to represent a single parent or by adding a single concept that could be used as a value for a current concept model attribute, the concept will be placed in the Can Be Fully Defined Assemblage. For example, 207959006 |Closed fracture lumbar vertebra, wedge (disorder)| currently cannot be considered as fully defined because an Associated Morphology concept doesn't currently exist to represent a Wedge Fracture. By adding that single concept, the concept will then be able to be fully defined.



3. If the evaluated concept requires a change to the Concept Model, for example, adding a new attribute or changing the range of values an existing Concept Model attribute takes, then it will be placed in the Can Not Be Fully Defined Assemblage. For example, 427252003 |Pain radiating to right side of chest (finding)| cannot be fully defined because there is no concept model attribute to represent that the pain radiated to the right side of the chest. Currently, it only indicates radiating pain and doesn't specify the body structure of chest or that it is on the right side.

5.4.2.2.7. Symmetric Concepts

Symmetry is the complete and consistent representation of the concept model for a particular domain. Symmetry describes the need to eliminate two inconsistency issues that arise in large terminologies regarding completeness: *selection bias* (no ability to select the concept a user is looking for) and *measurement bias* (inconsistent semantic overloading of a parent concept due to the lack of appropriate children). In addition, issues of completeness of hierarchies can also arise from the inconsistent application of the concept model causing concepts to subsume under the inappropriate hierarchy.

We consider modeling of concepts to be "symmetrical" if:

1. Concepts which are opposites of each other (i.e., inverse concepts):
    - Exist in SNOMED and

- Reside in the correct hierarchy under the correct parent concept

  Note: Some keywords that could indicate the need for symmetry are not always reliable, for example:

  - *Traumatic vs. non-traumatic* - concepts without a stated "traumatic" in the FSN are considered non-traumatic by default.
  - *With vs. without* - not every concept that has a "with" or "without" in the FSN needs its opposite, e.g. Diagnostic arthroscopy of elbow with synovial biopsy (procedure) does not need a "…without biopsy".

  Example 1 Inverse Concepts: In this example, it is the two children concepts that are being evaluated for symmetry, not the parents.

Table 5.10. Inverse Concepts #1

299331007 |Knee joint varus deformity (finding)| has two children, which are opposites. Both are present and under the correct parent concept: • 64925008 |Acquired genu varum (disorder)| • 79168008 |Congenital genu varum (disorder)|

Note: Inverse concepts do not necessarily have to reside under the same parent to be considered symmetrically modeled.

Example 2 Inverse Concepts: In the example below, again it is the children concepts that are being evaluated for symmetry and not the parent. In this example, the child concepts reside under different (but correct) parents.

Table 5.11. Inverse Concepts #2

230763008 |Traumatic cerebral edema (disorder)| and 330011000119102 |Non-traumatic cerebral edema (disorder)| are inverse, where: • 230763008 |Traumatic cerebral edema (disorder)| is a child of 127295002 |Traumatic brain injury (disorder)| • 330011000119102 |Non-traumatic cerebral edema (disorder)| is a child of 2032001 |Cerebral edema (disorder)|

Further, for non-inverse concepts, we consider the following concepts to be modeled symmetrically if:

2. Concepts, which have more than one of the same attribute have the same attribute values in the inferred view.
3. Concepts are correctly modeled and in the correct hierarchy.
4. Concepts, which are Leaf Node concepts with one child have the correct Leaf Concept child.
5. Concepts, which are Grades, Scales, Stages, and Scores have no missing concepts and the concepts are consistently modeled.

Note: There can be overlap between 2 and 3, meaning that the same concept can meet the criteria in both 2 and 3 but does not have to.

Some keywords that could indicate the need for symmetry are not always reliable, for example:

- *Traumatic vs. non-traumatic*

  - concepts without a stated “traumatic” in the FSN are considered non-traumatic by default.

- *With vs. without*

  - not every concept that has a “with” or “without” in the FSN needs its opposite, e.g. Diagnostic arthroscopy of elbow with synovial biopsy (procedure) does not need a “…without biopsy”.

5.4.2.2.7.1. Approach

The below approach was used to identify the content to be reviewed to create the Assemblages:

1. **Missing Content – Via Inverse Work**

   - Prior Inverse Assemblage work identified roughly 6,000 concepts that needed to be reviewed to confirm missing opposing concepts. Some examples are shown below.

Table 5.12. Example of missing opposing concepts

Conceptid	Fully Specified Name (FSN)
8587003	Congenital diverticulum of colon (disorder) Missing opposite: Acquired diverticulum of colon (disorder)
8656007	Total traumatic cataract (disorder) Missing opposite: Partial traumatic cataract (disorder)
9027003	Normal pulmonary arterial wedge pressure (finding) Missing opposite: Abnormal pulmonary arterial wedge pressure (finding)
21370008	Tenotomy of abductor of hip, open (procedure) Missing opposite: Tenotomy of abductor of hip, closed (procedure)

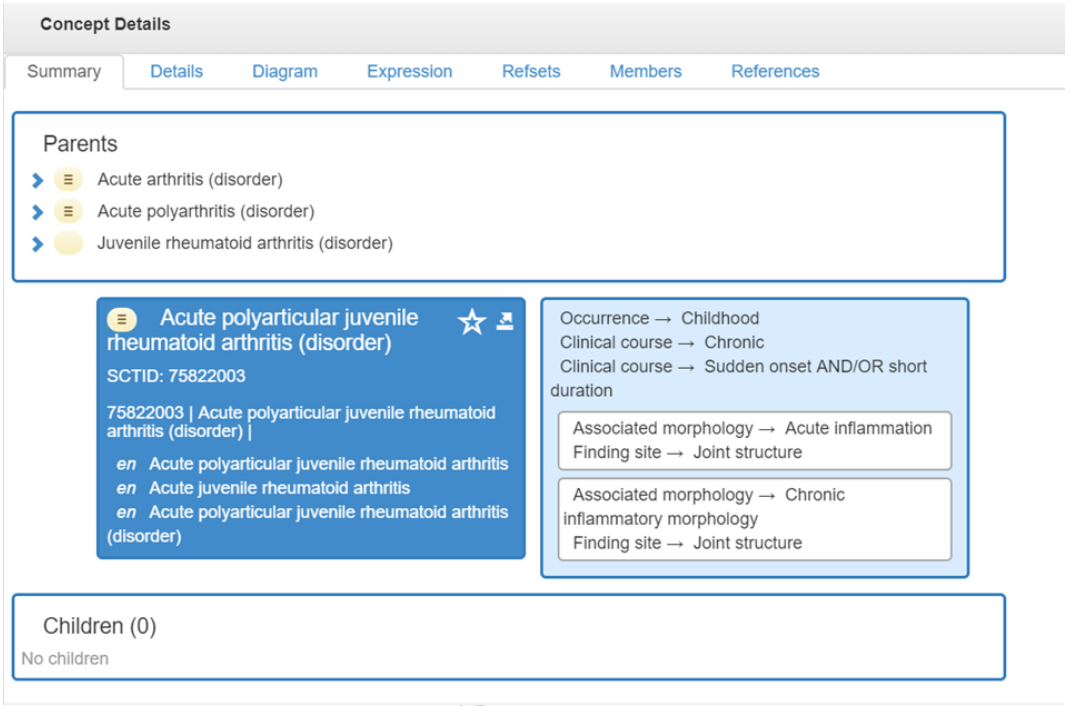
2. **Missing Content – Via Leaf Nodes**

   - Identify all concepts that are parents of a leaf with only one leaf (child).

3. **Content Modeled Inappropriately – Non-Inverse**

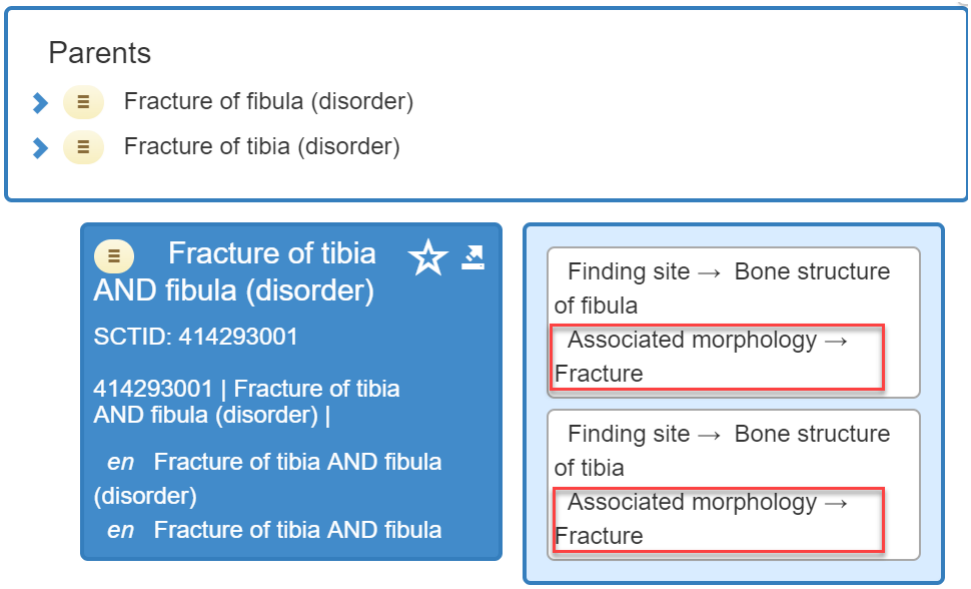
   - Concepts that are inferred where concepts each have more than one of the same Attribute Type

Figure 5.2. Concept with multiple Clinical Course attributes that have different values



- From this set of concepts, remove any Concept that is modeled with more than one of the same Attribute Type and the same Value

Figure 5.3. Concept with multiple Associated morphology attributes and the same values



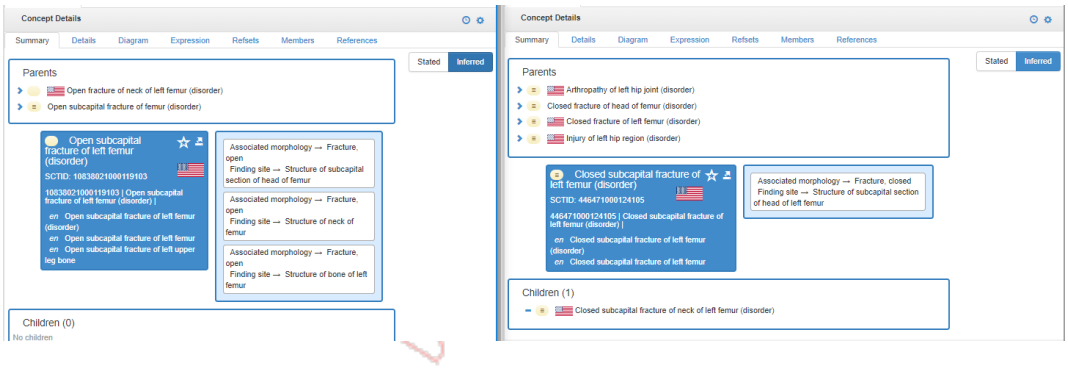
- Also remove from this set of concepts, any Concept with Attributes that are frequently used with different values, like Finding Site or Associated Morphology
- Finally remove from this set of concepts, any Concepts from hierarchies that will not be reviewed (Products, Substances, Qualifier value, Situations, Body structures)
- The remaining set of concepts are considered to potentially have content modeled inappropriately and should be reviewed.

4. **Concept Modeled Inappropriately – Inverse**

- Using concepts that are paired as inverse of each other, we identified those concepts that are modeled differently based on querying the number of defining relationship differences. Not all of the identified modeling differences are symmetrical modeling issues but can be an indicator of them.

Figure 5.4. Example of Inverse Concepts modeled with radical differences

The Open subcapital facture of left femur concept is incorrectly modeled with multiple role groups while the Closed subcapital fracture of left femur is correctly modeled with a single role group



Definition for radically different modeling: Inverse concept and its opposite where the modeling for each is not equivalent for data retrieval and queries in the inferred view.

Table 5.13. Examples for “radically different”

Inverse Concepts	Attributes	Comment
102461004 |Increased intolerance (finding)| 102462006 |Decreased intolerance (finding)|	Interprets -> General clinical state Role Group: [Has interpretation -> decreased Interprets -> Intolerance, function] Outside Role Group: Interprets -> General clinical state	Query for all findings that “interpret the function of intolerance” would not return the 102461004 |Increased intolerance (finding)| concept
164920002 |Electrocardiogram: R wave normal (finding)|	Role group:	Query for all “normalR-wave features” would not return 164920002 |Electrocardiogram:

Inverse Concepts	Attributes	Comment
	[Interprets -> Electrocardiographic procedure Interprets -> R wave feature]	R wave normal (finding)|; there is nothing in the modeling that “has interpretation” of “normal”.
164921003 | Electrocardiographic R wave abnormal (finding)|	Role Group: [Has interpretation -> Abnormal Interprets -> Electrocardiographic procedure] Outside Role group: Interprets -> R wave feature	164921003 | Electrocardiographic R wave abnormal (finding)| ”interprets” both a procedure AND an observable

Table 5.14. Example for “different, but not radically”

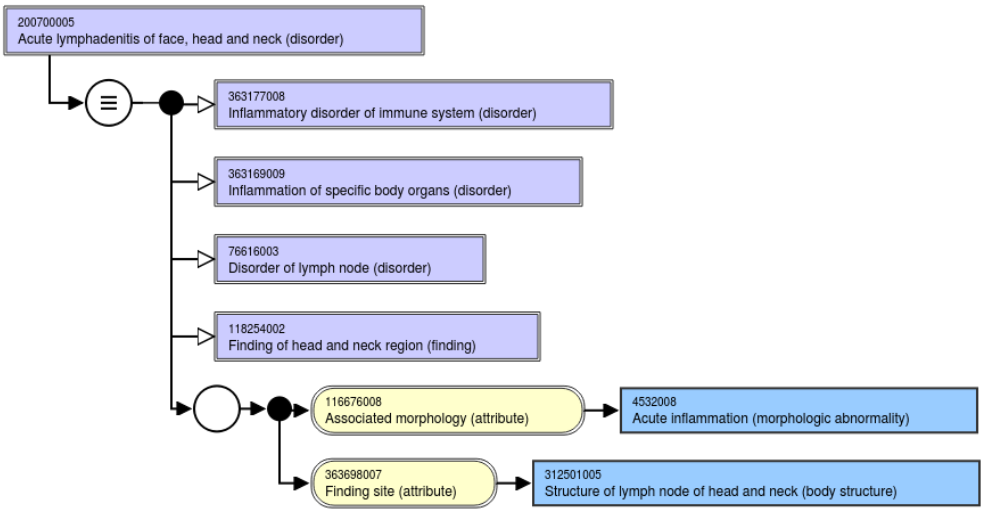
Inverse Concepts	Attributes	Comment
95750004 |Acute blepharitis (disorder)|	Role Group: [Associated morphology -> Acute Inflammation Finding site -> Eyelid structure]	Query for both “acute” or “chronic” inflammation of eyelid would return both concepts.
95751000 |Chronic blepharitis (disorder)|	Role group: [Associated morphology -> Chronic inflammatory morphology Finding site -> Eyelid structure] Outside Role Group: Clinical course -> Chronic	

5. **Identify concepts that contain a common phrase without the appropriate corresponding role.**

   This does not necessarily cause a symmetry issue as the concept may still be placed in the correct hierarchy, but can be used as a query to find a symmetry issue. If the concepts are in the appropriate hierarchy, they are considered to be symmetrical even though they are under-modeled.

   - Find all concepts that have common phrases like “Acute”, “Chronic”, “Acquired”, “Congenital” and that do not have the corresponding attribute.

Figure 5.5. FSN contains "Acute", but does not have a Clinical Course = Acute



6. **Grades, Scales, Stages, and Scores**

   - Review concepts that represent Grades, Scales, Stages and Scores to ensure all are present in the Finding and Disorder hierarchies.

5.4.2.2.7.2. Rules for Evaluating Membership in Assemblages

For the Symmetry project, four Assemblages were created that categorize our efforts as follows:

1. **Symmetric Concepts**

   - A simple Assemblage of concepts that were reviewed and deemed to be in the correct hierarchy and correctly modeled. This includes inverse concepts.

2. **Non-symmetric Concepts**

   - A simple Assemblage of concepts that were reviewed and deemed to be placed in the wrong hierarchy (under an incorrect parent). This includes inverse concepts.

3. **Symmetric Concepts Children Present**

   - A simple Assemblage of parent concepts that had correct children.

4. **Non-symmetric Concepts, Non-existent Children**

   - An Annotation Assemblage with parent concepts that are missing symmetrical children that should exist and any comments on what needs to be done to make them symmetrical.

Notes:

- Overlap can exist between the Symmetric Concept and Symmetric Concepts Children Present Assemblages as well as between Non-symmetric Concepts and Non-symmetric Concepts, Non-existent Children Assemblages. For example for Symmetric Concept Assemblage, we could have “Acquired bone deformity” and “Congenital bone deformity” as inverse child concepts, where both are children of “Bone deformity.” “Congenital bone deformity” could be a parent of a leaf node concept “Congenital defor-

mity of femur." Thus, that concept is a parent concept with correct symmetric children and the parent goes into the Symmetric Concepts Children Present Assemblage.

- Symmetric Concepts Assemblage and Non-symmetric Concepts Assemblage are mutually exclusive.
- Symmetric Concepts Children Present Assemblage and Non-symmetric Concepts, Non-existent Children Assemblage are mutually exclusive.

5.4.2.2.7.3. Rules for Placing Concepts in the Assemblages

Note: For this "symmetric modeling" review, we only consider concepts "incorrectly modeled" if the incorrect modeling pertains to symmetry. If concepts have other – unrelated – modeling issues, they are not referenced as "incorrectly modeled". This includes concepts that are under-modeled, such 162940005 On examination – vocal fremitus increased (finding) and 162941009 On examination – vocal fremitus decreased (finding). Except for the concept name, where the concepts are distinguished by "increased" and "decreased," the concepts are modeled exactly the same, with no attributes included for "increased" and "decreased."

Inverse concepts

- If an inverse concept has an existing opposite concept and it is in the appropriate hierarchy, it was considered Symmetric Correct Modeling and placed in the "Symmetric Concepts" Assemblage.
- If the child is an inverse concept, where its opposite would be included under a different parent but the opposite does not exist or the concept is incorrectly modeled, it was considered Symmetric Incorrect Modeling and placed in the "Non-Symmetric Concepts" Assemblage.

Parents of leaf concepts (concepts with only one child):

- If the child is in the correct hierarchy and is modeled correctly, it was considered Symmetric Correct Modeling and placed in the "Symmetric Concepts Children Present" Assemblage.
- If the child is an inverse concept and its opposite does not exist or the concept is incorrectly modeled, it was considered non-symmetric and placed in the "Non-symmetric Concepts, Non-existent Children" Assemblage.

Note: "correct modeling" only applies to the correct inferred view for this concept as it pertains to symmetry. **If a concept has other modeling problems, as previously noted, it is not marked as "incorrectly modeled".**

5.4.2.2.7.4. Inclusion Criteria by Assemblage

1. **Symmetric Concepts Assemblage**

Table 5.15. Symmetric Concepts Assemblage Inclusion Criteria

Concept Type	Rule	Symmetrical	Example	Comment
Inverse Concepts • Can be **parents** of leaf concepts • Can be **children** of leaf node concepts	Opposite exists **AND** resides in correct hierarchy	#	371350001 | Tolerance related finding (finding)| Is parent of # **102460003 | Decreased tolerance (finding)|**	Since inverse concepts can be parents of leaf concepts, concepts in this Assemblage can also appear in the Symmetric Concepts Children

Concept Type	Rule	Symmetrical	Example	Comment
			# 102459008 | Increased tolerance (finding)|	Present Assemblage
Non-inverse concepts • Can be **parents** of leaf concepts • Can be **children** of leaf concepts	Concepts, which have more than one of the same attribute have the **SAME** attribute values in the inferred view	**#***	**414293001 | Fracture of tibia AND fibula (disorder)|** 116676008 | Associated morphology (attribute)| - 72704001 |Fracture (morphologic abnormality)| occurs twice: one for tibia, one for fibula. Correctly modeled in separate Role Groups.	***Concepts, which fit this rule will be in the "SymmetricConcepts" Assemblage, unless they have other modeling issues that pertain to symmetry
	Concepts **are correctly modeled** and in the **correct hierarchy**	#	**306963008 | Choanal stenosis (disorder)|** Is parent of 34821005 | Congenital stenosis of choanae (disorder)|	
Grades, Scales, Stages, and Scores • Can be inverse concepts • Can be non-inverse concepts	Have no missing concepts **AND** the concepts are consistently modeled	#	**446766005 | Assessment using arthritis impact measurement scale (procedure)|** 304708005 | Arthritis impact measurement scale (assessment scale)| 446478005 | Arthritis impact measurement scale score (observable entity)|	

2. **Nonsymmetric Concepts Assemblage**

Table 5.16. Nonsymmetric Concepts Assemblage Inclusion Criteria

Concept Type	Rule	Symmetrical	Example	Comment
Inverse Concepts • Can be **parents** of leaf concepts • Can be **children** of leaf concepts	Opposite does not exist **OR** resides in wrong hierarchy	#	**432734004 | Congenital asymmetry of breasts (finding)|** Opposite Acquired asymmetry of breasts does not exist but should	Since inverse concepts can be parents of leaf concepts, concepts in this Assemblage can also appear in the Nonsymmetric Concepts Non-Existing Children Assemblage
	Concepts, where the opposites are modeled radically different	#	**102461004 | Increased intolerance (finding)| vs.** **102462006 | Decreased intolerance (finding)|** "Increased" is modeled only with an "interprets" attribute and a "General clinical state" value; "Decreased" is modeled with the same attribute, but additionally with an "interprets" attribute and a "intolerance, function" value and a "has interpretation" attribute with a "decreased" value.	
Non-inverse concepts • Can be **parents** of leaf concepts • Can be **children** of leaf concepts	Concepts **DO** have more than one of the same attribute with **DIFFERENT** values in the inferred view	#	**16024431000119108 |Acute polyarticular juvenile idiopathic arthritis (disorder)|** has 2 "clinical course" attributes, one with a	

Concept Type	Rule	Symmetrical	Example	Comment
			"chronic" and one with a "sudden onset and/or short duration" value.	
Grades, Scales, Stages, and Scores • Can be **inverse** concepts • Can be **non-inverse** concepts	Not all concepts exist **OR** are consistently modeled	#	**396922003 |World Health Organization grade I central nervous system tumor (finding)|** has 2 "interprets" attributes with different values	

3. **Symmetric Concepts Children Present Assemblage**

Table 5.17. Symmetric Concepts Children Present Assemblage Inclusion Criteria

Concept Type	Rule	Symmetrical	Example	Comment
Parents of Leaf Concepts • Can be **inverse** concepts • Can be **non-inverse** concepts	**Children** are in the correct hierarchy **AND** no children missing	#	**168555002 |Plain X-ray skull normal (finding)|** Has child: 168562006 |Plain X-ray nose normal (finding)|, which is inverse. Its opposite 168563001 |Plain X-ray nose abnormal (finding)| exists and is in correct hierarchy	Since parents of leaf concepts can be inverse concepts, concepts in this Assemblage can also appear in the Symmetric Concepts Assemblage

4. **Nonsymmetric Concepts Non-Existing Children Assemblage**

Table 5.18. Nonsymmetric Concepts Non-Existing Children Assemblage Inclusion Criteria

Concept Type	Rule	Symmetrical	Example	Comment
Parents of Leaf Concepts That Should Have Multiple Children that Are Inverse • Can be **inverse** concepts	**Children** are missing	#	**237784000 | Adrenal cyst (disorder)|** Has child: 205744006 |Congenital cyst of adrenal gland (disorder)|, which is inverse. Its opposite	Since parents of leaf concepts can be inverse concepts, concepts in this Assemblage can also appear in the Nonsymmetric Concepts Assemblage

Concept Type	Rule	Symmetrical	Example	Comment
• Can be **non-inverse** concepts			"Acquired cyst of adrenal gland" is not present.	

5.4.2.2.7.5. Other Symmetry Issues

During our review, we identified another symmetry issue, as shown below, which was out of scope for this deliverable, but could possibly be proposed to the IHTSDO as an area of content to be reviewed and edited to achieve consistency.

- Clinical Course vs. Associated Morphology

  Throughout SNOMED, inconsistent modeling using attributes "clinical course" and "associated morphology" exists.

  Example:

  19429009 |Chronic ulcer of skin (disorder)| is modeled using 116676008 |Associated morphology (attribute)| = 405719001 |Chronic ulcer (morphologic abnormality)|

  111422001 |Chronic abscess of breast (disorder)| is modeled using both the |Associated morphology (attribute)| = 79203009 |Chronic abscess (morphologic abnormality)| and the 263502005 |Clinical course (attribute)| = 90734009 |Chronic (qualifier value)|

5.4.2.2.8. Grades, Scales, Stages, and Scores

As a part of the Symmetry Assemblage creation all concepts that represent Grades, Scales, Stages and Scores were evaluated to ensure all are present in the Finding and Disorder hierarchies and are consistently modeled with the appropriate Observable entity, Procedure and Stage and scales hierarchies.

The following analysis of the inconsistent use of Procedures and/or Observable Entities as the value of the "Interprets" Attribute is exploratory and not part of the Assemblage creation.

The Findings and Disorders reviewed were found to use a Procedure 42 times vs. an Observable Entity 352 times. In 41 cases, both a Procedure and Observable Entity were used for the Interprets attribute. 400 of the concepts had no Interprets Attribute at all.

Figure 5.6. Grade concept with an Interprets = Procedure

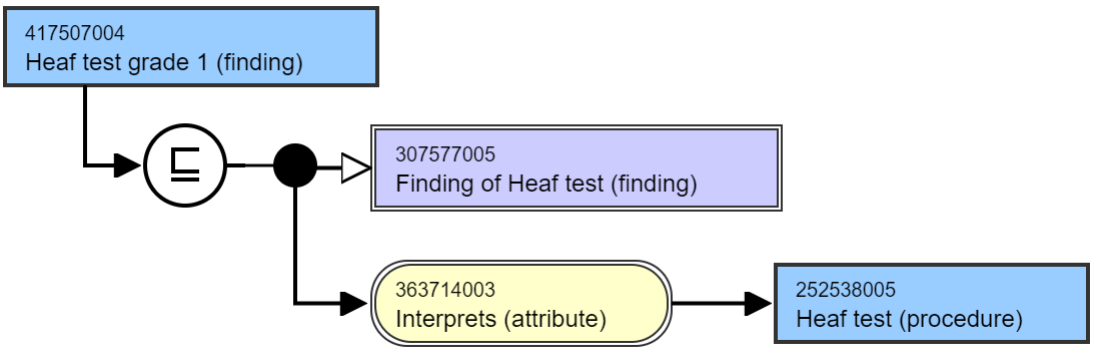


Figure 5.7. Grade concept with an Interprets = Observable Entity

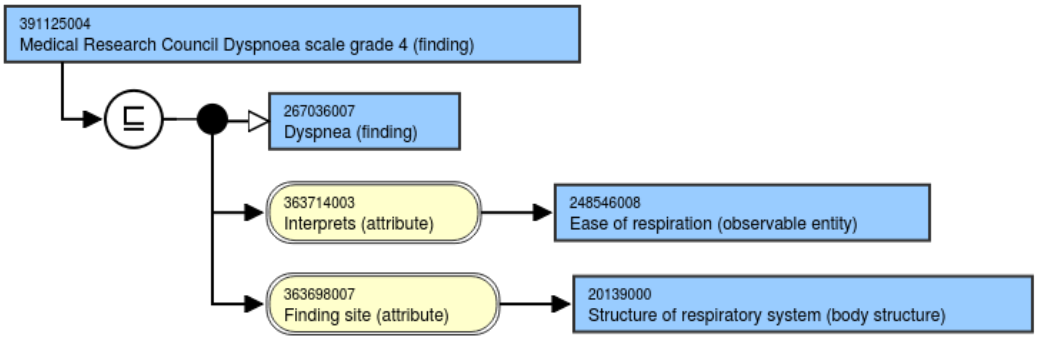


Figure 5.8. Grade Concept with both a Procedure and Observable Entity used for the Interprets Attribute

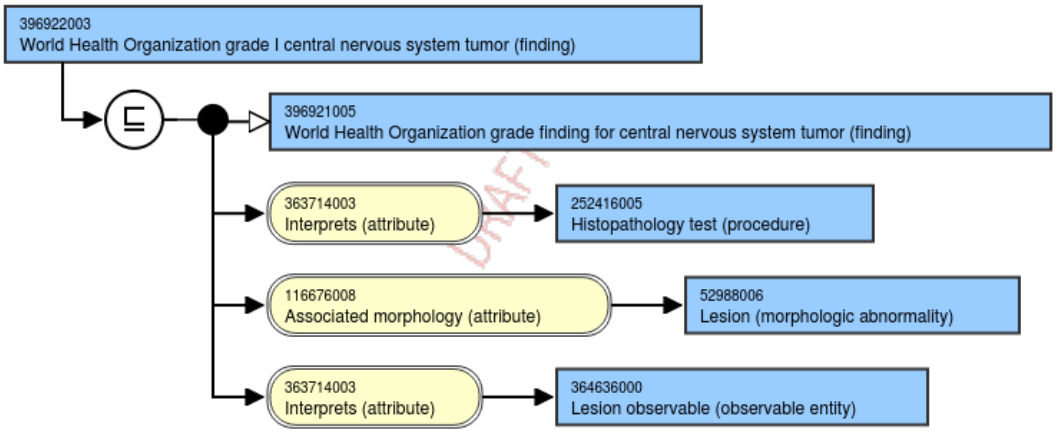
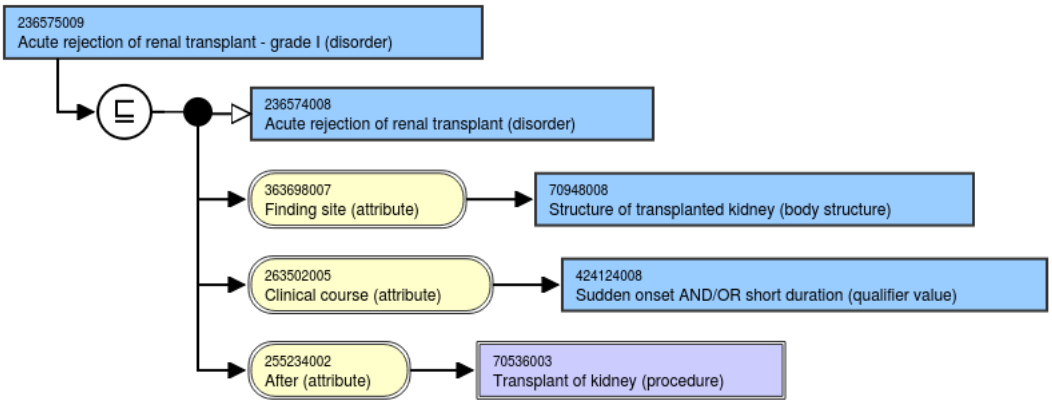


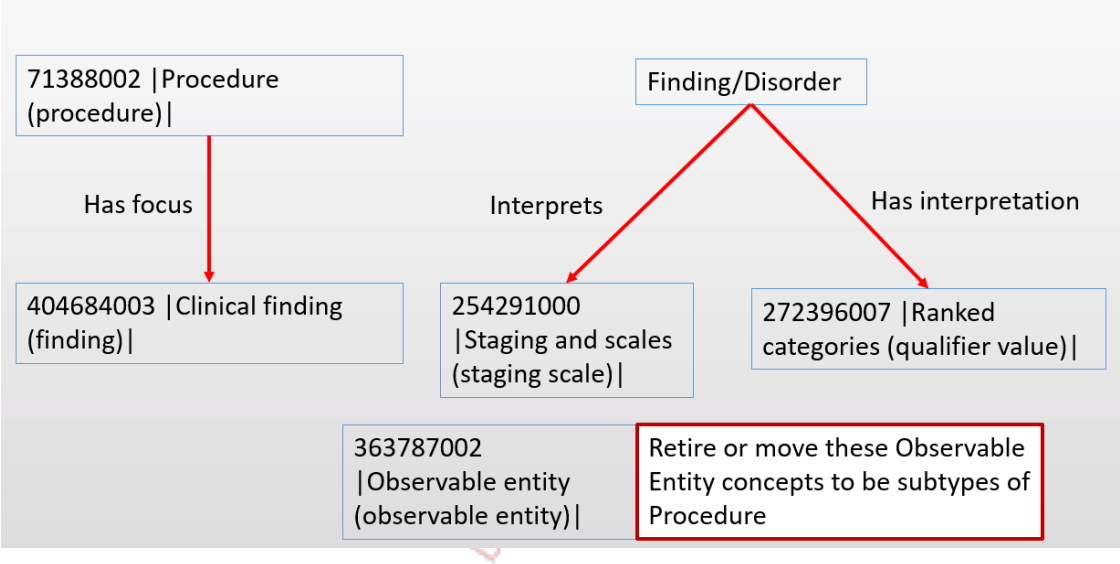
Figure 5.9. Grade with no Interprets Attribute



5.4.2.2.8.1. Potential Changes to Grades, Scales, Stages, and Scores Concepts

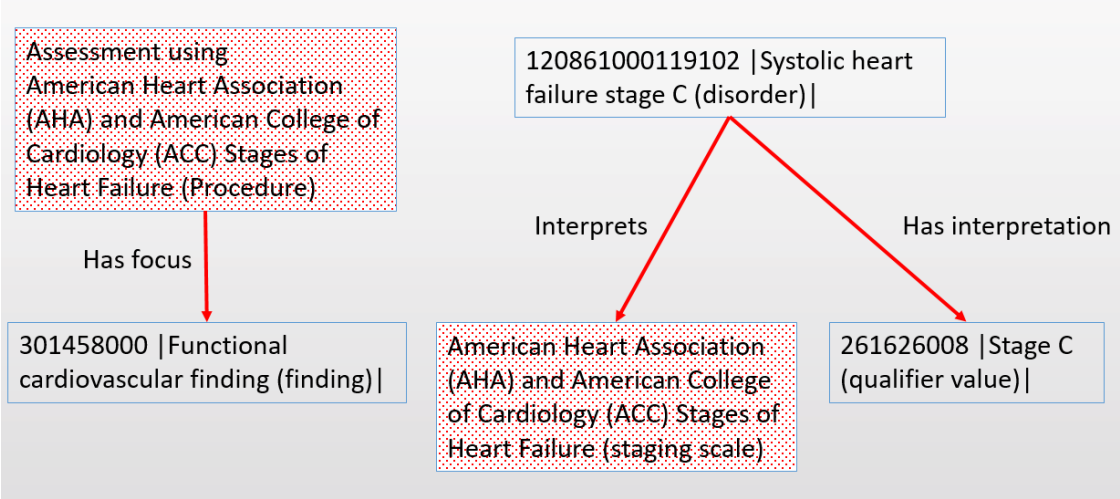
A consistent model needs to be developed and implemented to ensure Grades, Scales, Stages, and Scores concepts are symmetrical. There are many possible options available for creating a consistent concept model for Grades, Scales, Stages, and Scores but the options outlined below can be accomplished without the addition of new concept model attributes. It will require the addition of 254291000 |Staging and scales (staging scale)| as an allowable value for Interprets. A large number of Observable Entity concepts would either need to be retired or remodeled as subtypes in the Procedure hierarchy.

Figure 5.10. Proposed Model for Grades, Scales, Stages, and Scores Concepts



In the example below, the 120861000119102 |Systolic heart failure stage C (disorder)| concept is modeled using an Interprets to a new concept |American Heart Association (AHA) and American College of Cardiology (ACC) Stages of Heart Failure (staging scale)| and a Has interpretation to the existing concept 261626008 |Stage C (qualifier value)|. Separately, a new Procedure concept would need to be created, |Assessment using American Heart Association (AHA) and American College of Cardiology (ACC) Stages of Heart Failure (Procedure)|. Since these stages focus on the functioning of the cardiovascular system, the new procedure concepts would have a Has focus attribute that would link it to the 301458000 |Functional cardiovascular finding (finding)|.

Figure 5.11. Example of Systolic heart failure stage modeled with the new concept model



5.4.3. Concrete Domains

5.4.3.1. Introduction

Concrete domains are used to model concrete properties such as numbers, time intervals, and spatial regions[8]. However, a limitation of description logic is the ability to fully represent concrete values. For example, "male husband is younger than a female spouce" could only be represented by an abstract mean and does not fully capture the semantics [9].

In patient records, there is no shortage of a need to represent these values (weight, temperature, dosages, etc), and SNOMED CT has already begun work on addressing concrete domain. In order to "stand on the shoulders of giants", Solor developers rely heavily on current SNOMED CT work to extend the representation of concrete domains. Therefore, the intent of this discussion is to propose the use of the SNOMED CT model to represent and reason over values like integers in Description Logic.

This initial work focused on medications and evaluating the use of concrete domains to represent not only the product strength, but also the unit of use size. To fully test the feasibility of concrete domains, additional attributes were also added, in order to fully represent all information regarding medications, which will then allow concepts to be fully defined. Thus, this will enable testing the equivalence and subsumption of concepts by the Description Logic classifiers within the tooling.

At the beginning of the project there was no ability to represent numeric attributes of concepts in SNOMED CT, which made machine readability of numeric attributes difficult, prone to error, and left a large portion of Products as primitive concepts. Without the ability to fully represent the numeric properties, equivalence checking and subsumption using the Description Logic classifier is not possible. With the introduction of the new Drug Concept Model in the July 2017 International Release the representation of product strength and units has begun to be modelled. However, this new Drug Concept Model currently does not utilize concrete domains but instead creates the strength numbers as concepts themselves to be used as values for the product strength attributes.

[8] Lutz, C. (1999). IJCAI'99 Proceedings of the 16th international joint conference on Artifical intelligence - Volume 1. Retrieved from https://www.ijcai.org/Proceedings/99-1/Papers/014.pdf

[9] Lutz, Carsten. (2019). The Complexity of Description Logics with Concrete Domains. Retrieved from http://www.cs.man.ac.uk/~ezolin/dl/bib/Complexity_of_Description_Logics_with_concrete_domains_(Lutz_PhD_2002).pdf

5.4.3.2. Approach

By using a lexical search for strings containing integers and textual representation of integers, 10,114 potential Pharmaceutical / Biologic Product concepts were identified, which were modeled with the proposed attributes, including one attribute to represent product strength. To properly represent the numeric information contained in these products, the Australian Medicines Terminology Approach was applied to the International SNOMED CT content.

To speed up the modeling process, already available data around strength and units from NDF-RT through RxNorm RXNSAT relationships that was linked to the SNOMED CT concepts through the RXCUI was used. Technical validation was performed on these values and any incorrect strength or units identified were corrected before using these values to populate the relationships. After loading the new relationships into the terminology editor, further manual review was conducted to verify the relationships and add any missing information.

Using the findings from the drug modeling, the team evaluated other hierarchies that were identified as having potential for modeling concrete domains.

5.4.3.3. Attributes for Representing Medications

Below are attributes that have been added to the medications model to represent concrete domains:

- **Has Basis of Strength Substance (BoSS)** – The substance(s) that correspond to the strength. If strength is not stated, then this attribute is not used. The Has Active Ingredient attribute is still used and grouped together with this attribute.
  - Range: << Substance (substance)
- **Has Product Strength** – The strength of the Has Basis of Strength Substance and is always grouped together.
  - Range: Float 0 to 1,000,000,000
- **Units** – Unit of Measure is always associated with the Strength.
  - Range: <<Unit (qualifier value)
- **Has Unit of Use** – Describes a discrete unit that a product presents in, for example a vial, bag, etc.
  - Range: (<<)Type of drug preparation (qualifier value) and (<<) Unit of drug administration (qualifier value)
- **Unit of Use Size** – Represents the size of the unit of use.
  - Range: Float 0 to 1000000000
- **Unit of Use Quantity** – Represents the packaging quantity.
  - Range: Float 0 to 1000000000

5.4.3.4. Findings

Under the new SNOMED CT International Drug Concept Model, existing concepts will be updated to meet the new modeling guidelines and terming updated to conform to the terming guidelines in the editorial guide. One of the most frequent issues found while modeling the medication attributes was that the Fully Specified Names (FSN) were not completely fully specified or that the values needed to fully define a concept were not available. For example, the common issues seen around FSN's were due to the salt or

dose form not present or not fully defined in the FSN, but modelled with the more specific value in the current Has active ingredient and Has dose form attributes. With the SNOMED CT International review and application of the new modeling guidelines, these FSN's should be corrected and fix the issues found with FSN's.

- *Example:*

  (FSN does not explicitly state that it is an Oral suspension):

  370762006 |Azithromycin 1g/packet oral (product)|

  <<< 392327001 |Oral form azithromycin (product)| :

  127489000 |Has active ingredient (attribute)| = 391805000 |Azithromycin dihydrate (substance)|, 411116001 |Has dose form (attribute)| = 385024007 |Oral suspension (qualifier value)|

Another common issue with fully defining concepts using this proposed model was associated with sugar free, gluten free, preservative free, etc. dose forms. This issue is currently out of scope for the new SNOMED CT International Drug Concept Model and will prevent the concepts that currently exist in SNOMED CT from being fully defined. A potential solution for representing these dose forms and fully defining the drug concepts would be to create concepts in the qualifier value hierarchy for sugar free dose form, gluten free dose form, etc and use a nested relationship to combine it with the other appropriate dose form. This would eliminate the need to create all the possible combinations of dose forms required to support the Drug Concept Model.

- *Example:*

  320108004 |Salbutamol 2mg/5mL sugar free syrup (product)|

  <<< 135639005 |Oral form albuterol (product)| :

  127489000 |Has active ingredient (attribute)| = 48474002 |Albuterol sulfate (substance)|, 411116001 |Has dose form (attribute)| = (385032004 |Syrup (qualifier value)| + XXXXXX|Sugar free dose form (qualifier value)|)

The sections of the SNOMED CT International Drug Concept Model dealing with Grouper, Virtual Medicinal Product (VMP), and Virtual Medicinal Product Form (VMPF) concepts in the Pharmaceutical / biological product hierarchy did not affect the concrete domain work as these concepts do not include product strength as a part of their FSN. However, correcting issues with these concepts will have downstream effects on the modeling of concepts.

The section that was most relevant to the concrete domain work was the Virtual Clinical Drug model. The main differences between the approaches are:

- Strength is not represented as a number in the SNOMED CT International model, but as a conceptid that is a representation of that number.
- The SNOMED CT International model currently has no way to represent ranges of strength (for example radiopharmaceuticals).
- The SNOMED CT International model separates out numerator and denominator for both strength and units whereas this model normalized the strength.

After the testing of concrete domains using the pharmacy model, concepts in findings, procedures and observables were reviewed to determine the feasibility of applying concrete domains to concepts in those hierarchies as well. 3668 concepts were identified that may potentially benefit from the use of concrete domains in these hierarchies.

These concepts mainly fall into 4 categories:

- **Grades/Stages/Scales**

  This category of concepts is least likely to benefit from concrete domains as some grades/stages/scales are alpha-numeric and would more likely fall into a similar model as the SNOMED International Drug Concept Model.

  - *Examples:*

    109970006 |Follicular lymphoma, grade 1|

    112110007 |Glasgow coma scale, 4|

    112241002 |Lymphoma stage III 1|

- **Measurements/Percentiles**

  This category of concepts mirrors the requirements of the Drug Concept Model most closely and would be very similar in that it would require both an attribute for recording the numeric value and another attribute to record the unit. This would also require the ability to capture less than, greater than and equal to which is not currently something supported in the SNOMED CT International Drug Concept Model. Therefore using concrete domains would be a much more suitable solution as it allows for that capture of that information but would require a change to the SNOMED CT Release Format to accommodate these relationships.

  - *Examples:*

    314643009 |Child head circumference < 0.4th centile|

    385303005 |pT3: Tumor more than 5 cm in greatest dimension (anal canal)|

- **Timing/Frequency**

  While these concepts contain numeric values, they may not lend themselves to being captured by concrete domains due to the fact that there are some expressions like “every 12 months”, “once a week”, “five times a week”, etc.

  - *Examples:*

    34259007 |Measurement of glucose 5 hours after glucose challenge for glucose tolerance test|

    416755008 |Cervical smear every 12 months for life|

- **Dosing Number/Episode**

  This would be a small subset of concepts that would be affected but would be a good target for a set of relationships to use for post-coordination instead of adding pre-coordinated concepts to the standard. Making these relationships strictly available through post-coordination and using concrete domains would not require a change to the release format. It would however require existing concepts (less than 100) to be retired in order for all concepts to be aggregated appropriately.

  - *Examples:*

    170425007 |Typhoid and Paratyphoid first dose|

    231499006 |Endogenous depression first episode|

5.4.4. Disjoint Content

5.4.4.1. Introduction

Classes are disjoint if they cannot have common instances. In an ontology, all classes are assumed to have potential overlapping instances unless they are explicitly stated to not have them. Since Solor relies heavily on SNOMED, a discussion of this topic is necessary.

The current modeling of SNOMED CT does not contain any explicit statements stating disjointness, therefore all concepts are considered to have the potential to allow overlapping concepts. For example, there is no formal statement that would prohibit the clinical findings and body structure hierarchies from containing concepts that have parents from both hierarchies even though this should never be the case. With the exception of the physical object and products that currently overlap, the top level primitive hierarchies like clinical findings and body structures should be disjoint.

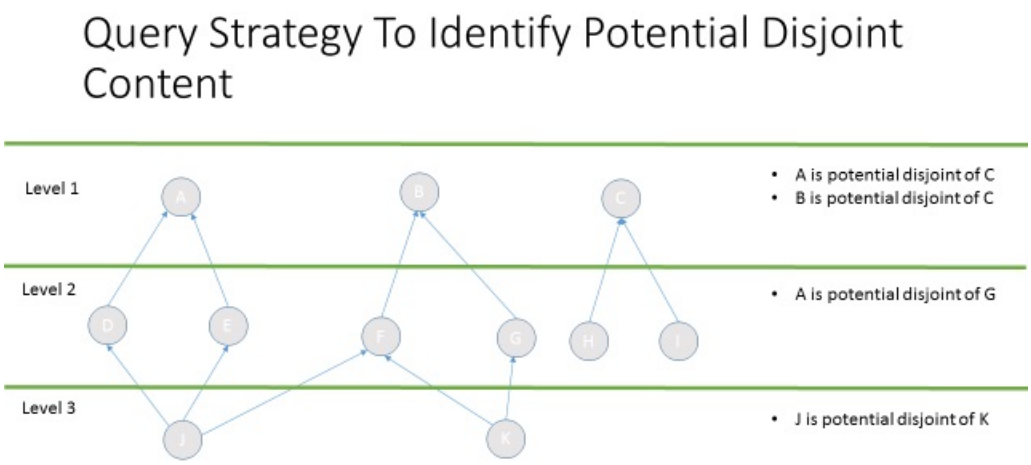
5.4.4.2. Problem

Explicitly stating disjoint content would assist not only in detecting potential modeling errors, but also potentially aid in creating correct post-coordinated expressions. With more extensions to SNOMED CT being created at the National Release Center level and at the local implementations, more rich features are needed to ensure the correct creation of local content. SNOMED CT contains many concepts with similar Fully Specified Names across upper level primitive hierarches that can easily be assigned as a parent to a concept in another upper level primitive hierarchy. For example, “Hematoma” exists in both the disorder and morphologic abnormality hierarchies. If you are modeling a subtype of hematoma in the disorder hierarchy the morphologic abnormality could easily be chosen by a less experienced modeler if the tools used to model do not appropriately specify the hierarchy the parent comes from. Without the disjoint statements explicitly stated, the classifier would not be able to detect this error and a separate Quality Assurance (QA) statement derived from documentation would be needed to prevent this error. Likewise, having explicit disjoint statements can assist in the creation of post-coordinated expressions as they can be queried and used to restrict the allowable parents assigned when using multiple focus concepts.

5.4.4.3. Solution

All top level primitive concepts should be stated as disjoint with the exception of 260787004 |Physical object (physical object)|and 373873005 |Pharmaceutical / biologic product (product)|. A particular focus was placed on primitive hierarchies of substance and body structure. For each hierarchy, all concepts that are currently disjoint from each other beginning at the top of the hierarchy and traversing downward were the focus. This method will identify potential disjoint statements, which were reviewed by clinicians to confirm that they are correct.

Figure 5.12. Query strategy to identify potential disjoint content



5.4.4.4. Results

The US Extension to SNOMED CT was utilized to perform an initial assessment for disjoint statements. While calculating the disjoint statements for the upper level hierarchies, 243796009 |Situation with explicit context (situation)| and 123037004 |Body structure (body structure)| were not being calculated as potentially disjoint. The single concept that was causing them not to be stated as disjoint was 119741000119108 |History of amputation of right lower limb (situation)| due to the fact that it was modeled in the US Extension as having parents in both hierarchies. This issue was reported to the National Library of Medicine and has been corrected in the March 2017 US Extension.

169 disjoint statements were added to the upper level primitive hierarchies to test the feasibility of running a reasoner over them successfully and within a reasonable amount of time using disjoint statements using the minimum number of statements needed.

The tls2_StatedRelationshipsToOwlKRSS_Script_INT.pl from the SNOMED International GitHub registry was utilized to create an OWL file from the March 2017 US Edition release. Utilizing this file within the Protégé 5.2.0 editor and the included HermiT reasoner, the OWL file without disjoint statements was reasoned in 3,015,366 milliseconds. The 169 disjoint statements were then added to the upper level primitive concepts and reasoning over this version took 2,494,176 milliseconds.

The same test was performed using the Snorocket reasoner plugin and achieved the results of 122,438 milliseconds and 54,498 milliseconds respectively. Therefore adding disjoint statements does not increase the time to reason over the OWL version of SNOMED, but actually significantly decreased the amount of time using both reasoners we tested.

An additional 133 concepts were tested for potential disjointness within the substance, body structure, and situation with explicit context hierarchies as these hierarchies are most likely to benefit from the addition of disjoint statements. 13 substance statements, 1193 body structure statements, and 12 situation with explicit context statements were able to be added. These disjoint statements only cover the immediate children for all the hierarchies listed above except for body structures, where a traversal down three levels deep was performed to identify potential disjoint content.

However, adding disjoint statements to these concepts will provide limited benefit for error checking. The body structure and substance hierarchies will have limited use cases for extension and post-coordination

once the redesign is complete. The situation with explicit context hierarchy is one where heavy post-coordination and/or extension will take place, however most of this work will involve assigning a single parent that is a direct subtype of the upper level primitive. A more productive use of resources would be to focus on addressing any modeling issues in these hierarchies and introducing a mechanism for blocking the editing of these concepts without editorial approval. Focusing only on the first level below the upper level primitives in each of these hierarchies would be the best use of resources in the short term until the redesign of the concept model for body structure and substances is complete.

5.4.4.5. Conclusion

Without statements to detect disjoint content, there is a potential for modeling errors, such as modeling incorrect parents for SNOMED CT concepts. This will affect both equivalence detection and content retrieval via the SNOMED CT hierarchies. Adding disjoint content statements to the SNOMED CT definitions will assist both SNOMED CT International and extension content creators by providing built-in QA to prevent errors in assigning parents. The creation of these statements should focus on the upper level primitive hierarchies and their direct descendants. Assigning further statements may become more useful once the redesign of the concept model for the various hierarchies is complete.

5.4.5. Meronomy / Partonomy

5.4.5.1. Introduction

Meronomy / Partonomy is a type of hierarchy that deals with part-whole relationships. *Part-of* Relationships are:

- Transitive – a part of a part is also a part of the whole, example below:
  - Atrioventricular junction: Part of = Entire Heart
  - Entire Heart: Part of = Entire heart and pericardium
  - Entire heart and pericardium: Part of = Entire middle mediastinum, Part of = Entire cardiovascular system

    Therefore, Atrioventricular junction is a part of the Entire heart and pericardium, Entire middle mediastinum, and the entire cardiovascular system.
- Reflexive – a part is a part of itself
- Antisymmetric – nothing is a part of its parts
  - The Entire Heart is not a part of the Atrioventricular junction

Unless properly identified, it is difficult for a reasoner to determine if it is part or whole.

This study evaluated the representation of *Part-of* relationships in the Body Structure, Pharmaceutical/Biologic Product, and Laboratory Procedure (LOINC) hierarchies, and developing and testing a proposed model where appropriate.

5.4.5.2. Tooling

termMed's termSpace authoring tool was used to evaluate the proposed model for the three hierarchies. termSpace currently supports Object Properties with reflexive and transitive properties. For the Pharmaceutical/Biological Product hierarchy, Nested Expressions were used to represent the powders used for injection solutions, as they do not currently exist as pre-coordinated concepts. termSpace can represent LOINC concepts to support the partonomy modeling of laboratory concepts; however, these concepts will

need to be transformed into a SNOMED RF2 format in order to load them into termSpace. However, the LOINC model was unable to be tested due to the complexities of adding LOINC to termSpace. Collaboration will continue with termMed to represent LOINC in termSpace to potentially test the model in future iterations.

5.4.5.3. Body Structure Concepts

There are currently 42,596 *Part-of* Relationships assigned to Body Structure concepts remaining from the 2003 decision to transform them to non-defining.

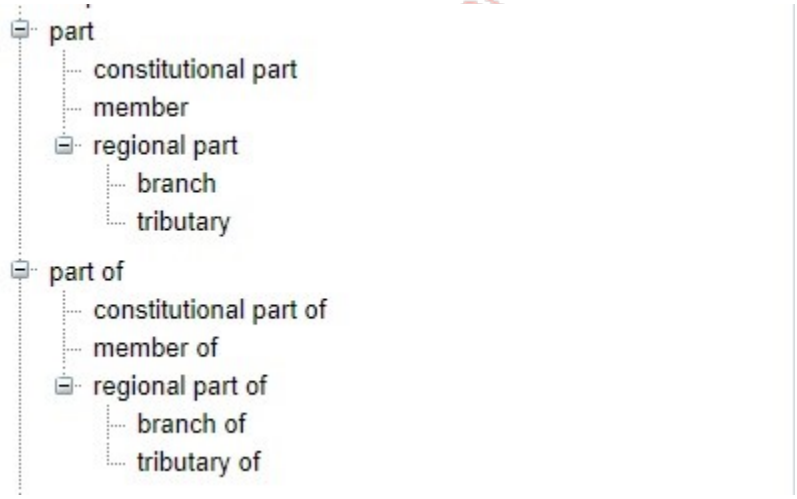
SNOMED International is currently in discussions with the Foundational Model of Anatomy (FMA) to collaborate on an anatomy model in SNOMED CT. SNOMED International is currently modeling *Part-of* relationships in a Protégé version of the Body Structure hierarchy; however, they are only exporting the resulting *IS-A* relationships. As a part of the *IS-A* and *Part-of* Modeling Subproject at SNOMED International, they plan to perform Quality Assurance (QA) to the *Part-of* relationships and assign sub-attributes of *Part-of*:

- Regional part of
- Constitutional part of
- Systemic part of

SNOMED International is currently in the process of documenting the updated Anatomy Model at: https://confluence.ihtsdotools.org/display/IAP/Revision+of+IS-A+relationships+for+anatomy

FMA also includes a role hierarchy for *Part-of* relationships.

Figure 5.13. FMA Part-of Role Hierarchy



5.4.5.3.1. Proposed Body Structure Model

With the forthcoming update to the SNOMED CT Anatomy concept model, exploration of this area is not recommended for concept model work, but instead focus on the Pharmaceutical/Substance and Laboratory hierarchies, where no current implementation of partonomy is planned.

5.4.5.4. Pharmaceutical / Substance Concepts

At this time, SNOMED project groups have not held a discussion around partonomy for Pharmaceutical/Substances. The most promising area where partonomy would apply within the Pharmaceutical Prod-

uct hierarchy are products that are made up of two or more products, for example a package that contains two separate tablets. For example, Clarithromycin 500mg tablet and lansoprazole 30mg capsule would be considered parts of a concept like 317329000 |Clarithromycin 500mg tablet and lansoprazole 30mg capsule and amoxycillin 500mg capsule pack (product)|. Concepts like this are different from a single product that contains two or more active ingredients. These packages can be made of products that have different active ingredients or can be products that have the same active ingredient, but different strengths for each product in the package.

5.4.5.4.1. Proposed Pharmaceutical / Substance Model

Add a new attribute |Has packaging component (attribute)| that will take as a value another concept from the product hierarchy. This will be a *Part-of* attribute and will need to be transitive and reflexive. These concepts will need to have a new hierarchy to live under as they are not really subtypes of the product that make up the packages but are packages that contain them. Creating a new hierarchy named “Package” containing multiple products (product) and as needed create sub-hierarchies to ease navigation is suggested.

Below are examples of the products that potentially require the addition of new product concepts in order for the new attribute to be modeled or require the use of nested expressions to represent the missing content. The pilot study represented these concepts using nested expressions, however if the model were implemented in the International Release of SNOMED CT, it may require creating pre-coordinated concepts.

- Disodium etidronate 400mg tablet and calcium carbonate 1.25g effervescent tablet pack (product) – Disodium etidronate 400mg tablet and calcium carbonate 1.25g effervescent tablet exist and will be used to fully define this concept. The purpose of the parent concept, 346404007 |Disodium etidronate +calcium carbonate (product)|, must be determined.

- Lutropin alfa 75iu injection (pdr for recon)+solvent (product) – solvent is packaged separate from the powder. Being able to model the solvent part + the powder part will allow for a fully defined concept.

There are some drugs, mainly multi-tablet packages that do have the individual clinical drugs represented as pre-coordinated concepts and will not require the use of a nested expression.

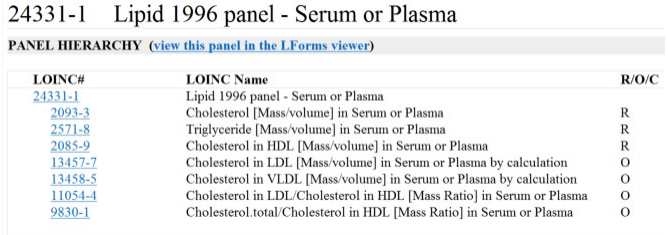
- 324934004 |Proguanil hydrochloride 100mg tablet and chloroquine phosphate 250mg tablet pack (product)| - Proguanil hydrochloride 100mg tablet and chloroquine phosphate 250mg tablet both exist as separate pre-coordinated concepts and could be used to fully define this concept.

- Quetiapine 25mg+100mg+150mg tablet starter pack (product) – This concept is a representation of three separate tablets contained within a pack. All three tablets exist as separate pre-coordinated concepts and could easily be fully defined with three separate “Has packaging” components.

5.4.5.5. Laboratory Concepts

Part-of Relationships will be useful in the definition of LOINC concepts that represent Panels. These panel concepts contain both individual laboratory tests and other panel concepts. Panels may also require multiple sufficient sets to represent tests that are not always a part of the panel but optional.

Figure 5.14. LOINC Panel with optional parts



24331-1 Lipid 1996 panel - Serum or Plasma

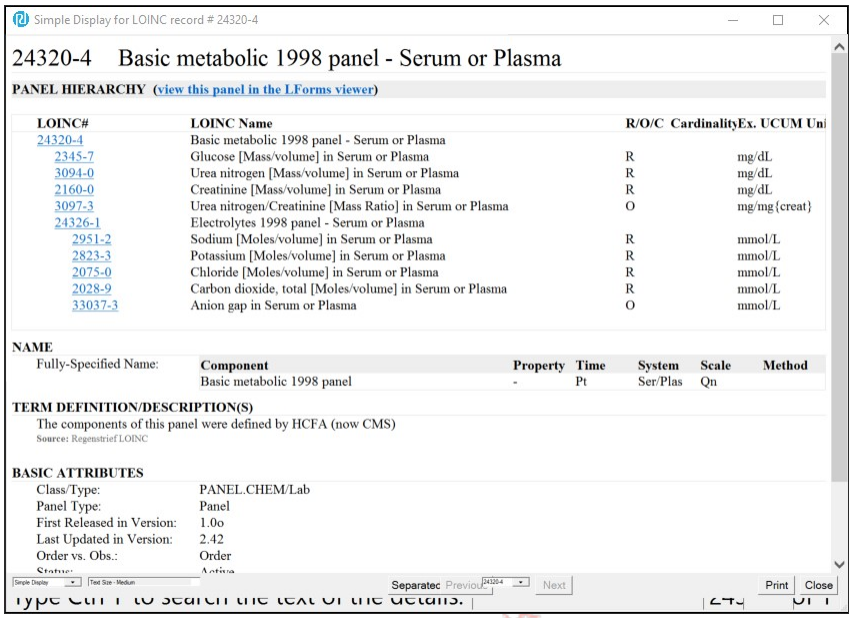
PANEL HIERARCHY (view this panel in the LForms viewer)

LOINC#	LOINC Name	R/O/C
24331-1	Lipid 1996 panel - Serum or Plasma	
2093-3	Cholesterol [Mass/volume] in Serum or Plasma	R
2571-8	Triglyceride [Mass/volume] in Serum or Plasma	R
2085-9	Cholesterol in HDL [Mass/volume] in Serum or Plasma	R
13457-7	Cholesterol in LDL [Mass/volume] in Serum or Plasma by calculation	O
13458-5	Cholesterol in VLDL [Mass/volume] in Serum or Plasma by calculation	O
11054-4	Cholesterol in LDL/Cholesterol in HDL [Mass Ratio] in Serum or Plasma	O
9830-1	Cholesterol.total/Cholesterol in HDL [Mass Ratio] in Serum or Plasma	O

5.4.5.5.1. Proposed Laboratory Model

Add a new attribute that applies to concepts in the Observable Entity hierarchy named Contains lab test (attribute). This attribute will take other Observable Entity concepts as values and will be transitive and reflexive.

Figure 5.15. LOINC Panel with multiple levels of parts



To fully represent the information contained within the LOINC Panel spreadsheet, an Ordered RefSet would have to be created because the tests contained in the panel are ordered in the spreadsheet.

In order to represent the optional tests that are sometimes part of a Panel there are several options. These optional tests and panels could be represented in an Association Reference Set, but a better representation may be using multiple sufficient sets.

5.4.6. Logical Nesting

5.4.6.1. Introduction

Figure 5.16. Example of Compositional Grammar with a nested laterality

```
125605004 |Fracture of bone (disorder)| :
        { 363698007 |Finding site (attribute)| = (72001000 |Bone structure of lower limb
(body structure)|: 272741003 |Laterality (attribute)|= 7771000 |Left (qualifier value)|),
        116676008 |Associated morphology (attribute)| = 72704001 |Fracture
(morphologic abnormality)| }
```

A Nested Expression is an expression that is defined within another expression, the enclosing expression. Due to simple recursive scope[10] rules, a Nested Expression is itself invisible outside of its immediately

[10]https://en.wikipedia.org/wiki/Scope_(programming)

enclosing expression. The nesting is theoretically possible to any ideas of depth, although only a few levels are normally used in practice. Nested Expressions have been a part of SNOMED CT post-coordinated expressions for years and are able to be represented as a part of the compositional grammar.

However, SNOMED International is not currently permitting the use of nesting outside of post-coordinated expressions. The rationale as stated in the SNOMED CT Logic Profile Enhancements document[11] is due to two main reasons currently limiting the use of nested expressions:

1. Lack of support in RF2

2. Potential for arbitrary levels of nesting

The ability to have Nested Expressions applied to pre-coordinated concepts in SNOMED CT would be beneficial to fully define concepts where the values for attributes are currently not represented as pre-coordinated concepts, for example lateralized body structures. Since creating pre-coordinated concepts to cover every aspect of medicine would lead to combinatorial explosion, Nested Expressions allow for the creation of a wide variety of concepts to supplement content that is currently missing from the international release. However, since nested expressions can be recursive, there need to be some limitations on the amount of expressivity allowed to keep content creation using nested expressions understandable and reproducible and to keep quality checks simple. Although Nested Expressions are easily represented in the compositional grammar syntax and OWL, they would require major changes to the SNOMED CT RF2 structure.

The purpose of this study is to identify a sample of expressions that are not nested and do not require nesting and a sample of expressions that should be nested and where a model for nesting is proposed.

5.4.6.2. Tooling

termMed termSpace authoring tool was used to evaluate the proposed model for nesting. Additionally, the tool allowed for addition of alternate definitions to SNOMED CT concepts and testing its effects. Although termSpace currently supports nesting, this feature will be removed with the September 2018 US Release due to the Model Advisory Group's decision not to support nesting. One concern not to support nesting is the notion that it would cause a Description Logic classifier to take longer to complete classification. However, significant decrease in classification time using nested expressions at a single level have not been seen.

5.4.6.3. Pharmaceutical / Biological Concepts

During our work on partonomy, the need to use Nested Expressions to fully define products in two instances was identified. The first instance involved concepts that did not have a precoordinated concept available to fully define drugs that were representing packages that contain multiple drugs. The second set of concepts represented a powder that is packaged separately from the solution used to mix prior to use. The model below represents the pharmacy model tested, and it has not been updated to the new drug model SNOMED International released in January 2018.

[11] https://docs.google.com/document/d/1tqNEA6S4fEF4fgj15OPabYA2E0VTz8epxvRRwczKizQ/edit#heading=h.yijdvy700v01

Figure 5.17. Example of starter pack that contains multiple tablets. Diagram contains the current SCT definition (top) and the updated definition (bottom) using partonomy with a nested expression.

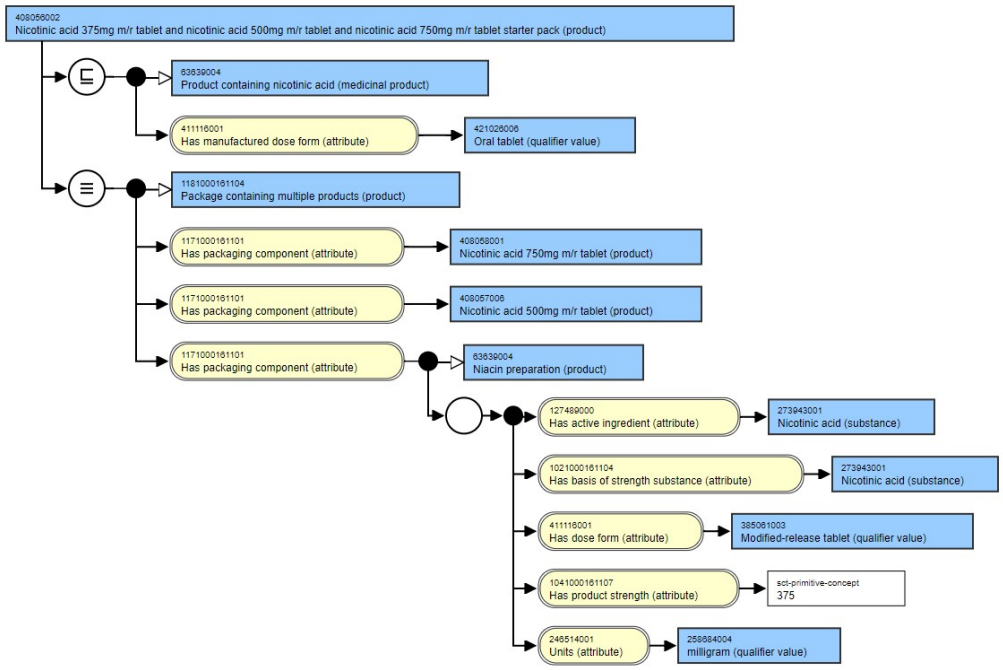
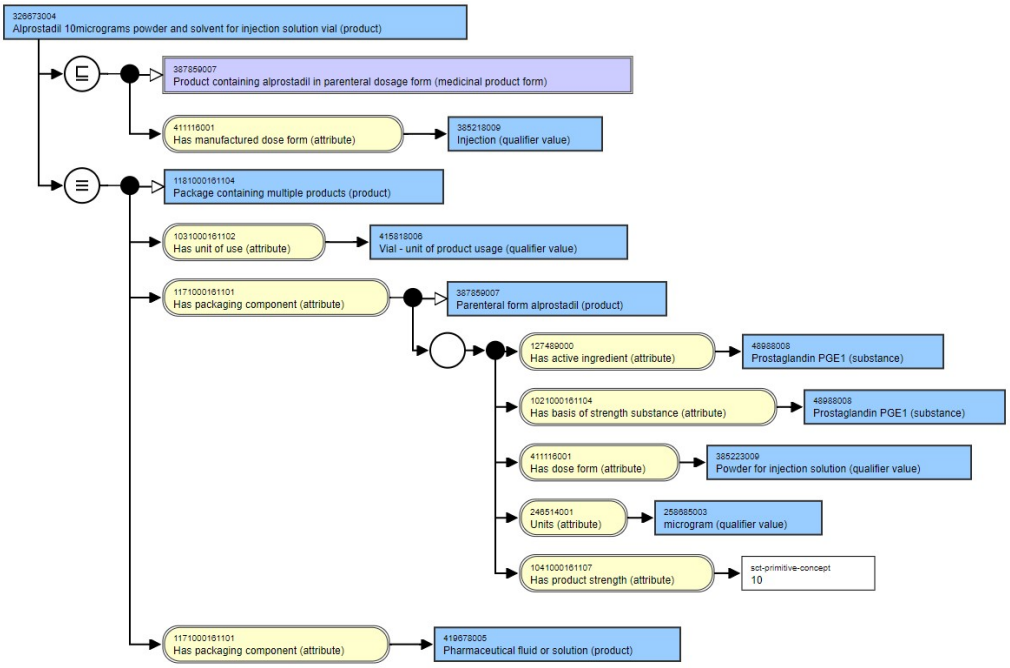


Figure 5.18. Example of an injection powder that is packaged separate from the solvent. Diagram contains the current SCT definition (top) and the updated definition (bottom) using partonomy with a nested expression.



5.4.6.4. Findings and Procedures Involving Laterality

Our work on identifying Findings and Procedures that incorrectly use laterality has identified a set of concepts that are not currently modeled correctly due to a lateralized body structure not existing as a precoordinated concept. Rather than add these concepts to an extension or submit them for addition, they could easily use nested expressions to represent the missing body structure concepts that have a laterality attribute assigned to them.

Figure 5.19. Current definition of Chronic deep venous thrombosis of femoral veins of both lower extremities (disorder) where laterality is not correctly defined.

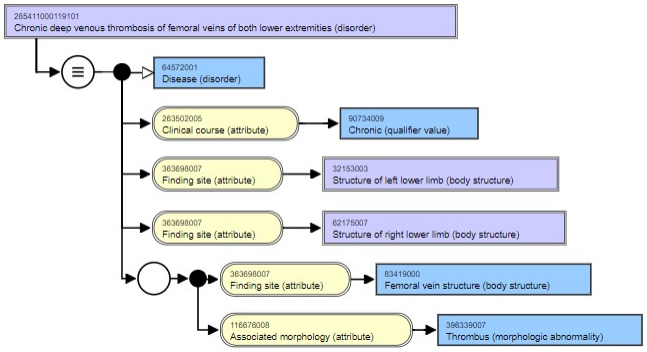
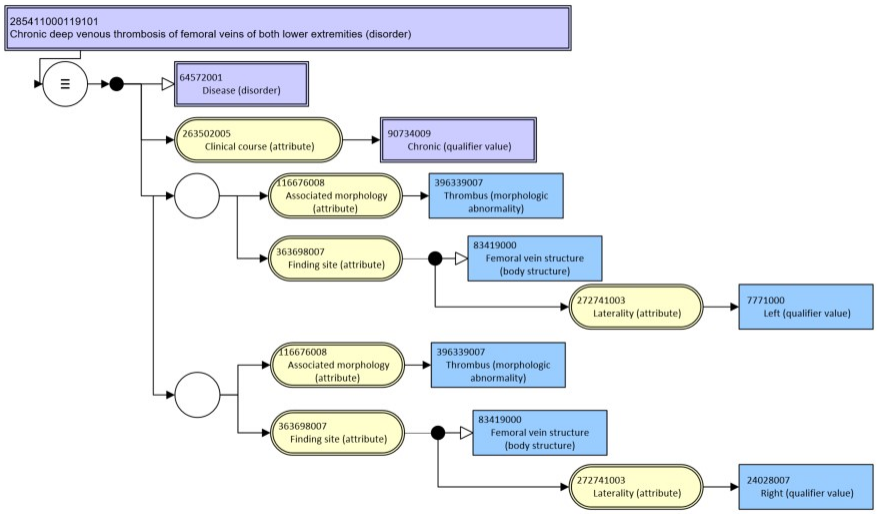


Figure 5.20. Updated definition of Chronic deep venous thrombosis of femoral veins of both lower extremities (disorder) where laterality is represented as a nested expression.



5.4.6.5. Recommendations

Support for Nested Expressions in the international and national releases would require major changes to the RF2 specification and are not a part of the recommended Logic Profile Enhancements. In addition, there must be constraints on the ability to model Nested Expressions to ensure errors are not introduced due to the ability to infinitely Nest Expressions. While Nested Expressions are not supported at the international and national level due to distribution issues, there is a definite benefit for including them in local extensions.

Using Nested Expressions to represent missing lateralized anatomy concepts will cut down the need to request new body structure concepts or temporarily creating new concepts in local extensions that will have to be reconciled with each international/national release. Representation of package concepts using Nested Expressions was the chosen method rather than creating new concepts in an extension. However, these concepts would be much better suited as a pre-coordinated concept in the international or national release as they have the potential to be used for data recording or retrieval. Any further uses of nesting outside of laterality would need to be evaluated and constrained to ensure that modeling can be easily checked for completeness and consistency. Outside of the two use cases tested for Nested Expressions, one could make modifications to findings and procedures that are used as values for defining relationships. However, in most cases these concepts would probably be better suited as pre-coordinated concepts.

5.4.6.6. Resulting Artifacts

Two Reference Sets were created:(1) those reviewed concepts where nesting could be used to represent both the product and laterality nesting and (2) those reviewed concepts that would not need nesting.

5.4.6.7. Additional Issues

When modeling the Pharmaceutical/Biological Product hierarchy, a model developed by Solor developers was used to test concrete domains. SNOMED International has since started utilizing the new drug model in the January 2018 international release, which will make the Solor developer model obsolete once the SNOMED model is implemented. The new SNOMED CT drug model will allow for the addition of more fully defined content including the addition of more values to represent concepts that include units of

presentation like cartridges. With the new SNOMED model, removal of definitions added during previous work has begun, where some were partially modeled via an automated update. For example:

- 318166002 |Bendrofluazide+potassium 2.5mg/7.7Mmol m/r tablet (product)|
- 318171009 |Bendrofluazide+potassium 2.5mg/8.4Mmol m/r tablet (product)|
- 134499006 |Budesonide + formoterol fumarate 100/6mcg breath-actuated dry powder inhaler (product)|
- 134498003 |Budesonide+eformoterol fumarate 200/6mcg breath-actuated dry powder inhaler (product)|
- 318165003 |Bumetanide+potassium 500mcg/7.7Mmol m/r tablet (product)|
- 447089002 |Amlodipine 5mg + hydrochlorothiazide 25mg + olmesartan medoxomil 40mg tablet (product)|

In the Pharmaceutical/Biological Product hierarchy, allergy kits that are represented as separate concepts were identified and should potentially be considered duplicates:

- 358640003 Silver birch allergy initial kit (product)
- 358641004 Silver birch allergy maintenance kit (product)
- 346734001 Timothy grass allergy initial kit (product)
- 346754000 Timothy grass allergy maintenance kit (product)
- 346735000 Treemix allergy initial kit (product)
- 346755004 Treemix allergy maintenance kit (product)

Part IV. Statement representation

Table of Contents

6. Representing Statements

> My Design in this Book is not to explain the Properties of Light by Hypotheses, but to propose and prove them by Reason and Experiments.
>
> —Isaac Newton

The purpose of this document is:

1. To define a *statement* for the purpose of data representation.

2. To define the types of *statements* and their *attributes*.

3. To provide a set of guidelines to model *statements*.

A statement is an expression of facts or plans. We will use two common—and misleadingly simple—statement topics: *Pulse Rate* and *Blood Pressure* as expository statements. If a patient told a clinician that their pulse rate was 120 and their BP was 160/95, or a clinician told a patient that they should keep their resting *pulse rate* below 70, and their *Blood Pressure* below 120/70, they would be mutually understood. The ability for the creator of the statement and the interpretor of the statement to each believe that they *understand* the statement is the first requirement.

6.1. Clinical Observation Modeling

Supporting Domain Semantics, Flexibility, and Interoperability

Walter Sujansky

6.1.1. Introduction

This white paper emerged from discussions among informaticists, computer scientists, and medical doctors about the appropriate modeling of clinical observations in information systems. The participants included representatives of the VHA-DoD, CIMI, HL7-FHIR, FHIM, SNOMED-CT, and OpenEHR initiatives[1]. The paper does not necessarily represent a consensus among the discussants or the viewpoint of any particular discussant. Its purpose is to provide background on the topic, to summarize a number of the viewpoints expressed, and to provide preliminary recommendations for further consideration. The contents are subject to further modification as the discussion evolves.

6.1.2. Statement Models

Statement models ([2]) are conceptual-level data models of the discrete statements about patients that can be stored in, processed by, and retrieved from a clinical information system. Statement models are defined for discrete types of clinical statements such as blood pressure measurements, lab test results, physical exam findings, patient-reported symptoms, clinical diagnoses, and other observations.

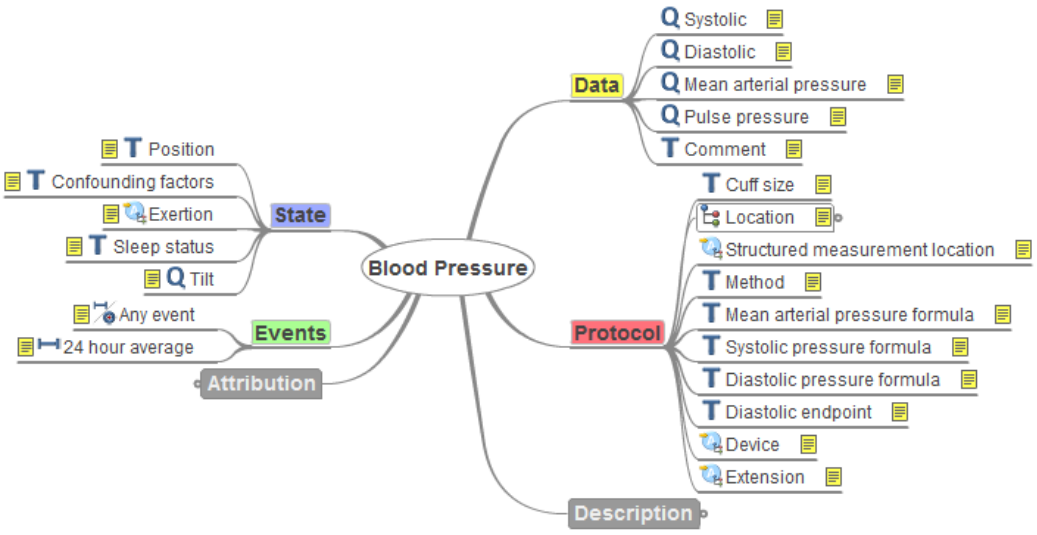
Statement models define the structure and semantics of discrete clinical observations as formal "types" that are later instantiated to represent specific recorded observations that apply to particular patients. Like object types in programming languages, these type definitions include enumerations of the specific data elements that may make up the observation, the datatypes used to populate those elements, and which

[1]VHA = Veterans Health Administration; CIMI = Clinical Information Modeling Initiative; HL7-FHIR = HL7 Fast Healthcare Interoperability Resources working group; FHIM = Federal Health Information Modeling.

[2]Statement models are also referred to as "Clinical Observation Models," "Archetypes," "Clinical Event Models," and "Clinical Models" in the informatics literature and vernacular.

elements must be populated in every instantiated object versus optionally populated. Figure 6.1 shows the graphical depiction of an example statement model for a blood pressure measurement.

Figure 6.1. Example clinical object model for a blood pressure measurement



6.1.2.1. The Role of Clinical Observation Models

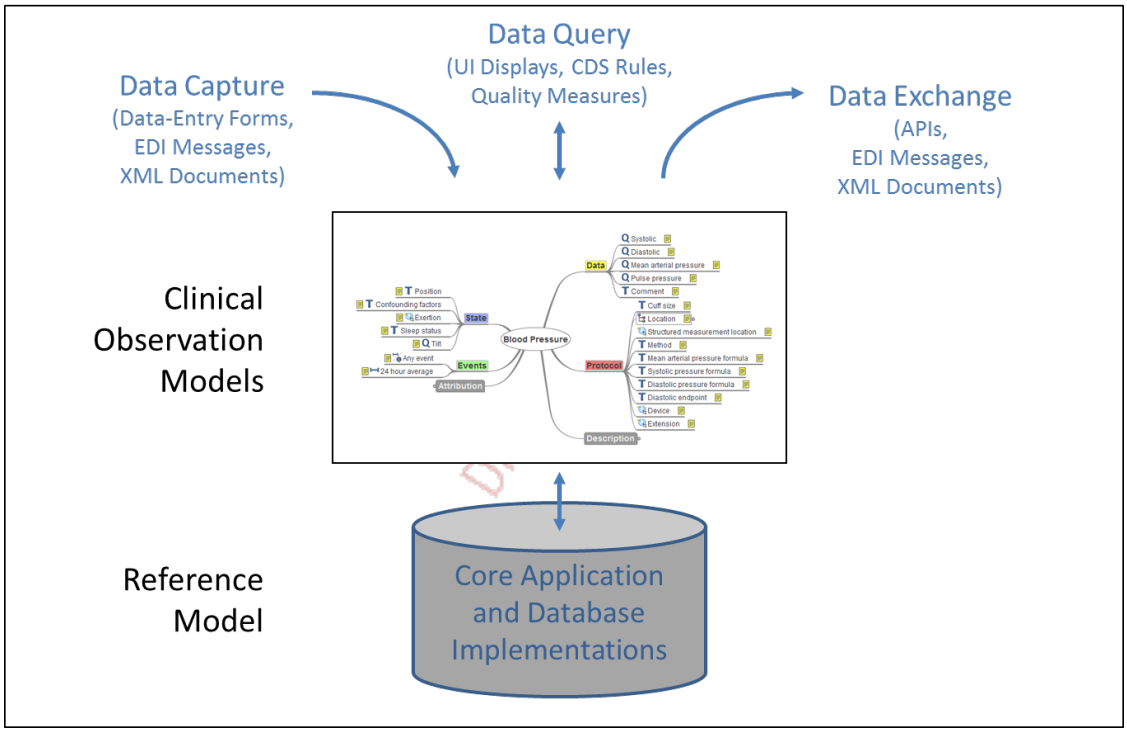
In general, clinical observation models serve at least two purposes:

1. Statement models *standardize* the capture, retrieval, and exchange of clinical observations within and between information systems. As seen in Figure 6.1, even relatively basic observations can comprise numerous sub-components. Different implementers of clinical information systems may model these sub-components and their relationships in arbitrarily different ways, which can prevent different software modules from managing and processing the same observations consistently and correctly. Formal and agreed-upon statement models provide a shared model of each type of observation that enables software modules created by different implementers to handle the same observations uniformly. Note that such software modules may comprise different parts of the same information system (such as the user interface and the rules engine of a single EHR) or entirely different information systems (such as distinct EHRs from different commercial vendors).

2. Statement models *de-couple* the creation and maintenance of domain-specific objects in clinical medicine (such as observations) from their technical implementation in software code and database structures. The types of clinical observations that may be recorded in software systems are numerous, diverse, and subject to relatively frequent modification over time, as well as customizations across clinical sub-domains. Meanwhile, the technical implementation of software applications and clinical databases is an arduous process that requires the careful design, detailed writing, and extensive testing of software code. Whenever changes are required to an application or database, a time-consuming and costly implementation process must be applied. Clinical applications and databases, however, that are implemented at a more abstract level can process *any* statement models that conform to a certain high-level reference model. Such implementations may not need to change as statement models are added or updated. Statement models can therefore serve as conceptual-level objects that represent domain-specific data and drive domain-specific functionality without being tightly coupled, at least in theory, to the underling implementation of the information system.

Figure 6.2 shows how statement models serve both of these purposes in an information system. Note how the set of clinical information models serves as a "view" or "interface" to all clinical data that may be stored by and retrieved from the information system. The design of the statement models is flexible and must conform only to a "reference model" of basic data structures. These basic structures are, in fact, the only objects tightly coupled with the underlying application and database implementations. In this manner, the statement models provide a standard conceptual model against which all data-input, data-query, and data-exchange functions operate, and that can be readily extended without (again, in theory) costly modifications to the underlying application and database. The approach for creating and maintaining information systems in this way is called *Model Driven Development*.

Figure 6.2. The role of clinical observation models in electronic health record systems

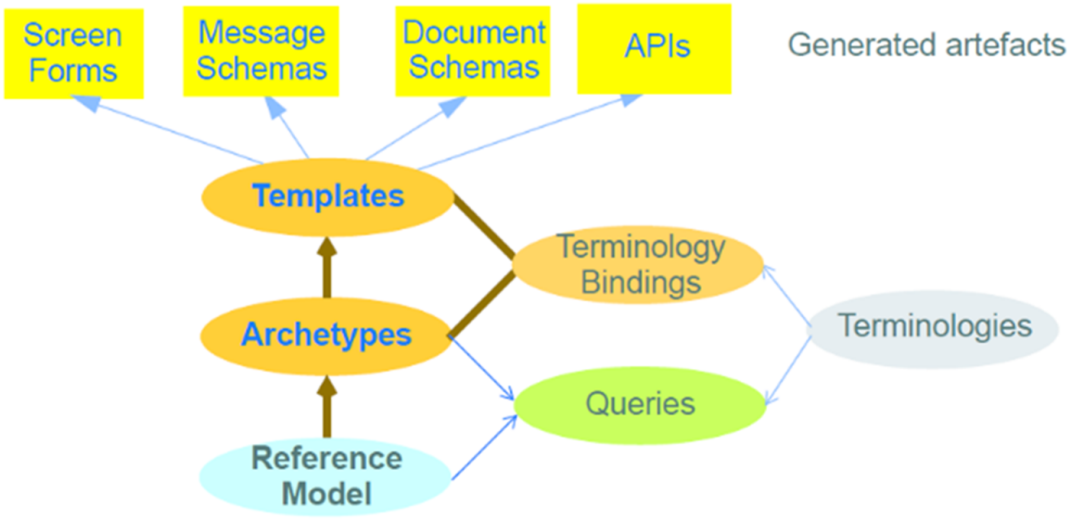


6.1.3. OpenEHR: An Example Framework for Clinical Observation Modeling

In considering the appropriate design of statement models, it's useful to review how such models will be used in practice within a Model Driven Development architecture. OpenEHR[3,4] offers one such architecture that is relatively complete and mature, so it serves as a good example. Figure 6.3 illustrates the components of the OpenEHR architecture, which are further described below.

[3]Demski H, Garde S, Hildebrand C. Open data models for smart health interconnected applications: the example of openEHR. BMC Med Inform Decis Mak. 2016 Oct 22;16(1):137. (available at https://www.ncbi.nlm.nih.gov/pubmed/27770769).
[4]http://www.openehr.org/what_is_openehr.

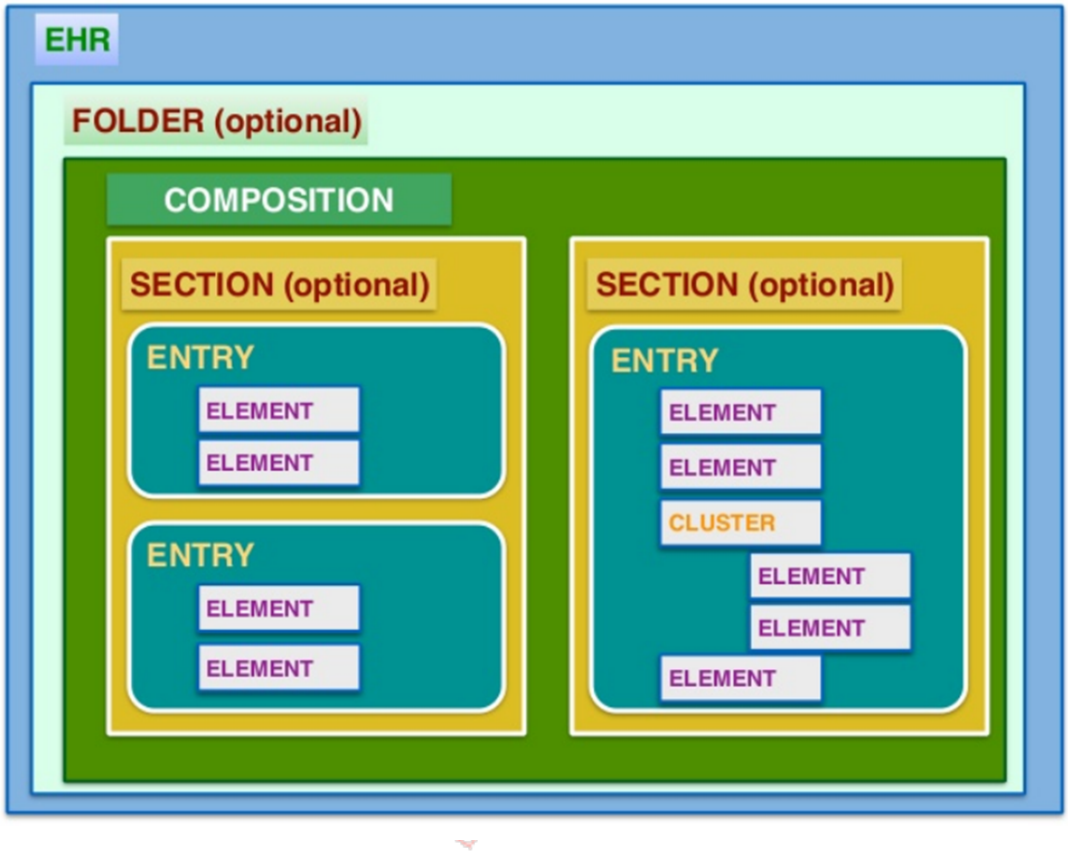
Figure 6.3. OpenEHR architecture



6.1.3.1. OpenEHR Reference Model

The foundation of the OpenEHR architecture is a reference model that contains only the most generic set of objects and data types needed to define the contents of an EHR. These objects include organizing structures such as “Folders”, “Compositions”, and “Sections”, as well as generic clinical data objects such as “Entries”, “Clusters” of entries, and “Elements” that comprise the entries. The reference model also includes several dozen data types that may be used to populate the values of Elements, such as “Quantity”, “Text”, and “Timed Event”. Collectively, these constructs define the general building blocks available to construct more detailed models for representing clinical observations, actions, and other data in EHRs. Figure 6.4 shows the constructs of the OpenEHR reference model and how they are hierarchically organized to create the “scaffolding” for patient records.

Figure 6.4. OpenEHR Reference Model



Within the reference model, the “Observation” class is a specific sub-type of the “Entry” object, and it is used to record information from a direct observation or measurement on a patient or to record the perspective of the patient, such as in history taking. The Observation class includes only a small number of data elements that are inherited by all clinical observation models, such as “Subject” (the person to whom the observation applies) and “Information Provider” (the person or agent who generated the observation). Otherwise, all Entries and Elements used to record actual observations are specified within sub-types of the Observation class, which OpenEHR calls “Archetypes.”

6.1.3.2. OpenEHR Archetypes

Archetypes are clinical object models that specify:

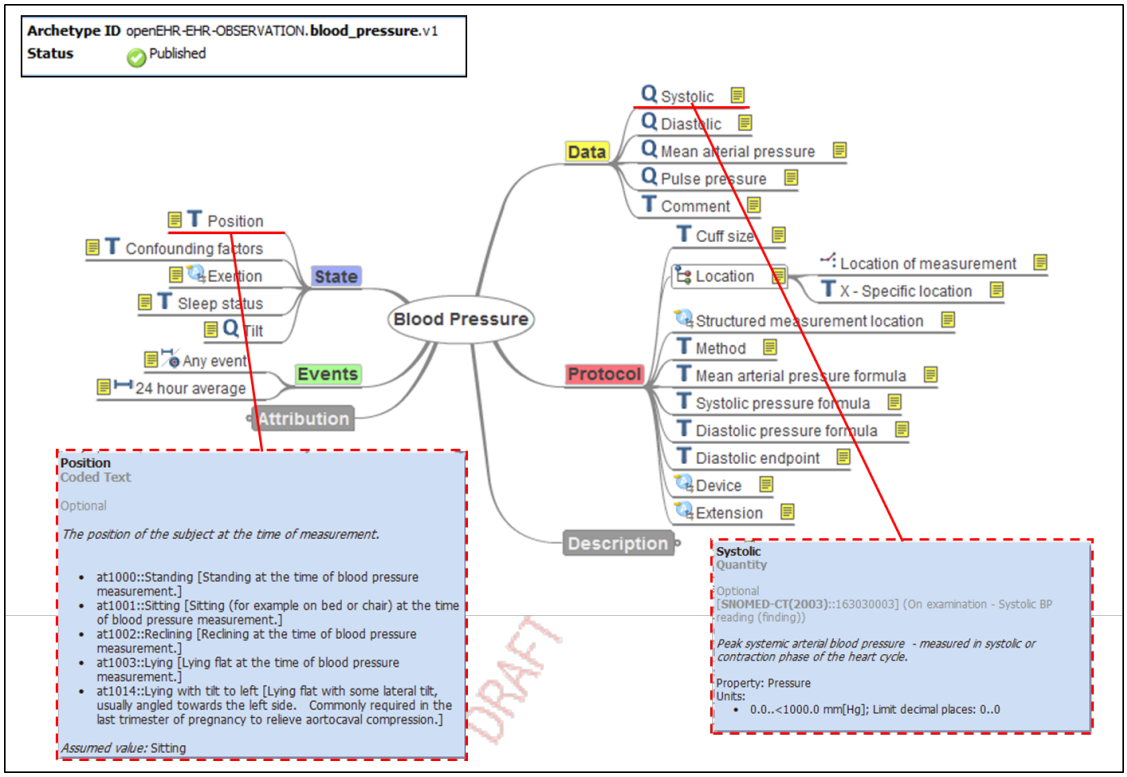
1. The set of Elements that may be used to represent various kinds of observations
2. The datatypes used to populate those Elements
3. Which Elements must be populated versus being optional, and
4. Whether Elements can have only one or may have multiple values.

The values of Elements, themselves, may be collections of other Elements (“Clusters”) or instances of other Archetypes (effectively, nested Archetypes). Figure 6.5 shows the graphical representation of an OpenEHR archetype.

For primitive Elements, the Archetype may define further constraints that define how the Element may be populated, as shown in the callouts of Figure 6.5. For example, the value of the “Systolic” Element in the Blood Pressure Artifact is specified to be a “Quantity” datatype, to represent the property of “Pressure”,

and to be recorded using the units of measure “mm[Hg]”. Similarly, the “Position” Element is specified to be a “Coded Text” datatype and to be populated by one of several enumerated code values, with the code for “Sitting” being the default if no other value is specified.

Figure 6.5. Example of an OpenEHR Archetype



OpenEHR Archetypes must be defined using only the constructs of the underlying Reference Model, as shown in Figure 6.3. This constraint ensures that the Archetypes may be stored and processed by the underlying database and application implementations, which are otherwise loosely bound to the specific structures of the Archetypes themselves.

The OpenEHR framework uses a specific structured language to define Archetypes, the Archetype Definition Language (ADL). Figure 6.5 shows the graphical rendering of an Archetype, although the actual definition is specified using a text-based ADL expression (not shown). Other Model-Driven Development frameworks, of course, may use different languages for defining statement models and different graphical rendering methods.

Like structured data types and object classes in programming languages, Archetypes specify and constrain in detail how instances of actual data (clinical observations, in this case) may be represented within the information system. These specifications govern how software modules must create instances of those observations (i.e., modules such as graphical user interfaces or EDI interface engines) and how software modules may retrieve and process instances of those observations (i.e., modules such as user displays or decision-support rule engines). Using conceptual-level Archetypes rather than low-level data structures for these purposes allows domain experts to formally specify Archetypes, and (in theory, at least) de-couples Archetype specifications from low-level implementation dependencies.

OpenEHR currently includes several hundred Archetypes[5], including many for clinical observations. The framework, however, remains very much a work in progress, and many Archetypes remain in draft form.

[5]See http://www.openehr.org/ckm/ for an online listing.

6.1.3.3. OpenEHR Templates

To support specific use cases and system functions, OpenEHR allows Archetypes to be combined and/or further constrained to create purpose-specific data structures called “Templates”. Templates may then drive the automated generation of computing artifacts used to collect, retrieve, or export clinical observations (see Figure 6.3).

Figure 6.6 shows an example OpenEHR Template that represents the information captured during an initial visit to a heart failure clinic. Note that the template combines a number of Archetypes, such as Blood Pressure, Pulse, and Full Blood Count, as well as adds navigational and organizational nodes such as “Physical Exam.” The latter nodes are also Archetypes, specifically sub-classes of the Section object specified in the Reference Model.

Figure 6.6. Example of an OpenEHR Template

Although not shown in Figure 6.6, Templates may also include additional constraints applied to their constituent Archetypes. Such constraints may entail the inclusion of only a subset of the Archetype’s Elements, the allowance of only a subset of the coded values specified for an Element, the designation of default values for Elements, etc. The purpose of these constraints is to customize an Archetype for use in a specific context, while ensuring that any data collected or retrieved using Templates that contain the Archetype conform to the Archetype’s underlying constraints.

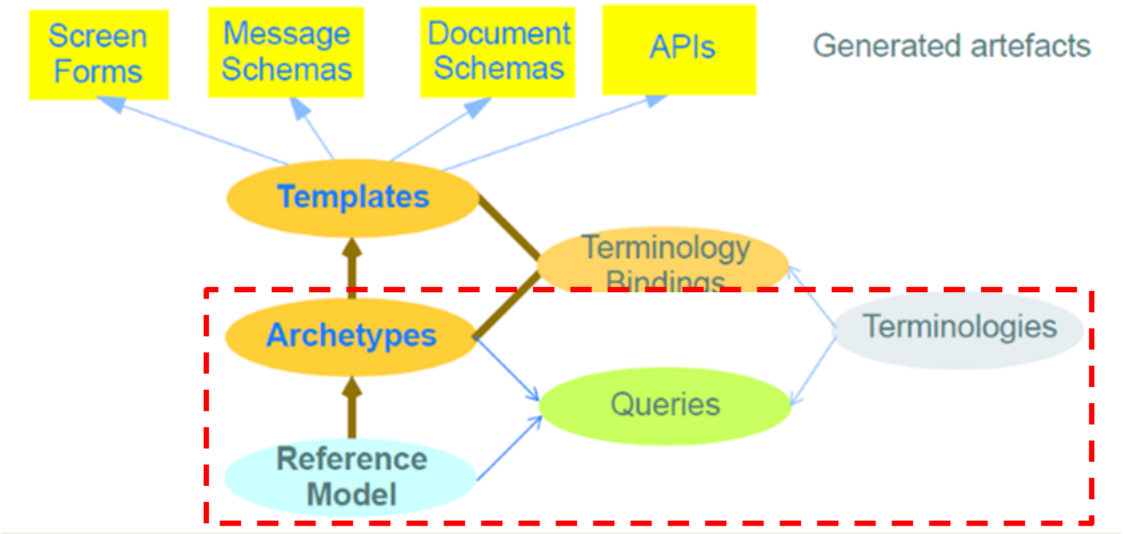
For example, Figure 6.7 shows a graphical user interface (“Screen Form”) for data entry generated from the heart-failure Template in Figure 6.6. Because the Template design constrained the Blood Pressure Archetype to include only the “Systolic” and “Diastolic” Elements (as opposed to the full set of Elements shown in Figure 6.5), the Screen Form displays only those two Elements. Note that the display includes the units of measure and allowed value ranges specified for the “Systolic” and “Diastolic” Elements, as derived from the complete Archetype. In this manner, all data collected via Screen Forms generated from the Template in Figure 6.6 will conform to the constraints specified within the Archetypes that the Template includes. This aspect of Model Driven Development allows the observation modeling features and constraints that are formally specified in Archetypes to be uniformly and automatically applied across various uses of the Archetypes (through Templates) within and across information systems.

Figure 6.7. Example of a Screen Form generated from an OpenEHR Template

6.1.3.4. Querying OpenEHR Data

Although OpenEHR Templates may combine and further constrain Archetypes to enable purpose-specific data collection and data processing, the querying of OpenEHR data need not consider the structure of any individual Templates that were used to instantiate clinical observations. Rather, querying requires knowledge of only the Archetypes, the underlying Reference Model, and any controlled terminologies used in the definition of Archetypes (See Figure 6.8 for a graphical representation of these dependencies).

Figure 6.8. Architectural components used in querying of OpenEHR data.



As discussed above, all persisted observation data must conform to the constraints of the Archetypes used to collect them (even if those Archetypes are combined and further constrained in Templates). Further, none of the navigational elements of Templates (such as the grouping of Archetypes into a “Physical Exam” category, as shown in Figure 6.6) influence the semantics of the Archetype data collected via Templates. Specifically, the semantics of a clinical observation represented by an Archetype should exist independently of any encompassing navigational or organizational category in which that Archetype may appear within a Template (Archetypes must be carefully designed to confer this property).

At the same time, queries may reference sub-parts of an OpenEHR medical record in which the Archetype instances were recorded. These named sub-parts of a record, such as “Problem List” and “Medication Order List,” are also Archetypes defined to specialize the “Section” class of the Reference Model (see Section 6.1.3.1).

Finally, queries may also reference the terminology model from which specific codes were drawn when defining clinical observation Archetypes. For example, a query could seek to retrieve any patient with a

diagnosis subsumed by the coded concept “Cardiovascular Disease,” although no Archetype specifically references that very general disease concept. Such a query would rely upon the hierarchical subsumption relationships represented in the terminology model to associate the general “Cardiovascular Disease” concept with the specific disease concepts (such as “Atherosclerosis”) that are actually referenced in defined Archetypes.

6.1.4. Patterns for Clinical Observation Modeling

Model-Driven Development provides a useful framework to build EHR systems that include standardized representations of medical data and that are flexible and extensible. However, the ultimate effectiveness of these EHR systems depends to a great extent on the specific design of the clinical observation models they include. As discussed, the same types of observations may be modeled in many different ways, and the design choices made will influence the ease and consistency with which the clinical observation models can be used. This section discusses some of those choices and the design criteria that should govern them.

6.1.4.1. Clinical Observations in the Abstract

It’s useful to consider what clinical observations essentially are. In the abstract, they are discrete patient descriptors that document information gathering, diagnostic testing, and decision making about patients. Such descriptors may include, for example, a diagnosis, an LDL cholesterol level, a systolic blood pressure measurement, an Apgar score, a patient-reported symptom, or a family history.

Each clinical observation pertaining to a patient consists in the abstract of two general components:

- The Aspect of the patient that is being described, either implicitly or explicitly. For example, the observation “The patient’s systolic BP is 130 mmHg” explicitly describes the Aspect “Systolic Blood Pressure,” whereas the observation “The patient has asthma” implicitly describes the aspect “Diagnosis”. If the general form of a patient descriptor is “The patient has X of Y”, the aspect denotes “X”.
- The Value or Magnitude of the descriptor. For example, the observation “The patient’s systolic BP is 130 mmHg” specifies the magnitude “130” whereas the observation “The patient has asthma” specifies the value “Asthma”. If the general form of a patient descriptor is “The patient has X of Y”, the value or magnitude denotes “Y”.

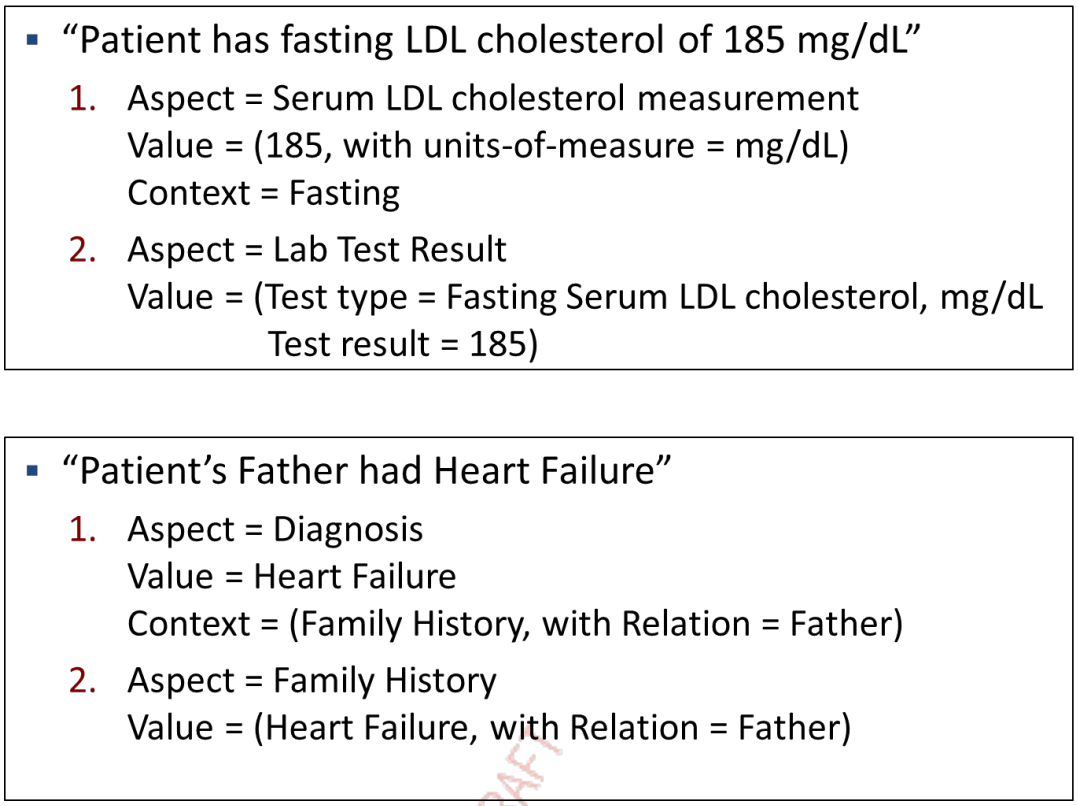
The aspect and the value/magnitude of an observation may, themselves, be further modified or qualified to denote the complete semantics of the observation. For example, the aspect “Systolic Blood Pressure” in the example above could be further qualified by the date/time that the measurement was taken or the position of the patient at the time it was taken. Likewise, the magnitude “130” in the example above could be further qualified to specify that the units of measure that apply are “mmHg”.

Sometimes, a third component of a clinical observation is specified:

- The Context in which the clinical observation occurred or was recorded. This component typically denotes information that is important to record but does not directly modify the Aspect or the Value/Magnitude. Examples may include who specifically reported the observation (e.g., the patient versus the patient’s mother) or what instrument or technique was used to collect the observation (e.g., by rhythm strip versus 12-lead EKG). Notably, there is sometimes a fuzzy distinction between information that modifies the Aspect of a clinical observation and information that denotes its Context. For example, the fasting state of a patient at the time a serum LDL cholesterol measurement was taken could be considered to denote the Context of the measurement (with the Aspect being simply “Serum LDL Cholesterol”) or the fasting state could denote a qualifier of the Aspect (with the Aspect being “Serum LDL Cholesterol, with FastingState = True”).

Based on these abstract components of a clinical observation, the same observation can be modeled in different ways. The examples in Figure 6.9 show reasonable variations in the use of aspect, value, and context to represent the same observation semantics.

Figure 6.9. Example variations in modeling of clinical observations



6.1.4.2. General Design Patterns for Clinical Observations

At least three general structural patterns may be considered for the design of clinical observation models: Assertion, Evaluation, and Belief:

- Assertion pattern. No Aspect is explicitly specified; a Value, with possible qualifiers is always specified; a Context is optionally specified. Example:
  - Aspect = NULL
  - Value = (Asthma, with type = intrinsic, with severity = mild, with status = active)

  This pattern assumes that, for every Value, the Aspect of the patient that is being described is implicit and unambiguous, and therefore need not be explicitly specified. The pattern is most naturally suited for symptoms, exam findings, past medical history findings, and diagnoses, where the assumption usually holds. However, exceptions exist. For example, the Assertion pattern cannot distinguish between a patient-reported symptom of "arm weakness," and a physical exam finding of "arm weakness" (unless "patient-reported" or "physical-exam" are denoted as Contexts). .

- Evaluation pattern. An Aspect is always specified; a Value, with possible qualifiers is always specified; a Context is optionally specified. Example:
  - Aspect = Serum LDL Cholesterol
  - Value = (185, with units-of-measure = mmHg)
  - Context = Fasting

  This pattern explicitly specifies the Aspect and considers it the "question" that the observation is addressing. The Value constitutes the "answer" to the question. The pattern is most naturally suited to observations represented as "attribute/value" pairs, such as simple testing results (blood glucose, FEV1),

scoring instruments (Apgar, Braden scores), and patient characteristics with quantitative or ordinal values (pulse, pain intensity).

- Belief pattern. An Aspect, with possible qualifiers, is always specified; a Value, with possible qualifiers is always specified; a Context is optionally(but rarely) specified. Examples:
  - Aspect = Diagnosis
  - Value = (Asthma, with type = intrinsic, with severity = mild, with status = active)
  - Aspect = Serum LDL Cholesterol, with Fasting-State = True
  - Value = (185, with units-of-measure = mg/dL)

  This pattern is the most general and can be applied equally to symptoms, findings, diagnoses, test results, scoring instruments, and quantitative characteristics. It does require, however, that an Aspect is explicitly specified in all cases as part of the observation model (although this constraint does not necessarily require that the Aspect be specified by users at the time the observation is instantiated, since user-interface functionality may populate the Aspect automatically and "behind the scenes" for observations where it is implied and unambiguous).

6.1.4.3. Desiderata for Clinical Observation Model Design Patterns

Given that multiple design patterns exist for clinical observations, it's useful to consider design criteria that can guide modeling choice. Among the best known criteria for designing clinical concepts are the properties of Understandability, Reproducibility, and Usability (URU)[6], defined as follows:

- Understandability: Concept definitions should be understandable by average clinicians and others who use the definitions (such as data analysts), given brief explanations.

- Reproducibility: The retrieval and representation of the same concept should be consistent regardless of the nature of the interface, user preferences, or time of entry.

- Usefullness: One should model concepts, concept properties, and distinction among concepts only for which there is current use in healthcare.

Among these criteria, reproducibility is arguably the most important in selecting an optimal design pattern for clinical observations, because the property of reproducibility most influences the value of clinical observations as *standardized* representations of clinical information that can be shared by different software modules and information systems. As illustrated in Figure 6.2, multiple software modules may use the same clinical observation models to implement distinct functions. To ensure that the creation, use, and exchange of clinical data is done uniformly, the clinical object models must not vary according to the contexts in which they are created or processed, i.e., they must be reproducible.

To help ensure reproducibility, modelers should follow at least two guidelines when creating clinical observation models: Avoid arbitrary variation and explicitly represent clinically relevant distinctions. Figure 6.10 illustrates relevant examples and counterexamples of these guidelines. Note that the first example shows three different modeling patterns for the same type of observation. In this case, it would be preferable to model all observations of this type using only one of the patterns (applying any one of the patterns to all three observations is left as an exercise for the reader). The second example shows an observation for which the complete clinical meaning of the finding ("Weakness in Right Arm") depends on whether it was objectively discerned by the physician through examination, or just subjectively reported by the patient.

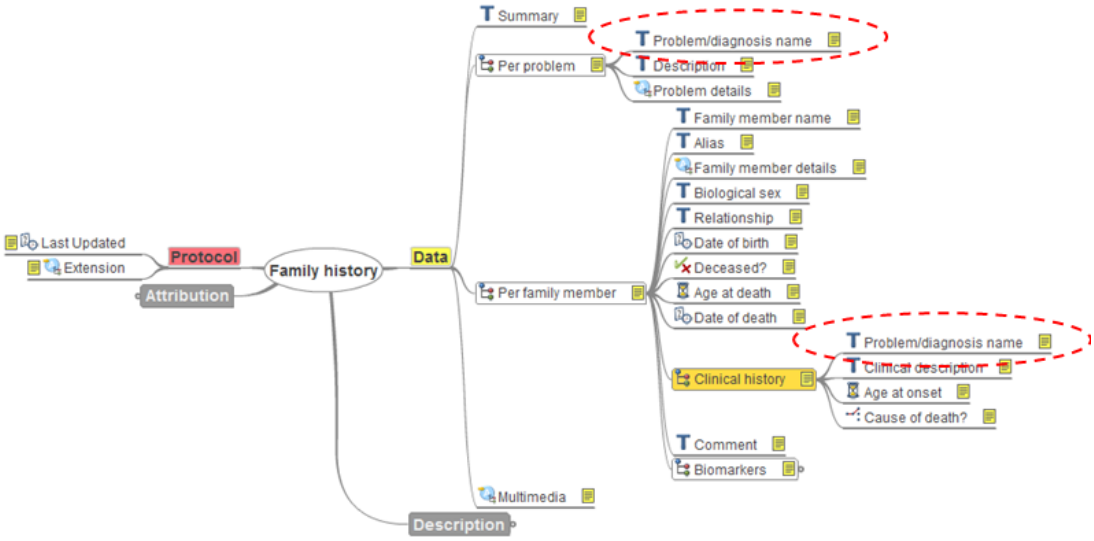
[6]Walker D. GP Vocabulary Project—Stage 2 Report: SNOMED Clinical Terms (SNOMED CT); November, 2004. Available from: https://www.semanticscholar.org/paper/Gp-Vocabulary-Project-Stage-2-Snomed-Clinical-Walker/4353b85e1afbeb93b81b38398f94882c6d5119cd.

Figure 6.10. Guidelines for designing clinical observation models

- Avoid arbitrary variation, such as
  1. Aspect = NULL
     Value = Regular pulse vs.
  2. Aspect = Skin Turgor
     Value = Normal vs.
  3. Aspect = Physical Exam Finding
     Value = Brisk Knee Reflex
- Explicitly represent clinically relevant distinctions, such as
  1. Aspect = Patient-Reported Symptom
     Value = Weakness in Right Arm vs.
  2. Aspect = Physical Exam Finding
     Value = Weakness in Right Arm

Figure 6.11 shows a poorly designed clinical observation model that violates the reproducibility criterion. Using this model, the family history of a particular problem or diagnosis could be represented in two different ways, depending on the user's preference. Such variation in the representation of the same observation entered by one user or another will necessarily complicate subsequent data querying and analysis. For example, a data analyst seeking all patients with a family history of coronary artery disease would have to search both the "Per problem" and the "Per family member" paths of each "Family History" observation stored in the EHR.

Figure 6.11. A poorly designed clinical observation model



6.1.4.4. Recommendations

Given the Model-Driven Development approach and the design considerations described above, two general options exist for specifying clinical observation models:

1. Standardize on a single design pattern for all clinical observation models (i.e., either the Assertion, Evaluation, or Belief pattern described in Section 6.1.4.2). This approach may facilitate the tasks of data analysts and software developers, who will need to learn many clinical observation models to use them effectively in application development, CDS rule design, clinical measure specifications, etc.

   With this option, the "Belief" pattern is likely preferred, as it is the most generic and supports all manner of clinical observations, as described in Section 6.1.4.2.

2. Allow multiple design patterns, specific to individual types of observations models (e.g., all lab results, all symptoms, all physical exam findings), or even to specific observation models (e.g., distinct models for skin turgor versus knee reflex). This approach offers maximum flexibility in modeling specific clinical observations in the most natural manner. Because individual clinical observation models will often be quite complex and extensive in any case (as seen from the examples in this report), the basic pattern they follow (i.e., Assertion vs. Evaluation vs. Belief) may be the least of the variations among them that data analysts and software developers will need to be concerned with. Hence, it may not practically matter whether clinical observation models conform to a single pattern or to multiple patterns, as long as the models are clearly documented.

In net, option 2 may be the preferred approach. Modelers should allow for multiple design patterns, as needed, but strive for maximum standardization for any specific type of observation (i.e., lab result versus symptom versus diagnosis, etc.). Such an approach will enable maximum flexibility for modeling different observations in an optimal fashion, while minimizing arbitrary variations among clinical observation model designs.

6.2. Examples

A *statement* represents an entry in a record that documents in a structured/computable manner information about a subject of information, such as a patient or a relative of the patient, and that is asserted by a particular source, recorded, and potentially verified.

Clinicians author clinical statements and enter them into their organization's electronic health record (EHR). Clinicians typically input the information via a manner that we call here the *clinical input form (CIF)*. However, the CIF is not a literal form that clinicians select and enter data in. Rather, it refers to the manner in which information is presented to the clinicians and how they input the data, such as by constraining the information to allow only certain values to be entered, such as through a drop-down list or radio button, or breaking up large chunks of related information into smaller parts. For example, when a clinician orders a medication, rather than selecting this information all at once with a single item, they will choose the various parts of the medication order, such as:

- Kind of drug and strength (e.g., Acetaminophen 150 mg)
- Amount and how often the patient should take the medication (e.g., 1 tablet twice daily)
- Duration (e.g., 2 days)
- Any constraints (e.g., do not exceed a total daily dosage of 600 mg)

Ideally, the way the information is presented to clinicians is in a manner that is most efficient for the clinicians to use. However, what is an efficient way for clinicians to select and input data may not be the most efficient way for data analysts to use when they are querying data once it has been normalized and

stored in a database, such as when creating a new CDS rule or compiling prevalence statistics. For this, the data is normalized using the *analysis normal form (ANF)* and stored in a database. Again, the ANF is not necessarily a physical structure, but is how a data analyst might see the data when they are looking at it in a database, and not as clinicians would see it in the user interface (i.e., CIF).

- Clinician collects data via Clinical Input Form.
- Data is normalized via Transformation process from CIF to ANF à Representable/storable in multiple types of databases, which could include VistA but a separate process would need to be performed to make that happen.
- Data analyst who is using or querying the data (e.g., creating a CDS rule or working on prevalence statistics) via ANF (it is how the data is represented or stored in the database; must know enough about the data to know what is stored in the topic vs. what is stored as a result or detail).

Table 6.1. General Statement Model

Statement
Narrative:
Topic:
Subject of information:
Statement time:
Act:

Editorial Rule 6.1. Topic

The *topic* is the center of interest or activity represented by the statement. A few examples of topics include [# ###### ################ # ###### #### 1 # , [# ##### ###] , [# ### ######## ### #######] . For each of these topics, the information that must be described is quite different, so CIMI describes topic types that contain the appropriate properties to describe the required information for the given topic. The number of topic types will change as CIMI progresses, but currently the allowable topic types are EvaluationResult, Assertion, and Procedure.

Editorial Rule 6.2. Subject of information

The *Subject of Information* represents who or what the statement refers to. In most cases, the *Subject of Information* refers to who or what the record within which this statement is embedded is about. In such cases, the *Subject of Information* may be referred to as the *Subject of Record*. In other cases, the *Subject of Information* may refer to a relative of the *Subject of Record* (mother, father, uncle...), and would be recorded appropriately in such circumstance.

Editorial Rule 6.3. Statement time

The Statement time is the time the statement is made. The statement time is independent of the period of time that a statement refers to, which may be past, present, or future, and is represented separately as part of the act.

Editorial Rule 6.4. Act

The Act is information that details the act related to the topic, either a request act, or a performance act.

6.2.1. Statement Layer Concerns

The statement layer is primarily concerned with representation of instance data.

6.2.1.1. Measurement

6.2.1.2. Reporter

6.2.1.3. Performer

6.2.1.4. Subject of information

6.2.2. Crosscutting Concerns

6.2.2.1. Query

6.2.3. Understandable, Reproducible, and Useful

Given a narrative, fill out the form.

Example 6.1. Pulse observed to be 110

A patient tells their health-care provider that they had a [### ###] on Monday, April 23rd at 9:15 am Pacific Standard Time.

Example 6.2. Resting pulse requested to be less than 70

A health-care provider tells a patient that they would like their resting pulse to be less than 70.

In the case of a human interpreter, they can often believe that they understand a statement, even when there is a great deal of information missing from the statement. In the above example, it was probably assumed that the units used to measure the blood pressure was mm/Hg, that the patient was at rest and seated, and that the pressure was measured from a brachial artery, either the brachial artery in the right arm, or the brachial artery in the left arm.

In a face-to-face interaction, statements can often be clarified to confirm assumed content, and to ensure effective communication of information from the creator to the interpretor. When recording statements for future interpretation, such verification of assumed content cannot be performed. This inability to clarify statements after the fact requires that statements sufficiently record the circumstances necessary to reproducibly interpret the statement.

Editorial Rule 6.5. Understandable

Editorial rules must be understandable to an editor or user simply by reading the definition or rule. A statement must be understandable to the creator and the interpreter.

Editorial Rule 6.6. Reproducible

Independent observers encountering a topic and equivalent circumstances will record equivalent statements.

Editorial Rule 6.7. Useful

The representation must be useful for the purposes that the modeling is intended to support.

6.2.4. Structured Statement

Narrative: Pulse observed to be 100 bpm on Monday, April 23rd, 2018 at 9:15 am Pacific Standard Time

Action Topic: Pulse

Circumstance: facts or conditions relevant to an action; Two types of action: request, performance

Table 6.2. Patient pulse representation of narrative with Structured Statement

Performance Statement	
Narrative:	Pulse observed to be 100 bpm on Monday, April 23rd, 2018 at 9:15 am Pacific Standard Time
Topic:	Pulse
Subject of information:	Patient of Record
Statement time:	Monday, April 23rd 2018 at 9:15 am Pacific Standard Time
Act:	Circumstance: Timing: Result: 120 beats per minute

6.2.4.1. Modeling Principles

The modeling guidelines were developed in accordance with the principles shown below.

- **Separation of Concerns:** As defined by Wikipedia[7]: Separation of Concerns (SoC) is a design principle for separating a computer program into distinct sections, such that each section addresses a separate concern. A concern is a set of information that affects the code of a computer program. A concern can be as general as the details of the hardware the code is being optimized for, or as specific as the name of a class to instantiate. A program that embodies SoC well is called a modular program. Modularity, and hence separation of concerns, is achieved by encapsulating information inside a section of code that has a well-defined interface. Encapsulation is a means of information hiding. Layered designs in information systems are another embodiment of separation of concerns (e.g., presentation layer, business logic layer, data access layer, persistence layer). The value of separation of concerns is simplifying development and maintenance of computer programs. When concerns are well-separated, individual sections can be reused, as well as developed and updated independently. Of special value is the ability to later improve or modify one section of code without having to know the details of the other sections, and without having to make corresponding changes to those sections.

  The use of immutable objects (see Immutability principle below) is a technique that fulfills the Separation of Concerns principle.

  Attributes that describe specific semantic concepts should be grouped together into a single class and not be spread across a number of classes. Doing the latter leads to tight coupling between classes. Doing the former leads to better decomposition of a potentially complex domain.

- **Example:** Attributes for a Role (e.g., Practitioner) should not be mixed with attributes for an Entity (e.g., Person). This allows a person to assume a number of roles over their lifetime or to function in more than one role.

[7] https://en.wikipedia.org/wiki/Separation_of_concerns

- **Immutability:** An Immutable Object as defined by Wikipedia[8]: Used in object-oriented and functional programming, an immutable object is something that cannot be changed after it is created, in contrast to mutable objects that can be changed after they are created. There are multiple reasons for using immutable objects, including improved readability and runtime efficiency and higher security.

  Although building immutable objects…requires a bit more up-front complexity, the downstream simplification forced by this abstraction easily offsets the effort. One of the benefits of switching to a functional mindset is the realization that tests exist to check that changes occur successfully in code. In other words, testing's real purpose is to validate mutation – and the more mutation you have, the more testing is required to make sure you get it right. If you isolate the places where changes occur by severely restricting mutation, you create a much smaller space for errors to occur and have few plates to test.

  Finally, one of the best features of immutable classes is how well they fit into the composition abstraction.

- **Composition Over Inheritance:** Composition over inheritance (or composite reuse principle) in object-oriented programming is the principle that classes should achieve polymorphic behavior and code reuse by their composition (by containing those instances of other classes that implement the desired functionality) rather than inheritance from a base or parent class.

  To favor composition over inheritance is a design principle that gives the design higher flexibility. It is more natural to build business-domain classes out of various components than trying to find commonality between them and creating a family tree.

  Initial design is simplified by identifying system object behaviors in separate interfaces instead of creating a hierarchical relationship to distribute behaviors among business-domain classes via inheritance. This approach more easily accommodates future requirements changes that would otherwise require a complete restructuring of business-domain classes in the inheritance model.

  Item for Consideration: *Should we say that we only allow inheritance for a single concern, i.e., we can subtype measurement but not subtype a combination of phenomenon type and measurement type?*

- **Statement Model Stability:** Stability is different from immutability. Stable means that the model can still meet unanticipated requirements without having to change. It is not acceptable to change the model every time a new way to administer a drug or to treat a condition is identified. By representing these types of potentially dynamic concerns in the terminology expressions, as opposed to static fields in a class structure, we do not have to change the model every time something new is discovered. As Terry Winograd said, anticipating breakdowns, and providing a space for action when they occur, is a design imperative.

  In some regards, in this context "stable" means "not brittle." A model easily broken by changes that someone could anticipate is one possible definition of brittle. A stable model is critical in the phase of a known changing landscape. We do that by isolating areas of anticipated change into a dynamic data structure. That dynamic data structure may also be immutable in an object that represents a clinical statement.

- **Overall Model Simplicity:** In cases where different principles collide, we shall favor the enhancement of simplicity of the entire system over simplicity in one area of the system.

- **Cohesion:** Related classes should reside in the same module or construction. The placement of a class in a module should reduce the dependencies between modules.

- **Reusability:** Architectural patterns should encourage class reusability where possible. Reusability may further refine encapsulation when composition is considered.

[8] https://en.wikipedia.org/wiki/immutable_object

- **Assumption-free:** Implied semantics must be surfaced explicitly in the model.
  - **Example:** Implicit in the statement, "I order a book from Amazon" are: paying for the book, delivery of the book to some location, and the transfer of ownership of the book from the vendor to the client.
- **Design by Composition and/or Class Specialization:** The capture of additional model expressivity must be captured by composition and/or by class specialization. The modeling approach should avoid the use of design by constraint (except for terminology binding and attribute type constraints) as it violates proper decoupling and encapsulation. An example of design by constraint is to create a single procedure class containing all attributes for all known procedures and constraining out irrelevant attributes in a more specialized model. This approach is very difficult to implement and violates numerous object-oriented best practices.
- **No False Dichotomies:** Dichotomies that are not completely disjoint (mutually exclusive) lead to arbitrary classification rules and result in ambiguity based on different assumptions about the domain. These must be avoided.
- **Model Should Avoid Semantic Overloading (semantic precision):** Semantic overloading occurs when a model attribute's meaning changes entirely, depending on context. While the refinement of the semantics of an attribute in a subclass is acceptable, a change of meaning is problematic. For instance, in FHIR, the Composition class defines an attribute called Subject. In some subclasses, the attribute may be the entity that this composition refers to (e.g., the patient in a medical record). In other cases, it is the topic being discussed by the composition (e.g., a medication orderable catalog).
- **Convention Over Configuration:** Convention over configuration (also known as coding by convention) is a software design paradigm used by software frameworks that attempt to decrease the number of decisions that a developer using the framework is required to make without necessarily losing flexibility.
- **Model Consistency:** Patterns should allow the consistent representation of information that is commonly shared across models. For instance, attribution and participation information should be captured consistently. Failure to do so forces implementers to develop heuristics to capture and normalize attribution information that is represented or extended differently in different classes (e.g., FHIR).
- **Model Symmetry:** There should be symmetry in the models wherever we can have it.
- Iterative development and validation using use cases

Table 6.3. Pulse Measurement Statement

Performance Statement		
Narrative:		
Topic:		
Subject of information:		
Statement time:		
Performance Act:	Circumstance:	Timing:
	Result:	120 beats per minute

Table 6.4. Pulse Request Statement

Request Statement
Narrative:
Topic:

Request Statement		
Subject of information:		
Statement time:		
Request Act:	Circumstance:	Timing:
		Repetition:
	Requested result:	< 70 beats per minute

6.2.4.2. Measurement

Editorial Rule 6.8. Measurement

Define measurement

Editorial Rule 6.9. Lower bound

The lower bound is the smallest reported value of the measurement. If only one value is reported, then the lower bound is the same as the upper bound.

Editorial Rule 6.10. Upper bound

The upper bound is the largest reported value of the measurement. If only one value is reported, then the upper bound is the same as the lower bound.

Editorial Rule 6.11. Include lower bound

Indicate if the lower bound is within or outside the interval represented by this measurement.

Editorial Rule 6.12. Include upper bound

Indicate if the upper bound is within the interval represented by this measurement, or outside the interval represented by this measurement.

Editorial Rule 6.13. Resolution

An optional numeric representation of the resolution of this measurement, using the same semantics as the measurement itself.

Editorial Rule 6.14. Measure semantic

A concept that defines the semantic interpretation of the upper and lower bounds of this measurement.

6.2.5. Statement Types

The types of clinical statements are listed and described below. The rationale for selecting these types is: Clinicians basically do two categories of things with a patient that need to be documented as clinical statements.

1. **Performance of action**: Actions may include passive observation of a phenomenon related to patients and their health status or family history, and may also include active interventions, such as providing education or administering medications or documenting that a patient is participating in exercise to improve their overall health status.

2. **Request for action**: Requests for future actions may include defining goals, consultation with other providers, or active interventions.

NOTE: Given that this work is not finalized yet, it is possible that additional clinical statement types may need to be added in the event during creation of the KNARTs there are clinical terminology artifacts identified that do not fit into any of the types listed above.

Any statement that states or implies an “if/then” clause should be expressed and captured as an Event Condition Action (ECA) rule.

Example:

- “Free-text reminder: Consider [ordering X procedure] for patients with suspected pericarditis, myocarditis, hypertrophic cardiomyopathy, or pulmonary hypertension.”
- Implied “if/then” clause: **IF** pericarditis, myocarditis, hypertrophic cardiomyopathy, or pulmonary hypertension is suspected – **THEN** consider ordering X procedure.
- Rather than capturing the above statement as a free text reminder, building an appropriate ECA rule should be considered.

6.2.5.1. Performance Statements

An action statement describes an action that has previously been performed, and – if applicable - the results of that action. As shown in the examples below, this can range from documenting that a subject of record:

- Was observed to have the presence or absence of a clinical phenomenon
- Underwent a specific test/screening or procedure, and its resultant value, if any
- Was administered a medication or other substance
- Was provided educational materials
- Has any other state or specific characteristic that is clinically relevant

If the action statement:

- Regards a measurement that was taken, all information about that measurement will be included as part of the clinical statement, such as its value and unit of measure and any details about how the measurement was taken.
- Results in an order(s) placed during the same encounter that was made to learn more about the phenomenon or to monitor it, then a link will be made to the order(s).

Examples of Action clinical statements:

1. Systolic blood pressure of 120 mmHg taken from right brachial artery while seated and no more than 30 minutes from when the patient last urinated
2. Diabetes mellitus is present
3. Diabetes mellitus is not present
4. Three dot blot hemorrhages
5. Dot blot hemorrhage is present
6. Patient taking one Acetaminophen 100 mg tablet by mouth daily as needed for pain
7. Positive screen for fall risk
8. Negative screen for PTSD and depression

9. Family history of colon cancer

10. Patient provided educational materials on pre-diabetes diagnosis

11. Patient counseled on the health risks of continuing smoking

6.2.5.2. Request Statements

A Request clinical statement describes a request for an action made by a clinician. Most of the times, but not always, the object of the request (e.g., lab test, medication order) will be fulfilled by someone other than the clinician (e.g., lab technician, pharmacist) making the request. All information about the request will be documented in this clinical statement, including information about details relating to the request, such as patient must fast for 12 hours before having a lipids blood test.

Examples of Request clinical statements:

1. Lipids panel for patient Jane Doe. Patient must fast for 12 hours prior to the blood test.
2. Head CT with contrast for patient John Doe.
3. Cardiology referral for patient Mary Smith.
4. Penicillin medication for patient Michael Smith to be taken twice a day by mouth with food for 10 days.
5. Advised to participate in group tobacco cessation counseling once a week.
6. Advised to lose 15 pounds within 3 months.
7. Advised to exercise at least 3 times a week for 30 minutes per day for 3 months.
8. Advised to decrease the number of packs smoked per day from 3 to 2 within 6 months by using a nicotine patch.

6.2.6. Statement Building Blocks

The following components are used in multiple places within clinical statements.

6.2.6.1. STAMP Coordinate

The STAMP coordinate represents the versions of the integrated terminology and statement model used to represent a clinical statement.

6.2.6.2. Phenomena and Interval Values

In many representation models, such as SNOMED-CT and CIMI, a somewhat arbitrary distinction exists between the modeling of “Findings” and “Observable Entities.” The former typically document the presence or absence of some phenomenon in the patient (such as whether the patient has a pressure ulcer), whereas the latter characterize some feature of the patient or the patient’s condition (such as the number of pressure ulcers a patient has). Table 6.5, “An undesirable redundancy in representing clinical observations.” shows an example of the different representations for these two similar observations when modeled as Findings versus Observable Entity.

Table 6.5. An undesirable redundancy in representing clinical observations.

Pressure Ulcer as Finding	Pressure Ulcer as Observable Entity
[Pressure Ulcer(s)]#(value)#[Present]	[Pressure Ulcer(s)]#(value)#5

Pressure Ulcer as Finding	Pressure Ulcer as Observable Entity
[Pressure Ulcer(s)]#(value)#[Absent]	[Pressure Ulcer(s)]#(value)#0

Because the observation of pressure ulcers[# ######## ########] in a patient could be correctly modeled as either a Finding or Observable Entity Text before. [# # ######## ########] Text after. any subsequent query to determine whether a patient had a pressure ulcer would need to test for the observation in two different ways:

IF EXISTS object WHERE object.conceptId = "3456_PressureUlcers" AND (object.value = "Present" OR object.value > 0)

This duality of representation complicates data querying and significantly increases the possibility that data analysts will not be aware of and account for all the ways that an observation may be represented, resulting in false-negative query results.

To resolve the arbitrary distinction between "Findings" and "Observable Entities," one must consolidate these redundant concept types into the single concept type "Phenomenon." Further, one must introduce a new data type to represent the values of Phenomena, one that can express both the "presence" (present/absent/indeterminate) and numeric (integer, real) values that Findings and Observable Entities can currently represent, respectively. This new data type is an "interval value."

6.2.6.2.1. The Interval Value Data Type

An interval value data type (or "interval value") formally represents a numeric interval between two non-negative real numbers. The interval can be open or closed. Examples of interval values are:

[5,5], [0,10), $_{(0,\infty]}$, [0,0]

The formal syntax of interval values is represented by the following grammar:

Interval :: ['[' | '('] N1 ',' N2 [']' | ')']

N1 :: Non-Negative Real Number

N2 :: [Non-Negative Real Number | $_{\infty}$]

The semantics of this grammar are as follows:

'[' and ']' : Inclusive boundary (i.e. >= and <=)

'(' and ')' : Exclusive boundary (i.e., > and <)

$_{\infty}$: infinity, is > every Non-Negative Real Number

N1 <= N2

The interval value data type provides a single way to represent both "presence" values and numeric values for a phenomenon. In general, the interval value represents the numeric range within which the observed value of a phenomenon occurs. Note that this formalism allows both exact values and ranges of values to be expressed.

In the special case that the beginning and end point of an interval are the same number, *n*, the meaning is that the value of the phenomenon is *exactly n*.

[5,5] : exactly 5 ; [0,0] : exactly 0

In the special case that the beginning of the interval is a number, *n*, and the end point is $_{\infty}$, the meaning is that the value of the phenomenon is > *n* or >=*n*, depending on whether the interval is open or closed.

$_{(0,\infty]}$: > 0 ; $_{[10,\infty]}$: >= 10

The interval value also represents whether a phenomenon is "present", "absent", or "indeterminate". Specifically, any interval value that includes *only* numbers that are > 0 also denotes the value "present". Any interval value that includes only the number 0, itself, denotes the value "absent". Any interval value that includes *both* the number 0 and at least one number > 0 denotes the value "indeterminate". Lastly, there are two interval values that explicitly denote "present" and "absent," respectively. These values may be assigned to phenomena that would not otherwise take on a numeric value (such as "nausea"):

Nausea value = $_{(0,\infty]}$: present

Nausea value = [0,0] : absent

Figure 6.12, "The semantics of interval values assigned to phenomena, as shown through examples." lists a number of phenomena and how their current values (as "Findings" or "Observable Entities") would be represented instead as interval values under the model proposed here.

Figure 6.12. The semantics of interval values assigned to phenomena, as shown through examples.

Phenomenon	Current Value	Proposed Value	Semantics
Pressure Ulcer(s)	5	[5, 5]	Present, Exactly 5
Pressure Ulcer(s)	Present	(0, ∞]	Present, > 0
Pressure Ulcer(s)	Absent	[0, 0]	Absent, Exactly 0
Serum Potassium	4.5	[4.5, 4.5]	Present, Exactly 4.5
Blood Alcohol	0.8	[0.8, 0.8]	Present, Exactly 0.8
Nausea	Present	(0, ∞]	Present, > 0
Nausea	Absent	[0, 0]	Absent, Exactly 0
Nausea	Indeterminate	[0, ∞)	Indeterminate, >= 0
Daily Cigarette Use	n/a	[10, 30]	Present, >= 10 and <= 30
n/a	n/a	[10, 5)	NOT ALLOWED
n/a	-3	[-3, -3]	NOT ALLOWED

6.2.6.2.2. Comparing Interval Values using IsWithin()

Phenomena that represent clinical observations must be assigned interval values, so the querying of such phenomena for purposes of data retrieval and data analysis requires the comparison of interval values. Specifically, one must be able to test whether one interval value *is within* (i.e., encompassed by) another interval value. For example, if one wanted to retrieve only those patients who had between 1 and 5 pressure ulcers, one would test whether a patient had the phenomenon "pressure ulcer" recorded with a value interval that was within the interval [1,5]. Note that this test would retrieve patients who had pressure-ulcer interval values, for example, of [1,1], [4,4], and [3,5], but not those who had [0,0] or [1,10].

Formally, the comparison of two interval values is done using the predicate *IsWithin(i_1, i_2)*, where i_1, i_2 are interval values. The values of the IsWithin() predicate may be TRUE, FALSE, or UNKNOWN, determined as follows:

TRUE => if a number is in i_1, then it is definitely in i_2 (i_2 “subsumes” i_1)

FALSE => if a number is in i_1, then it is definitely NOT in i_2 (i_2 “$i_1$is disjoint with” i_1)

UNKNOWN => if a value is in, it may or may not be in i_2 (i_2 “overlaps” i_1)

Examples of interval-value comparisons:

IsWithin([5,5], [0,10]) => TRUE (interval i_2 “subsumes” interval i_1)

IsWithin([15,20], [0,10]) => FALSE (interval i_2 “is disjoint with” interval i_1)

IsWithin([5,15], [0,10]) => UNKNOWN (interval i_2 “overlaps” interval i_1)

Other useful examples:

IsWithin([2,2], (0,∞]) => TRUE

IsWithin([0,2], (0,∞]) => UNKNOWN

IsWithin((0,2], (0,∞]) => TRUE

IsWithin([0,0], (0,∞]) => FALSE

IsWithin([0,0], [0,0]) => TRUE

6.2.6.3. Querying Phenomena Using Interval Values

Based on the definition of the IsWithin() predicate, patient records may be queried for the presence or the numeric value of clinical observations using a single formalism.

6.2.6.3.1. UUID

The UUID is the means by which all clinical statement items that require unique identifiers are identified.

6.2.6.3.2. Logical Expression

6.2.6.3.3. STAMP Coordinate

6.2.6.4. Compound Statements

6.2.6.4.1. Use case: Systolic BP while seated with feet on the floor for 5 minutes

Principles

- **Proposed Principle 1:** Clinical statements have separable and inseparable components; clinical statements with separable components are considered *compound* clinical statements.
- **Proposed Principle 2:** Separable components are statements, which require a value.
  - The values can be:
    - numerical
    - pseudo-numerical, e.g. low/medium/high
    - Present/absent

- **Proposed Principle 3:** Clinical statements with values can stand alone.
- **Proposed Principle 4:** Clinical statements with present/absent values can be components that play a role in the focus of the statement.
- **Proposed Principle 5:** Inseparable components of clinical statements do not require values.

Compound clinical statements with separable components should be represented as "panels," with each separable clinical statement as a "stand alone" statement, which can be referenced by multiple "panels."

Examples:

Table 6.6. Separable/Inseparable Statements - Blood Pressure Measurement Use Case

USE CASE	SEPARABLE STATEMENTS	INSEPARABLE COMPONENTS
BP of 120/80 mmHg on right brachial artery, patient in sitting position for at least 5 min., using adult BP cuff, urinary bladder voided within 30 min. before measurement	Systolic BP = 120 mmHg	Using adult BP cuff
	Diastolic BP = 80 mmHg	Right brachial artery
	Time since last urination = 30 min. or less	Sitting position
	Time in sitting position = 5 min. or more	

The "panel" above would consist of the following statements:

1. Blood pressure on right brachial artery, using adult cuff, with patient in sitting position
2. Systolic BP = 120 mmHg
3. Diastolic BP = 80 mmHg
4. Time since last urination = 30 min. or less
5. Time in sitting position = 5 min. or more

Table 6.7. Separable/Inseparable Statements - Administration of Nitroglycerin Use Case

USE CASE	SEPARABLE STATEMENTS	INSEPARABLE COMPONENTS
Administration of nitroglycerin 0.4 mg tablet sub-lingual every 5 minutes as needed for chest pain; maximum 3 tablets (routine)	Strength = 0.4 mg	Administration
	Frequency = every 5 minutes	Nitroglycerin
	Maximum dosage = 3 tablets	Tablet
		As needed
		Sublingual
		For chest pain
		Routine

The "panel" above would consist of the following statements:

- Administration of nitroglycerin tablets as needed, sublingual, for chest pain, routine priority

- Medication strength = 0.4 mg
- Frequency = every 5 minutes
- Maximum dosage = 3 tablets

Pseudo-numerical values are qualitative scales, e.g.

- Low/medium/high
- Mild/moderate/severe
- Tumor staging and grading
- + pos./++ pos./+++ pos.

Statements with absent/present values are considered **inseparable** components, if they are **part of** the focus of the statement.

Example statement: Patient has warm skin and blue eyes.

Warm skin and blue eyes are the focus of this statement; both components have a value of "present" and they are **part of** the focus of the statement and are therefore considered inseparable:

- Blue eyes = present
- Warm skin = present

Other components, such as *right brachial artery* and *adult BP cuff* in the BP measurement use case are considered separable, although they may appear to be able to stand alone and have values of present/absent.

Example action statement: Systolic BP 120 mmHg taken on right brachial artery, using adult BP cuff

The right brachial artery and the adult BP have (implied) values of "present", but they are **not part of the focus** of the statement (Blood pressure). Therefore, they are considered separable.

- Right brachial artery = present
- Adult BP cuff = present

The right brachial artery **plays a role** as the site of the blood pressure. Similarly, the adult BP cuff **plays a role** as the device used to perform the measurement.

Example request statement: BP measurement to take on right brachial artery, using adult BP cuff

The right brachial artery and the adult BP have (implied) values of "present", but they are **not part of the focus** of the statement (blood pressure). Therefore, they are considered separable.

- Right brachial artery = present
- Adult BP cuff = present

The right brachial artery **plays a role** as the site of the blood pressure measurement. Similarly, the adult BP cuff **plays a role** as the device used to perform the measurement.

The two examples above show, that the focus of the statements does not change. It is in both cases the blood pressure. The roles of the right brachial artery and the adult BP cuff consequently do not change, either.

The separable components of a clinical statements are also variables. BP measurement can be performed at a different body site (e.g. left brachial artery) or using a different device (e.g. digital BP machine). However, the focus of the statement remains the same.

Other examples:

- Head CT with contrast: Contrast media plays a role as an imaging substance used
- Dobutamine stress echocardiogram: Dobutamine plays a role as a substance to induce cardiac stress
- BP measurement taken at doctor's office: The office plays a role as an environment
- Body temperature reported by nurse: The nurse plays a role as the finding informer

6.2.6.4.1.1. Details

- **Proposed Principle 1:** Details refine or further qualify the topic. Topic type and topic focus together with the details sufficiently define instance requests.
- **Proposed Principle 2:** Not every action or request requires details to be sufficiently defined.
- **Proposed Principle 3:** A detail has a key and a value, where the value can be a concept or a numeric range with unit.
- **Proposed Principle 4:** A detail can be a separable or inseparable part of a complex clinical statement.

The criteria for identifying the focus and details that are not part of the focus, but play a role in a clinical statement suggest that "details" are all components, which play a role and are therefore separable components.

Examples:

Table 6.8. Separable/Inseparable Statements – Details

Detail	Description	Has (Pseudo-) Numeric Value	Has Present/ Absent Value	Part of Focus of Statement	Plays Role	Separable/ Inseparable
Actor	Person making the request or documenting/reporting the action	no	yes	no	yes	separable
Approach/ Access Route	Passage used to reach the procedure site or take a measurement	no	yes	no	yes	separable
Body position	Position of the body during a procedure/test	no	yes	no	yes	separable
Priority	Priority of the request, e.g. Stat or Routine	yes	yes	no	yes	separable
Indication	Reason that a request was made or an action taken	no	yes	no	yes	separable
Duration	A length of time, such as for 7 days, within 24 hours, or as needed	yes	yes	no	yes	separable
Frequency	How often something must be done, such as daily, twice per day	yes	yes	no	yes	separable

Detail	Description	Has (Pseudo-) Numeric Value	Has Present/ Absent Value	Part of Focus of Statement	Plays Role	Separable/ Inseparable
Route of Administration	Way in which something, such as a medication, is given to a patient, such as by mouth/oral, intravenously, sublingual	no	yes	no	yes	separable
Strength	Strength of a unit of the medication/drug itself, such as 25 mg	yes	yes	no	yes	separable
Amount	Amount of the medication/ drug that is to be taken at a given time, such as 2 tablets	yes	yes	no	yes	separable
Dosage	Equals strength multiplied by amount, e.g. 2 tablets of 25mg equals 50mg	yes	yes	no	yes	separable
Projection	The path taken by an x-ray beam or ultrasonographical wave as it passes through the body	no	yes	no	yes	separable
Substance used	Substance such as contrast media for imaging or catecholamine for stress induction	no	yes	no	yes	separable
Device used	Device used to perform something, such as using a BP cuff to measure blood pressure	no	yes	no	yes	separable
Device setting	Specific settings for a device used to perform a procedure, such as O2 Flow Rate 5 to 12 L/min	yes	no	no	no	separable
Informer	Person who reports a test result or gives information about the patient	no	yes	no	yes	separable
Detail	**Description**	**Has (Pseudo-) Numeric Value**	**Has Present/ Absent Value**	**Part of Focus of Statement**	**Plays Role**	**Separable/ Inseparable**
Performer	Person who performs an action	no	yes	no	yes	separable
Assessment Scale	Reference scale use for scoring	no	yes	no	yes	separable

6.2.6.4.1.1.1. Details/Roles in the Context of Use Cases

- **Role: Approach/Access Route**

  - Passage used to reach the procedure site or take a measurement.
    - Excision of rib by *cervical approach*
    - Administration of enema via *rectal route*
- **Role: Body Position**
  - The position of the body during a procedure/test.
    - Colonoscopy in *right lateral position*
    - Blood pressure measurement in *seated position*
    - ECG in *lying position*
- **Role: Body Site**
  - The body site of a finding or a procedure
    - Blood pressure measurement on *right brachial artery*
    - Removal of tattoo from *left upper arm*
- **Role: Priority**
  - The priority of the request, such as Stat or Routine.
    - Blood sugar measurement 3 times/day, *routine*
- **Role: Indication**
  - The reason for a request made or an action taken.
    - ECG to evaluate *chest pain*
    - X-ray of hands to evaluate *rheumatoid arthritis*
    - Patient placed in observation status due to *suicidal thoughts*
- **Role: Duration**
  - A length of time, such as for 7 days, within 24 hours
    - Physical therapy for *3 weeks*
    - Administration of Aspirin 200mg oral tablets for pain as needed for *2 days*
- **Role: Frequency**
  - How often something must be done, such as daily, twice per day or once in a 24-hour period.
    - Chest x-ray *once daily* to evaluate pneumonia
    - Psychiatric evaluation *bi-weekly* for PTSD
- **Role: Route of Administration**

- The way in which something, such as a medication, is given to a patient.
  - Patient taking two Acetaminophen 100mg tablets *by mouth*

- **Role: Strength**
  - The strength of the medication/drug
    - Patient taking two Acetaminophen *100mg* tablets by mouth
- **Role: Amount**
  - The amount of the medication/drug that is to be taken at a given time, such as 2 tablets.
    - Patient taking *two* Acetaminophen 100mg tablets by mouth
- **Role: Dose Form**
  - The form of preparation of a medication
    - Patient taking *two* Acetaminophen *100mg tablets by mouth*
- **Role: Dosage**
  - Equals strength multiplied by amount.
    - Patient taking two tablets of Acetaminophen 100mg each = *amount of 200mg*.
- **Role: Projection**
  - The path taken by an x-ray beam or ultrasonographical wave as it passes through the body
    - MRI of brain *sagittal and transversal*
    - *Transthoracic* echocardiogram
- **Role: Substance Used**
  - Substance such as contrast media for imaging or catecholamine for stress induction
    - Head CT with *contrast*
    - *Radioisotope* study of musculoskeletal system
    - *Dye* test of fallopian tube
- **Role: Device Used**
  - A device used to perform an action, such as using a sphygmomanometer to measure blood pressure or a ventilator to help a patient breath.
    - Lithotripsy using *laser*
    - Biopsy using *Watson capsule*
- **Role: Device Setting**
  - Specific settings for a device used to perform a procedure, such as

- Oxygen therapy, *O2 Flow Rate 5 to 12 L/min.*
- Electrode setting for electro-surgery *12 watts*

- **Role: Family Member**
  - Blood relative of the patient, such as mother, maternal grandfather. This information is used to identify which family member(s) have a history of certain phenomena.
    - *Maternal* pyrexia
    - Drug misuse by *father*
- **Role: Informer**
  - Person reporting/documenting an action result or giving information about the patient.
    - Patient medical history reported by *spouse*
    - Bedside blood sugar measurement reported by *nurse*
- **Role: Performer**
  - Person performing an action
    - Blood pressure measurement taken by *physician*
    - Diabetes education given by *dietician*

6.2.6.5. Encoded Statements

6.2.6.5.1. Procedures

6.2.6.5.2. Finding, Observation, and Phenomenon

6.2.6.6. Statement Models

Analysis normal form and clinical input form

6.2.7. Validation

1. To provide a validation framework for inter-modeler reliability when applied in the field.
2. To provide information on how clinical statements will be modeled for the KBS Clinical Decision Support (CDS) Knowledge Artifact (KNART) project. Once the models are approved, model slots bound to terminologies will be identified for subsequent terminology binding definitions proposed by the VA Terminology Team. Modeling of clinical statements outside of the CDS KNART project is currently beyond the scope of this effort.

   These modeling guidelines were derived from several documented use cases. The main goal of this effort is to provide a reproducible and a principled approach to the formal capture of clinical knowledge within Information Models and their references to underlying Terminology Models. Currently, the proposal and examples are independent of any specific terminology.

   These guidelines will be distributed to a variety of participants to contribute to a modeling exercise. After having read the guidelines, participants will be asked to access a survey where they will view a

number of clinical statements and indicate how they would model them. ***When attempting the modeling exercise, it will be important to model per the guidelines specified in this document regardless of how existing terminologies, such as SNOMED-CT, may model these concepts***. In the future, an exercise to reconcile approaches may be conducted but is out-of-scope at this time.

7. Analysis Normal Form Statements

The goals of Analysis Normal Form (ANF) are to enable analysts to understand the data and how it is stored in lieu of having to teach them about the thousands of ways data can be entered (i.e., CIF), and to ensure the data we need expressed can be expressed in an operable, scalable way. The more normalized the data, the simpler it is to analyze, thus reducing the likelihood of analysis errors. The probability of patient safety risks increases greatly without the ANF. Examples of problems that can occur are:

- An inability to determine that two clinical statements are equivalent
    - Taking two 250 mg acetaminophen tablets is the same as taking one 500 mg tablet but the analyst only queries for one of the statements, not both.
    - Presence of dot blot hemorrhage and 2 dot blot hemorrhages observed are equal in regard to presence and absence but the analyst queries only for presence vs. a quantitative finding of dot blot hemorrhages.
- An inability to express something that is clinically significant
    - We may not be able to express chest pain on inspiration, which can be a sign of pleurisy. The ability to differentiate cardiac chest pain from other types of chest pain is clinically important. An example of something that needs to be represented is chest pain that worsens when you breathe, cough, or sneeze.
- An error is made in recording or in querying a repository for clinical statements
    - On October 1, 2016, a provider enters a medication order for acetaminophen 250 mg for a patient to take 1 tablet twice daily for 2 days starting October 1, 2016
        - CIF: Provider enters the medication order
        - ANF: Analyst creates a CDS rule to identify all patients ordered acetaminophen during the period September 1 – December 31, 2016. However, while the analyst creates a query to search for a clinical statement (i.e., Request) where acetaminophen was the direct substance and was ordered during the period September 1 – December 31, 2016, the analyst did not include a Request topic of “Administration of drug or medication PO BID for pain.” Thus, the medication order would not be included in the query results.

A. **ANF Clinical Statements Represent the Minimum Disjoint Set:** ANF clinical statements represent the minimum disjoint set of statement topic, result, and details and may not be further specified.

B. **ANF Classes Cleanly Separate Concerns:** ANF classes must cleanly separate the concerns of concept definition and the concerns of domain models.

- **NOTE:** Need to define the domain models thoroughly here. The strawman description is that domain models use concept definitions as a building block to define non-defining relationships or associations between concepts. The domain model represents cardinality, optionality, and other constraints.
    - **Example:** Laterality should be a concern of either the concept definition or the domain model, but not both. We can relax this principle for the Clinical Input Form (CIF) but for ANF we need a clean and invariant separation of concerns.
- **NOTE:** Need to determine better names for “concept definition” and “domain models.”

7.1. Clinical Statements

A clinical statement represents an entry in the patient record that documents clinical information:

- about a subject of information, such as a patient or a relative of the patient
- that is asserted and recorded by a particular source, such as a clinician
- in a structured/computable manner

Clinicians typically enter information into an EHR in a certain manner: the clinical input form (CIF) The CIF is not a literal “form”. It refers to the manner in which information is presented to the clinicians and how they enter the data, e.g.

- by constraining the information to allow only certain values to be entered, such as through a drop-down list or radio button
- breaking up large chunks of related information into smaller parts like in medication orders

7.1.1. Principles

- **Proposed Principle 1:** There are two types of clinical statements:
  - **Performance of action**, which include passive observation of a phenomenon related to patients and their health status or family history, and active interventions, such as providing education or administering medications.
  - **Request for action**, which may include passive observation of a phenomenon related to patients and their health status or family history, and active interventions, such as providing education or administering medications.
- **Proposed Principle 2:** Both types of clinical statements consist of topics and circumstances
- **Proposed Principle 3:** Each clinical statement can have only one topic and multiple circumstances

7.2. Clinical Statement Decision Tree

7.3. Clinical Statement Components

separation

Table 7.1. Example Clinical Statement Model

Clinical Statement	
Narrative:	Ibuprofen 400 mg tablet oral every 6 hours as needed for back pain; may increase dose frequency to one tablet every 4 hours
Statement type:	*[Request]*
Subject of info:	*[410604004 |Subject of record]*
Mode:	*[Template]*
Authors:	*[223366009|Healthcare professional]*
Action topic:	*[Procedure]-* *#(260686004|Method)#[129445006|Administration - action]* *#(363701004|Direct substance)#[197805|Ibuprofen 400 MG Oral Tablet]* *#(410675002|Route of administration)#[260548002|Oral]*
Circumstance:	Request Circumstance Timing: `[2007-04-05T14:30Z, 2007-04-05T15:00Z]±P5M` *[ISO 8601]* Purposes: *[161891005 |Backache (finding)]* Triggers: Ø associate statement backache present Participants: *[410604004 |Subject of record]* Priority: *[50811001 |Routine (qualifier value)]* Repetitions: Repetition Start: Anytime, as needed Duration: 24 hours Frequency: 4-6 hours Maximum: Ø Duration: Ø Result: 4
Associations:	Ø
Statement time:	`[2007-04-05T14:30Z, 2007-04-05T15:00Z]±P5M` *[ISO 8601]*
Stamp coordinate:	*[Solor Module]*, *[Release Path]*, 2007-04-05T14:30Z
Statement id:	a3b46565-f8cd-4354-b4b6-3dff42d33496
Subject of record ID:	Ø

7.3.1. Statement Identifier

The UUID is the means by which all clinical statements requiring unique identifiers are identified.

7.3.2. Mode

Needs clarification

7.3.3. STAMP coordinate

[Solor Module], [Release Path], [Date/Time in ISO 8601 Standard Format]

7.3.4. Narrative

The clinical statement as a whole, e.g. “Ibuprofen 400 mg tablet oral every 6 hours as needed for back pain; may increase dose frequency to one tablet every 4 hours”

7.3.5. Statement time

Time when the statement was documented in ISO 8601 Date/Time Standard Format

7.3.6. Subject of Record Identifier

UUID identifier for the subject of record.

7.3.7. Statement Authors

Figure 7.1. Participant

Optional list of participants, e.g. “Healthcare professional”, “Nurse”

7.3.8. Participant Role

Optional role for participants, e.g. “Requester”.

7.3.9. Participant Identifier

Optional. UUID Identifier for the participant.

7.3.10. Subject of Information

Subject of Information is used to express **WHO** the clinical statement is about, e.g. the patient or a family member.

7.3.11. Statement Type

Statement Type distinguishes between a performance (“performed”) and a request (“requested”). Performances may be observational performances, e.g. the observation of a clinical finding or disorder being present or absent. They can also be statements of a procedure or intervention, which has been performed on

the subject of record in the past, e.g. “12-lead electrocardiogram”. Performances can – but do not have to – include quantitative or qualitative results, e.g. “3 dot blot hemorrhages” or “Hepatitis A antibody positive”.

7.3.12. Topic

The topic is the expression of **WHAT** is being requested or what was performed. For both clinical statement types (request or performance) a pre-coordinated or post-coordinated Solor “procedure” concept as a logical expression is required to sufficiently capture the action, which is either requested or performed.

Requests for actions are always procedures or interventions:

- Stress echocardiogram
- Administration of Aspirin 81 mg oral tablet
- Systolic blood pressure measurement

Performances of actions can be performed procedures like the examples above. They can also be observational procedures, describing the absence or presence of clinical findings or disorders. In these cases, the observation action of the clinical findings and disorders is performed:

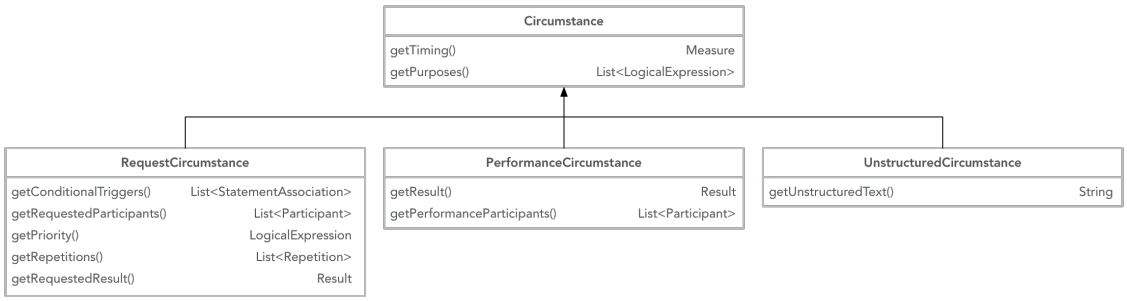
- Observation of congestive heart failure
- Observation of history of malignant neoplasm of bone
- Observation of numbness of left arm
- Observation of history of cognitive behavioral therapy

The topic is the central component of clinical statements.

- The topic defines the action being performed or requested.
- The topic has to be able to exist on its own yet still retain original intent and clarity of meaning.
- The topic includes what is being requested, measured or observed.

7.3.13. Circumstance

Figure 7.2. Circumstance, including request, performance, and unstructured subtypes

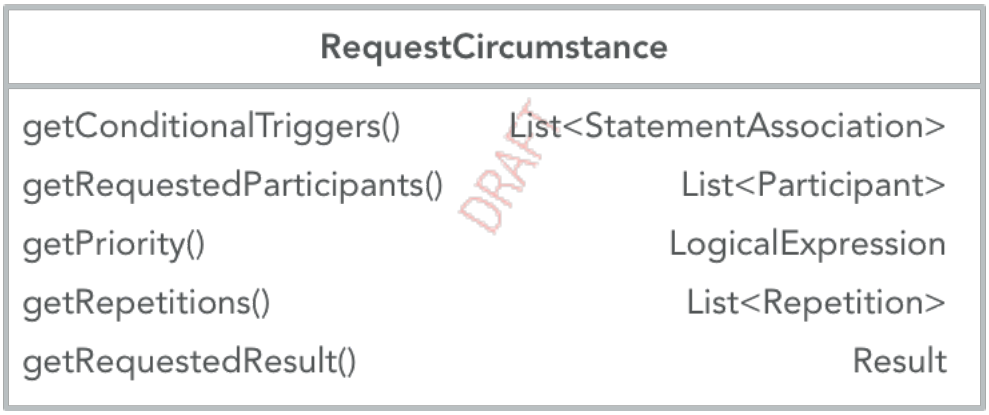


Circumstances can describe **HOW**, **WHY** and **WHEN** a requested or performed action will be or was carried out. Requests and performances have some shared circumstances:

- Timing: **WHEN** a requested action should be performed or **WHEN** an observed finding or disorder was present or absent.
  - Examples:
    - Cardiology Consult in 2 weeks
    - Breast cancer screening 3 months ago
- Purpose: **WHY** an action was requested or performed
  - Examples:
    - Echocardiogram to evaluate arrhythmia
    - Education about allergens for anaphylaxis management Other circumstances are specific to requests or performances.

7.3.13.1. Request Circumstance

Figure 7.3. Request circumstance



Request circumstances further specify **HOW** a requested action is to be performed, e.g. how often, how long or with which category of priority.

7.3.13.1.1. Conditional Triggers

Needs clarification

7.3.13.1.2. Requested Participants

Requested participants can be either specific persons or roles who perform an action, assist in performing an action or are targets of an action. **Examples:**

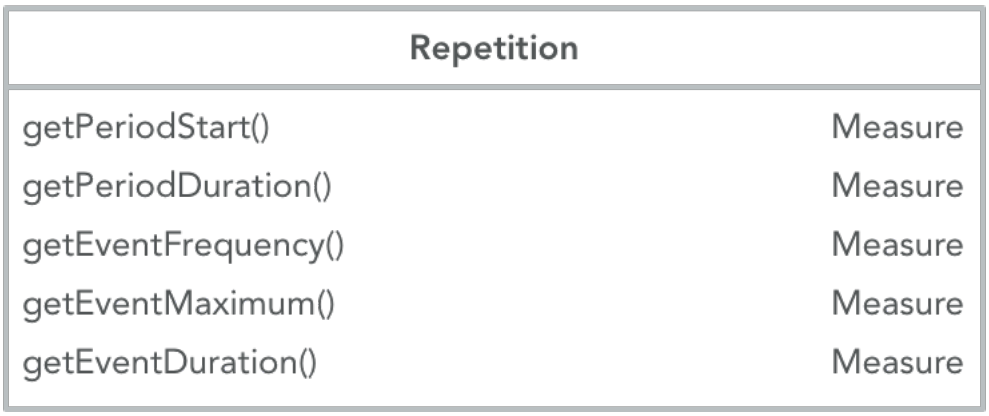
- Cardiology consultation with Chief Cardiologist
- Smoking cessation education with patient and patient’s spouse

7.3.13.1.3. Priority

Expresses the priority with which a requested action has to be carried out, e.g. “routine” or “stat”.

7.3.13.1.4. Repetitions

Figure 7.4. Repetition



If an action is requested for more than a single occurrence, the repetition allows to specify:

- When the repeated action should begin (PeriodStart), e.g. NOW
- How long the repetitions should persist (PeriodDuration), e.g. for 3 weeks
- How often the action should occur (EventFrequency), e.g. 3 times per week
- Maximal number of occurrences (EventMaximum), e.g. 10 times
- How long every occurrence should last (EventDuration), e.g. for 5 minutes

7.3.13.1.5. Requested Result

A requested result is a patient goal to be achieved or a request for action further specified or quantified.

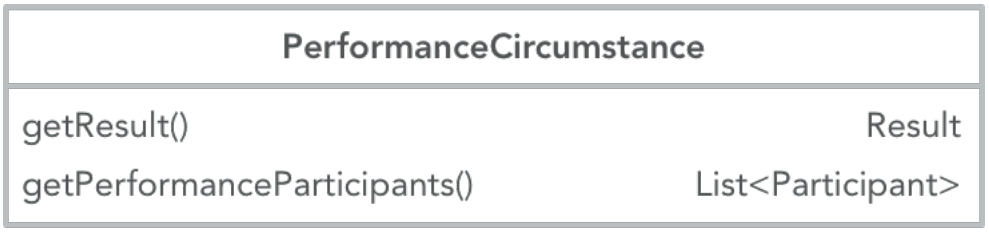
Examples:

Narrative: Administration of Metoprolol tartrate 50 mg oral daily 2 times to lower systolic blood pressure to <130 mmHg

Narrative: Diltiazem 30 mg, one tablet oral daily 4 times

7.3.13.2. Performance Circumstance

Figure 7.5. Performance Circumstance



7.3.13.2.1. Result

Result of diagnostic or observational procedures

Examples:

Narrative: Systolic blood pressure 120 mmHg

Narrative: Body weight 165 pounds

7.3.13.2.2. Performance Participants

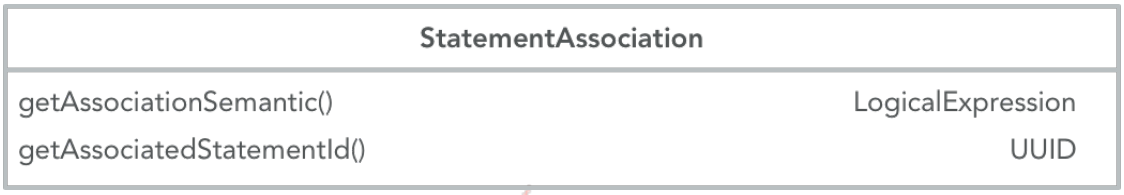
Participants in performing the action, e.g. technician, nurse

7.3.13.3. Unstructured Circumstance

7.3.13.3.1. Unstructured Text

7.3.14. Statement Associations

Figure 7.6. Statement Association



7.3.14.1. Association Semantic

7.3.14.1.1. Associated Statement ID

7.4. ANF Modeling Guidelines

7.4.1. Introduction

The purpose of this section is to describe editorial guidelines for modeling terminology artifacts used to express the content of Knowledge Artifacts (KNARTs), e.g. Documentation Templates, Consultation Requests and Order Sets, in a computer readable form. This section will attempt to outline background information related to terminology models for KNARTs as well as provide modeling guidelines necessary for encoding clinical statements. This is a working draft document and subject to change.

7.4.2. Background

Knowledge Artifacts are computable representations of Clinical Decision Support (CDS) knowledge. They consist of clinical statements and orders within a framework of structured clinical documentation. Terminology artifacts in this context are developed to represent the clinical assertions and their values and are composed of standard clinical terminologies. The prioritized terminologies for the representation are Solor terminologies (SNOMED CT, RxNorm and LOINC) in alignment with the recommendations and requirements by the Office of the National Coordinator for Health Information Technology (ONC) and the VA – Department of Defense (DoD) Interagency Program Office (IPO). This section will describe each of the terminology artifact components and provide guidelines for modeling the values of these components. These guidelines are under development and remain subject to change as a result of the need to develop a consistent terminology model and coding strategy.

7.4.3. KNART Types and Structure

Four types of KNARTs are described in the HL7 KNART Specification3):

- Documentation Template
- Order Set
- Consultation Request
- Event Condition Action (ECA) Rule

The clinical content of each KNART is specific to clinical domains and prioritized areas of focus within the domains.

Example:

- Domain: Cardiology includes
  - Chest Pain/Coronary Artery Disease
  - Atrial Fibrillation
  - VTE Prophylaxis

The “Composite KNART” for each of the clinical focus areas above is comprised of at least the documentation template, the order set and the consultation request. Many, but not all Composite KNARTs also have ECA rules.

7.4.4. Documentation Templates

Documentation templates are created to document clinical information about patients, such as History and Physical, and treatment provided in the past as well as past results from lab tests, imaging procedures and other diagnostic studies. In many cases, the clinical information captured here is associated with either a defined timeframe, e.g. diagnostic studies within the past year, or a more undefined timeframe, e.g. history of prior cardiac evaluations.

7.4.5. Order Sets

Order sets are used to document requests for diagnostic or therapeutic procedures for the patient. As such, these requested procedures will occur at a future time.

Common categories for the ordered procedures include:

- Administration/Prescription/Dispensing of medications
- Imaging procedures
- Electrophysiology procedures
- Therapies
- Laboratory procedures
- Education procedures

The requested procedures may also include additional information, e.g.

- Timing, e.g. when the action should be performed
- Specific instructions for the procedures
- Priorities
- Frequencies

7.4.6. Consultation Request

Consult Requests are often relatively short KNARTs, which include

- Reason for Consult, e.g. chest pain
- Consult Specialty, e.g. cardiology
- Priority, e.g. Routine
- Referring Physician
- Referring Physician Contact Information

7.4.7. ECA Rule

ECA Rules are used in Clinical Decision Support to trigger a defined action after a distinct event occurred. Example: Notify clinician if laboratory test result with “abnormal” flag has been received.

7.5. Terminology Service Request (TSR)

The clinical statements within a KNART, which have to be captured by standard terminologies using a number of codes from e.g., SNOMED CT, RxNorm or LOINC are represented in Terminology Service Requests (TSRs). One TSR contains a variable number of Instance Requests (IRs), each of which represents a single clinical statement. The format used to assemble and encode a TSR is a MS Excel spreadsheet template.

The example below shows orders as they potentially appear in a KNART:

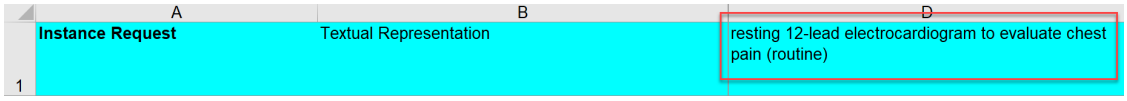
Figure 7.7. Order Example (Cardiology Order Set)

[Section Selection Behavior: More than one may be selected. Optional]

- ☐ resting 12-lead electrocardiogram to evaluate chest pain (routine)
- ☐x-ray chest to evaluate chest pain(routine)

The order from the KNART above appears in the TSR as an Instance Request:

Figure 7.8. Order Set Instance Request in TSR Template



	A	B	D
1	**Instance Request**	Textual Representation	resting 12-lead electrocardiogram to evaluate chest pain (routine)

7.6. KNART Information Modeling Overview

The Analysis Normal Form (ANF) provides a set of guidelines to model clinical statements. A clinical statement represents an entry in the patient record that documents in a structured/computable manner clinical information about a subject of information, such as a patient or a relative of the patient, and asserted by a particular source, recorded, and potentially verified.

The Analysis Normal Form (ANF) constitutes a model for defining the components of data elements from KNARTs on a general level, independent of any specific terminology. The ANF defines the principles, which distinguish the “topic” of clinical statements from the “circumstances” of e.g., an action request. The topic describes the “what” whereas the circumstances describe the “how”.

Details of the ANF model for clinical statements and their components have been discussed in previous sections of this document.

7.7. Terminology Modeling Guidelines

The request and performance clinical statement types as described in the ANF Model and Guidelines section of this document have a number of shared components. Other components are specific to the statement type. The following sections will define the terminology modeling principles for each component in detail. The choice of logical expressions to use for each component is not always straightforward, and the terms in the Solor terminologies are not always unambiguous in their semantic meaning. In situations where there may be more than one choice or more than one way to code a clinical statement or one of its components, it is important to ensure consistency of modeling approaches across clinical domains and clinical statements.

The following chapters will describe the terminology modeling guidelines based on the current ANF model and the current TSR template fields. The TSR template has two tabs for Instance Requests (IRs). One tab “request” contains IRs for requested actions, one tab “performance” contains IRs for performed actions. Both tabs have a number of fields in common. Some fields are different and unique to the specific type of IR.

7.7.1. Instance Request (Request and Performance)

Represents the clinical statement to be modeled.

7.7.2. statementID (Request and Performance)

Not for modeling. ID will be assigned by KNART developers.

7.7.3. statementType (Request and Performance)

Format: Logical Expression

Terminology: SNOMED CT

Coding: Either “385644000 |Requested (qualifier value)|” for request IRs or “398166005 |Performed (qualifier value)|” for performance IRs

7.7.4. METADATA: model fit (Request and Performance)

Currently not in use.

7.7.5. METADATA: model fit comments (Request and Performance)

Currently not in use.

7.7.6. subjectOfInformation (Request and Performance)

Format: Logical Expression

Terminology: SNOMED CT

Subject of information is in most cases the patient: 410604004 |Subject of record (person)|. However, it may also be about someone other than the patient, e.g. the patient's mother or another family member.

Examples: 72705000 |Mother (person)|, 303071001 |Person in the family (person)|

7.7.7. topic (Request and Performance)

The topic field represents, what is being requested or has been performed. Although both request and performance IRs share this field, the handling is different to a certain extent.

Format: Logical Expression

Terminology: Solor

The actual coding of the topic depends on the procedure requested or performed. Generally, pre-coordinated or post-coordinated expressions are used. Post-coordinated expressions can be "hybrids" and include terms from different terminology standards (See Medication example below).

The pre-coordinated or post-coordinated expressions in the topic field are ALWAYS procedures.

7.7.8. Medication (Request and Performance)

Currently, medications are interpreted as the administration of a medication, not the prescription. The administration can be either requested or documented as being done. Therefore, all medications are post-coordinated based on the SCT "416118004 |Administration (procedure)" concept. To capture the drug itself, RxNorm codes are used. The specific RxNorm codes depend on the specificity of the IR. Attribute/value pairs needed to fully post-coordinate the expression are SCT concepts.

Example Instance Request:

Naproxen sodium 550 mg tablet oral every 12 hours as needed for back pain 100 tablets 2 refills

Post-coordinated expression with *conceptual graph* [1] syntax:

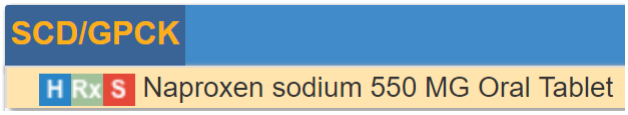
```
[416118004 |Administration (procedure)]
      ->(260686004 |Method (attribute))->[129445006 |Administration - action (qualifier value)]
      ->(363701004 |Direct substance (attribute))->[Rx;849431 Naproxen sodium 550 MG Oral Tablet]
      ->(410675002 |Route of administration (attribute))->[260548002 |Oral (qualifier value)]
```

[1] https://en.wikipedia.org/wiki/Conceptual_graph#Graph-based_knowledge_representation_and_reasoning_model

Notes:

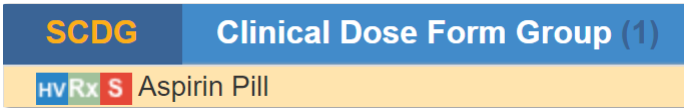
1. The IR is specific enough regarding strength and dose form. Therefore, the RxNorm SCD code can be applied

Figure 7.9. RxNorm SCD Code



2. Other medication requests or performances are less specific. The IR might only state “Aspirin tablet”. In these cases, the RxNorm SCDG codes are used:

Figure 7.10. RxNorm SCDG Code



3. If the IR states a class of drugs, e.g. “Glucocorticoids”, the coding approach is cascaded:

   → First choice: SNOMED CT concept from the “product” hierarchy

   → Second choice: NDF-RT code

4. “Route of administration - oral” is included in the post-coordinated expression. Although the RxNorm code includes “oral tablet” it does not sufficiently capture, that this tablet is administered orally.

5. The “Rx;” prefix for the RxNorm code in the post-coordinated expression indicated the terminology standard. Current modeling guideline: All concepts are SNOMED CT concepts, unless otherwise stated.

6. The IR example states: Naproxen sodium 550 mg tablet oral every 12 hours as needed for back pain 100 tablets 2 refills. Although it is not explicitly stated, the currently agreed upon policy is to interpret this as: 1 tablet at a time.

Coding guidelines for dosage, frequency, total number of tablets and refills etc. will be discussed in later sections. This detailed information is typically only included in medication requests, while performances typically only document that the medication has been taken as a “History of….” Statement.

7.7.9. Non-Medication Procedures (Request and Performance)

Other procedures in the “topic” field, e.g. diagnostic procedures, therapeutic procedures, consults or observational procedures are coded as pre-coordinated or post-coordinated expressions using SNOMED CT concepts.

For IRs (either request or performance) a “simple” procedure, e.g. “Echocardiogram”, entering the procedure code “40701008 |Echocardiography (procedure)|” in the topic field sufficiently captures the IR.

For more complex IRs, particularly where body sites or lateralities are included, some principles to ensure consistency in the modeling must be applied.

1. Always post-coordinate, when “laterality” is involved

- There are many pre-coordinated SCT concepts, which include body site and laterality, e.g. “1451000087102 |Computed tomography of right lower limb (procedure)|”, but not all body sites in SCT are lateralized.
  - To achieve consistency in the modeling approach, instead of using the pre-coordinated concept above, post-coordinate the body structure and the laterality:

    [241570001 |Computed tomography of lower limb (procedure)]-

    ->(363704007 |Procedure site (attribute))

    ->[61685007 |Lower limb structure (body structure)]- ->(272741003 |Laterality (attribute))->[24028007 |Right (qualifier value)];

2. For IRs without involving laterality, the choice for coding the topic is cascaded:
   a. 1st choice: existing pre-coordinated concept
   b. 2nd choice: post-coordinated expression, using existing concepts within the constraints of the concept model
   c. 3rd choice: post-coordinated expression, using existing concepts outside the constraints of the concept model, after discussion and approval
   d. 4th choice: new SCT HSPC Solor extension precoordinated concept, after discussion and approval; use generated UUID until the concept is created

7.7.10. Observational Procedures (Performance)

In the “performance” tab of TSRs, many of the IRs pertain to the documentation of findings or disorders. These are “observational” procedures, often documented within “history and physical” sections of documentation templates, which describe the presence or absence of a finding or disorder.

This category of IRs is always captured as a post-coordinated expression in the topic field.

Example IR: Weakness of neck

Post-coordination:

```
[a997cc03-3e99-40eb-833a-6374c7750a3a |Observation procedure (procedure)]
        ->(363702006 |Has focus (attribute))->[249931001 |Weakness of neck (finding)]
```

Example IR: Right arm pain

Post-coordination:

```
[a997cc03-3e99-40eb-833a-6374c7750a3a |Observation procedure (procedure)]-
        ->(363702006 |Has focus (attribute))->[22253000 |Pain (finding)]-
                    ->(363698007 |Finding site (attribute))->[53120007 |Upper limb structure (body structure)]-
                                ->(272741003 |Laterality (attribute))->[24028007 |Right (qualifier value)];
```

7.7.11. Unstructured (Request and Performance)

Format: Plain text

Currently used to capture textual information for which there is no model at this time.

7.7.12. statementAssociation.semantic (Request and Performance)

Format: Logical Expression

Terminology: TBD Currently not in use

7.7.13. statementAssociation.statementId (Request and Performance)

For use by KNART developers.

7.7.14. Timing (Request and Performance)

The “timing” circumstance has six components:

1. timing.lowerBound

   Format: Number (“float”)

2. timing.upperBound

   Format: Number (“float”)

3. timing.includeLowerBound

   Format: TRUE or FALSE (“Boolean”)

4. timing.includeUpperBound

   Format: TRUE or FALSE (“Boolean”)

5. timing.resolution (optional)

   Format: Number (“float”)

6. timing.measureSemantic

   Format: ISO 8601 Date/Time Format

Timing is used to capture a time or time range for

- Requests for action at a future time
- Performance of action, which has taken place in the past (including “History of X….)

The timing is always expressed as a time or time range relative to the statement time, using the ISO 8601 Date/Time Standard format[2].

If the actual time or time range is not specified in the IR, the following expressions are used:

- ISO 8601 prior to statement time

[2] https://en.wikipedia.org/wiki/ISO_8601

- ISO 8601 following statement time

If the time or time range is specified in the IR, the expression also follows the ISO 8601 Standard, using the appropriate prefixes for periods of time:

- P for period
- M for months
- W for weeks
- Y for years

Using additional fields in the timing circumstance depends upon the degree of specificity within the IR.

Example (unspecific): History of breast cancer

Table 7.2. Timing - unspecific

timing.lowerBound	1
timing.upperBound	inf
timing.includeLowerBound	TRUE
timing.includeUpperBound	FALSE
timing.resolution	
timing.measureSemantic	ISO 8601 prior to statement time

The IR implies:

- Breast cancer was present in the patient's history = timing.lowerBound = 1
- No time range specified = timing.upperBound = inf (infinite)
- There was at least 1 instance = timing.includeLowerBound = TRUE
- "upper bound" is infinite = timing.includeUpperBound = FALSE ("inf" is never included!)
- IR does not specify units of time, e.g. years, months = timing.resolution = blank

Note: The expression of "present" could also be correctly indicated using

timing.lowerBound = 0

timing.includeLowerBound = FALSE

Not including "0" also expresses that there has to be at least "1". However, it is the current agreed policy to use the "1/TRUE" option.

Example (specific range): Anticonvulsant therapy greater than 2 years

Table 7.3. Timing - specific range

timing.lowerBound	24M
timing.upperBound	inf
timing.includeLowerBound	FALSE

timing.includeUpperBound	FALSE
timing.resolution	1M
timing.measureSemantic	ISO 8601 prior to statement time

The IR expresses:

- Anticonvulsant therapy for more than 2 years (24 months) was present in the patient's history = timing.lowerBound = 24M
- No upper time limit specified = timing.upperBound = inf (infinite)
- There was anticonvulsant therapy for more than 24 months = timing.includeUpperBound = FALSE
- Timing.measureSemantic = ISO 8601 prior to statement time
- timing.resolution field:
  - This field is optional, but if a time or time range is specified, the resolution has to be specified.
  - The use depends on the desired granularity of the time increments
  - Some of the reasoning about how to use these fields depends on the clinical relevance.

Example (specific date): Completed Appointed on March 12, 2018 with Cardiology

Table 7.4. Timing - specific date

timing.lowerBound	2018-03-19T12:01
timing.upperBound	2018-03-19T23:59
timing.includeLowerBound	TRUE
timing.includeUpperBound	TRUE
timing.resolution	
timing.measureSemantic	ISO 8601

Note: ISO 8601 uses the 24 hour standard for time of day.

7.7.15. Purpose (Request and Performance)

Format: Logical Expression

Terminology: SNOMED CT

The "purpose" field is used to capture WHY a procedure was requested or performed in a post-coordinated expression, based on two possible procedures:

Evaluation procedure: 386053000 |Evaluation procedure (procedure)|

Therapeutic procedure: 277132007 |Therapeutic procedure (procedure)|

The procedure is refined by post-coordinating with a "363702006 |Has focus (attribute) |" attribute and identifying a finding/disorder or procedure concept as the value for the attribute.

Example IR: Resting 12-lead electrocardiogram to evaluate for arrhythmia

```
[386053000 |Evaluation procedure (procedure)]
->(363702006 |Has focus (attribute))->[ 698247007 |Cardiac arrhythmia (disorder)]
```

Example IR: Naproxen sodium 550 mg tablet oral every 12 hours as needed for back pain 100 tablets 2 refills

```
[277132007 |Therapeutic procedure (procedure)]
->(363702006 |Has focus (attribute))->[161891005 |Backache (finding)]
```

IRs can have more than one purpose.

7.7.16. requestedResult (Request and Performance)

The “requestedResult” circumstance has eight components:

1. requestedResult.lowerBound

   Format: Number (“float”)

2. requestedResult.upperBound

   Format: Number (“float”)

3. requestedResult.includeLowerBound

   Format: TRUE or FALSE (“Boolean”)

4. requestedResult.includeUpperBound

   Format: TRUE or FALSE (“Boolean”)

5. requestedResult.resolution (optional)

   Format: Number (“float”)

6. requestedResult.measureSemantic

   Format: Logical Expression

7. requestedResult.healthRisk

   Format: Logical Expression

8. requestedResult.status

   Format: Logical Expression

The “requestedResult” fields 1 – 6 above are used to capture IRs, which

- enumerate what is being requested, e.g. Administration of a medication **1 tablet at a time**
- specify the intended outcome of an action, e.g. Administration of Metoprolol to **achieve systolic BP < 130 mmHg**

Example IR: Metoprolol tartrate 50 mg tablet oral daily 2 times

Table 7.5. requestedResult -Example 1

requestedResult.lowerBound	1
requestedResult.upperBound	1
requestedResult.includeLowerBound	TRUE
requestedResult.includeUpperBound	TRUE
requestedResult.resolution	
requestedResult.measureSemantic	421026006 |Oral tablet (qualifier value)|

Note: This should not be confused with “frequency”. Although not stated explicitly, it is understood that the IR states: ONE tablet, twice a day.

Example IR: Acetaminophen 325 mg tablet oral two tablets every 6 hours

Table 7.6. requestedResult -Example 2

requestedResult.lowerBound	2
requestedResult.upperBound	2
requestedResult.includeLowerBound	TRUE
requestedResult.includeUpperBound	TRUE
requestedResult.resolution	
requestedResult.measureSemantic	421026006 |Oral tablet (qualifier value)|

7.7.17. conditionalTrigger (Request)

Format: Logical Expression

Terminology: TBD

Currently not in use.

7.7.18. conditionalTrigger.statementId (Request)

UUID as identifier for the conditionalTrigger statement.

7.7.19. Priority (Request)

Format: Logical

Expression Terminology: SNOMED CT

The priority field captures the standard priorities associated with a request for action, e.g. stat, routine

7.7.20. repetition.period (Request)

The “repetition.period” has twelve components. Six components for the repetition period start and six components for the repetition period duration. The fields are used to capture WHEN a repeated action should start and HOW LONG the requested action should be repeated.

1. repetition.periodStart.lowerBound

   Format: Number (“float”)

2. repetition.periodStart.upperBound

   Format: Number (“float”)

3. repetition.periodStart.includeLowerBound

   Format: TRUE or FALSE (“Boolean”)

4. repetition.periodStart.includeUpperBound

   Format: TRUE or FALSE (“Boolean”)

5. repetition.periodStart.resolution (optional)

   Format: Number (“float”)

6. repetition.periodStart.measureSemantic

   Format: Logical Expression

7.7.21. repetition.period components

Example IR: Naproxen sodium 550 mg tablet oral every 12 hours as needed for back pain

Table 7.7. repetition.period Example

repetition.periodStart.lowerBound	[NOW,NOW] relative to statement time
repetition.periodStart.upperBound	
repetition.periodStart.includeLowerBound	
repetition.periodStart.includeUpperBound	
repetition.periodStart.resolution	
repetition.periodStart.measureSemantic	
repetition.periodDuration.lowerBound	1
repetition.periodDuration.upperBound	inf
repetition.periodDuration.includeLowerBound	TRUE
repetition.periodDuration.includeUpperBound	FALSE
repetition.periodDuration.resolution	1
repetition.periodDuration.measureSemantic	258703001 |day (qualifier value)|

If the IR does not explicitly state a period start time, the default entry in this field is “[NOW,NOW] relative to statement time”.

Note: “[NOW,NOW]” is not to be confused with priority “stat”. The “NOW” is simply used, where there is not a specified time, e.g. 1 week from now.

If a repetition period start/stop time is specified, the “upper/lower bound” components and the measure-Semantic are used as in all other timing related circumstances.

7.7.22. repetition.periodDuration components

Every repetition has a duration, even if it is not explicitly stated in the IR. In the example above, the IR states a frequency (every 12 hours), but not a duration. In these cases it is understood that the duration is “infinite”. The same understanding is true for IR statements described as “daily”. The “upper/lower bound” components and the “measure.semantic” are used in the same way as in all other timing related circumstances.

Note: The “repetition.periodDuration” fields are currently also used to capture numbers of tablets (or other units) and number of refills, if these are stated in the IR. The tablets/refills are used to calculate how long the administration period can be.

Example IR: Aspirin 81 mg oral tablet daily as needed, 30 tablets, 3 refills

30 tablets + 3 refills = 120 tablets

1 tablet/day = 120 days

Table 7.8. repitition.periodDuration components Example

repetition.periodDuration.lowerBound	1
repetition.periodDuration.upperBound	120
repetition.periodDuration.includeLowerBound	TRUE
repetition.periodDuration.includeUpperBound	TRUE
repetition.periodDuration.resolution	1
repetition.periodDuration.measureSemantic	258703001 |day (qualifier value)|

7.7.23. repetition.eventFrequency (Request)

This circumstance is used to capture the requested frequency of any repeated action, e.g. 3 times/day, once/week.

The “repetition.eventFrequency” circumstance has six components.

1. repetition.eventFrequency.lowerBound

   Format: Number (“float”)

2. repetition.eventFrequency.upperBound

   Format: Number (“float”)

3. repetition.eventFrequency.includeLowerBound

   Format: TRUE or FALSE (“Boolean”)

4. repetition.eventFrequency.includeUpperBound

   Format: TRUE or FALSE (“Boolean”)

5. repetition.eventFrequency.resolution (optional)

   Format: Number (“float”)

6. repetition.eventFrequency.measureSemantic

   Format: Logical Expression

Example IR: Naproxen 550mg tablet oral every 12 hours

Table 7.9. repetition.eventFrequency - Example 1

repetition.eventFrequency.lowerBound	12
repetition.eventFrequency.upperBound	12
repetition.eventFrequency.includeLowerBound	TRUE
repetition.eventFrequency.includeUpperBound	TRUE
repetition.eventFrequency.resolution	
repetition.eventFrequency.measureSemantic	258702006 |hour (qualifier value)|

Example IR: Ibuprofen 400 mg tablet oral every 6 hours; may increase dose frequency to one tablet every 4 hours

Table 7.10. repetition.eventFrequency - Example 2

repetition.eventFrequency.lowerBound	4
repetition.eventFrequency.upperBound	6
repetition.eventFrequency.includeLowerBound	TRUE
repetition.eventFrequency.includeUpperBound	TRUE
repetition.eventFrequency.resolution	
repetition.eventFrequency.measureSemantic	258702006 |hour (qualifier value)|

The “upper/lower bound” components and the measureSemantic are used as in all other timing related circumstances.

7.7.24. repetition.eventSeparation (Request)

Currently not in use.

7.7.25. repetition.eventDuration (Request)

This circumstance will be used to capture, HOW LONG each requested event should last, e.g. “Physical therapy 3 times per week for 1 hour.

Currently not in use.

8. Clinical Input Form Statements

Ideally, clinical information is represented in a manner that is most efficient for use. The problem is that users have many different requirements for clinical information, thus no single representation can be the most efficient representation for all the various use cases. Thus, maximum efficiency for each use case necessitates that any particular clinical information be available in multiple representations. Although different in form, each of these different representations semantically model the same information. These are known as isosemantic models.

A clinical statement represents an entry in the patient record that documents in a structured/computable manner clinical information related to the patient that is asserted by a particular source, recorded, and potentially verified.

Clinicians author clinical statements and enter them into their organization's electronic health record (EHR). Clinicians typically enter the information via a manner that we call here the clinical input form (CIF). However, the CIF is not a literal form that clinicians select and enter data in. Rather, it refers to the manner in which information is presented to the clinicians and how they enter the data, such as by constraining the information to allow only certain values to be entered, such as through a drop-down list or radio button, or breaking up large chunks of related information into smaller parts. For example, when a clinician orders a medication, rather than selecting this information all at once with a single item, they will choose the various parts of the medication order, such as:

- Kind of drug and strength (e.g., Acetaminophen 150 mg)
- Amount and how often the patient should take the medication (e.g., 1 tablet twice daily)
- Duration (e.g., 2 days)
- Any constraints (e.g., do not exceed a total daily dosage of 600 mg)

Ideally, the way the information is presented to clinicians is in a manner that is most efficient for clinicians to select and enter data may not be the most efficient way for data analysts to use when they are querying data once it has been normalized and stored in a database, such as when creating a new CDS rule or compiling prevalence statistics. For this, the data is normalized using the analysis normal form (ANF) and stored in a database. Again, the ANF is not necessarily a physical structure, but is how a data analyst might see the data when they are looking at it in a database, and not as clinicians would see it in the user interface (i.e., CIF).

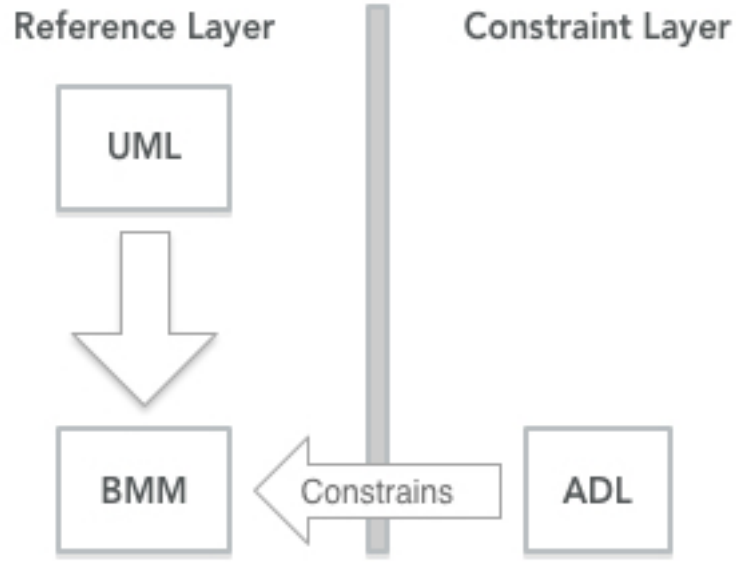
As a forward to this discussion it is necessary to provide some historical background about the Clinical Information Modeling Initiative (CIMI) model. The CIMI working group created a reference model with no working knowledge of a division between analysis normal form and clinical input form. The model they created was developed along standard lines of informatics thinking and thus ended up being a CIF model because CIF models are the norm in informatics. Thus, CIMI simply called this model the CIMI model. But now to distinquish it from the ANF model being proposed to CIMI, we will call the current CIMI model, the CIMI CIF model.

8.1. Basics of the CIMI Clinical Input Form

The CIMI CIF Model consists of two layers as shown in Figure 8.1, "CIMI CIF Model Layers". A reference model layer that defines the structural classes and named attributes, and a constraint layer which constrains these structural attributes by value, subtype, cardinality, and terminology. The basic modeling rule that CIMI CIF follows is: new named attributes are added in the Reference Layer and the constraining of existing attributes occurs in the Constraint Layer.

The CIMI CIF Reference Model layer is authored using Unified Modeling Language (UML). These class definitions may be viewed at *http://models.opencimi.org/cimi_doc/*.

Figure 8.1. CIMI CIF Model Layers



The constraint layer is described using Archetype Definition Language (ADL). ADL is a formal language with a textual syntax for describing constraints on the classes described in the reference layer. A re-usable formal constraint model defined in ADL is called an Archetype. The full collection of CIMI CIF Archetypes may be viewed at *http://models.opencimi.org*.

One complexity that needs to be addressed here is that ADL can only be used to constrain reference classes defined in a lightweight proprietary UML-like specification called Basic Meta-Model (BMM). For this reason, CIMI has developed tooling that transforms the CIMI UML models into the BMM specification. Although this complexity does exist, to ease understanding, the reader can simply imagine that ADL is directly constraining the UML classes.

The UML/BMM classes are more abstract and the archetypes are where specific semantics such as 'blood glucose' or 'diabetes present,' are asserted.

8.1.1. Structures

The CIMI UML/BMM model has three concentric layers: a Core that defines datatypes and a root class, a Foundation that describes compositional patterns similar to ISO 13606, and a Clinical model layer constructed on top of the Foundation.

Most clinical specifications will be based on the Clinical Statement pattern defined in the Clinical model layer. But this pattern does employ structures built out of Foundation and Core classes, so familiarity with these layers will be helpful. For more information consult the CIMI Architecture Guide[1].

8.2. Clinical Statement Pattern

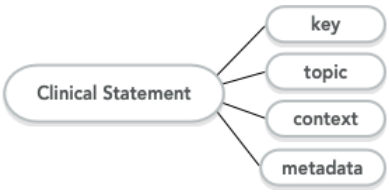
The central focus of the CIMI Reference Model is the Clinical Statement. A Clinical Statement represents structured electronic communication made about a patient typically documented as an 'entry' in the patient record. For example, Clinical Statement can be used to represent the following statements made about a patient.

[1] http://wiki.hl7.org/images/e/e6/2007-01_CIMI_Modeling%2C_Architecture_Methodology_and_Style_Guide_.pdf

- Patient has a diagnosis of congestive heart failure.
- Patient has a family history of breast cancer.
- Patient has a goal of smoking cessation.
- Patient has an order for Physical Therapy.
- Patient has a lab result of Serum Sodium equals 130 mEq/L with delta flag.
- Patient had an appendectomy.
- Patient has a paternal uncle with Hemophilia C.

Clinical Statement, shown in Figure 8.2, “Clinical Statement”, has a ‘key’, ‘topic’, ‘context’, and ‘metadata’. The ‘key’ is the terminology meaning binding for the entire Clinical Statement. The ‘topic’ is the clinical entity being described. The ‘context’ describes the circumstances that form the setting in which the ‘topic’ should be evaluated. Finally, ‘metadata’ is the collection of metadata that is associated with the clinical statement: the who, where, why and when information.

Figure 8.2. Clinical Statement



Topic The ‘topic’ is the clinical entity described by the Clinical Statement. A few examples of topic include clinical assertions, evaluation results, and procedures. For each of these topics the information described is quite different. Therefore, CIMI describes topic types that contain the appropriate attributes to describe the required information for the given topic. The number of topic types will change as CIMI progresses. Currently the allowable topic types are listed here.

- ProcedureTopic
- FindingTopic
  - EvaluationResultTopic
  - AssertionTopic

Context The ‘context’ describes the circumstances that form the setting in which the ‘topic’ should be evaluated. CIMI describes context types that contain the appropriate attributes to describe the required information for the given context. The number of context types will change as CIMI progresses. Currently the allowable context types are listed here.

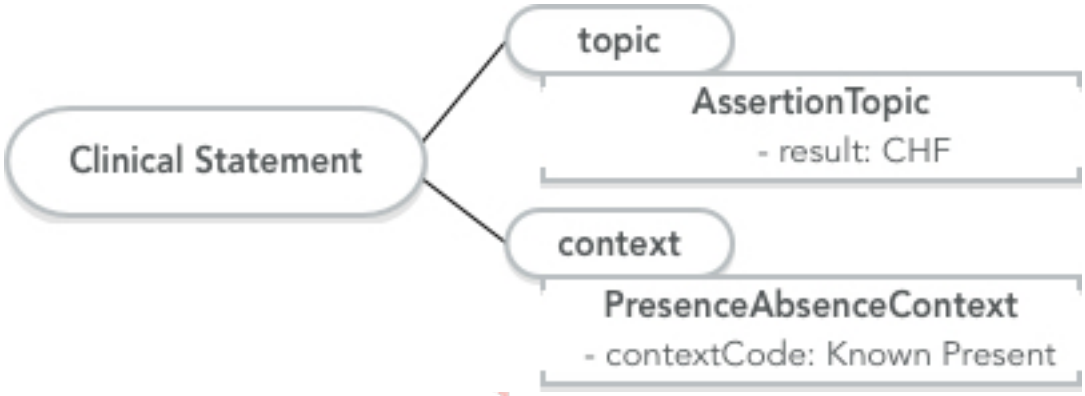
- ActionContext
  - RequestContext
  - OrderContext
  - PerformanceContext
- FindingContext
  - PresenceContext
  - AbsenceContext
  - GoalContext

Metadata ‘Metadata’ is not actually an attribute of ClinicalStatement, but is intended here to represent the various attributes in a clinical statement that represent metadata about the clinical statement. This includes attribution information relating to the statement itself such as who authored, verified, recorded, or signed the statement or more informally, the who, where, why, and when information. Other attributes include 'subject of record' and 'subject of information'.

8.2.1. Examples Using Topic and Context

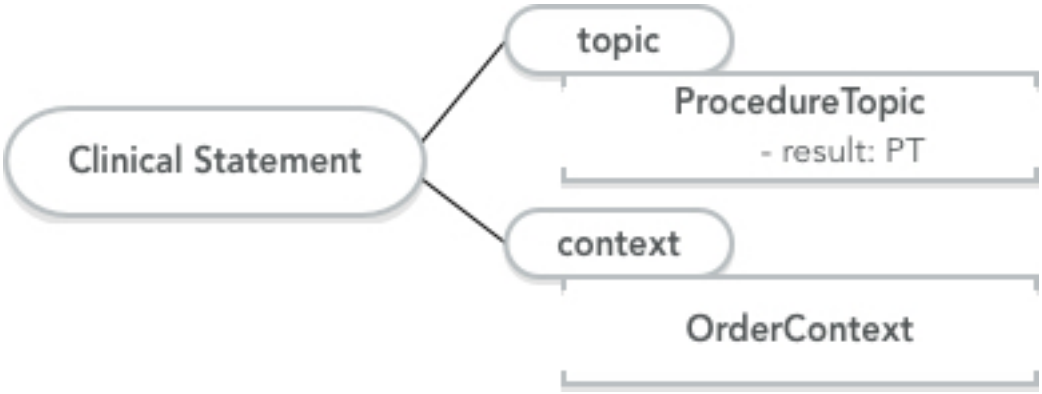
Earlier, descriptive examples of Clinical Statements were given. Here we will represent a few of these examples using the Clinical Statement 'topic - context' paradigm. In Figure 8.3, "Patient has diagnosis of congestive heart failure.", the example for "Patient has diagnosis of congestive heart failure" is illustrated. The topic has been declared to be of type AssertionTopic stating "assertion of congestive heart failure", and the context has been declared to be of type PresenceAbsenceContext stating "Known Present". What may not be apparent in the figure is that when the topic is declared to be of type AssertionTopic, then all the attributes of AssertionTopic are available for use. However, in the figure only the attribute named 'result' is shown for clarity.

Figure 8.3. Patient has diagnosis of congestive heart failure.



In Figure 8.4, "Patient has an order for Physical Therapy.", the example for "Patient has an order for Physical Therapy." is shown. The topic has been declared to be of type ProcedureTopic stating "procedure of type physical therapy", and the context has been declared to be of type OrderContext. Again, the majority of attributes for ProcedureTopic and OrderContext are not shown for clarity.

Figure 8.4. Patient has an order for Physical Therapy.



StatementTopic and StatementContext are both collections of attributes and have the following characteristics:

1. They are reusable components that can be assembled to form clinical statements. For instance, one can coordinate the ProcedureTopic with the ProposalContext to represent a ProcedureProposal statement. Alternatively, ProcedureTopic may be paired with OrderContext to create a ProcedureOrder statement.

2. They represent groupings of attributes aligned with the SNOMED Clinical Terms (SNOMED CT) Concept Model. For instance, ProcedureTopic is aligned with the SNOMED CT Procedure Concept Model. PerformanceContext aligns with the Situation with Explicit Context (SWEC) Concept Model.

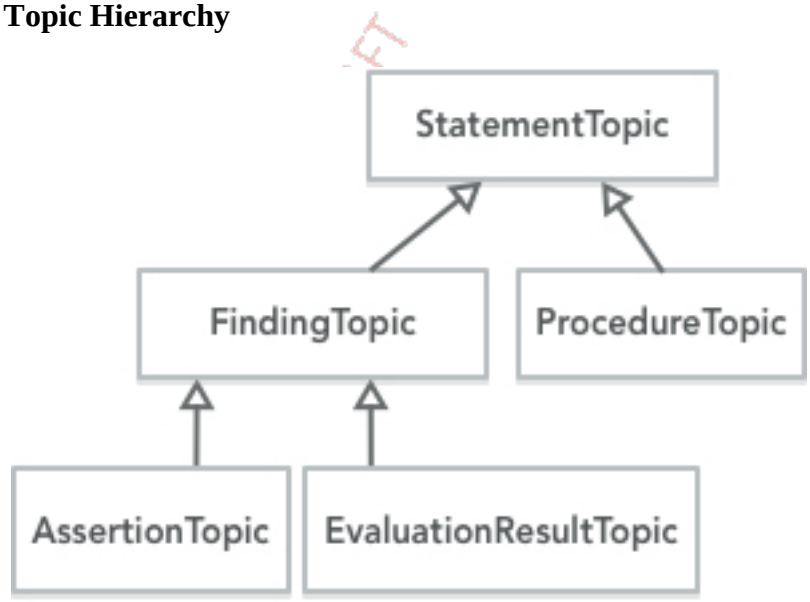
3. They provide for a mechanism to state presence or absence of a finding as well as performance or non-performance of an action. For instance, the pairing of ProcedureTopic with NonPerformanceContext allows for the expression of a procedure that was not performed.

8.3. Topic Patterns

Topic Patterns include all the attributes required to fully describe a clinical entity. The main topic pattern categories CIMI has developed to date include FindingTopic and ProcedureTopic, with FindingTopic having children of AssertionTopic and EvaluationResultTopic. They are shown in Figure 8.5, “Topic Hierarchy” and are described in the following sections. Each of these topic subtypes contain a collection of attributes that describe the given pattern. These patterns provide the foundational structure for detailed clinical model (DCM) archetype instances that can be visualized at *http://models.opencimi.org*

It should be noted that topics shown in Figure 8.5, “Topic Hierarchy” are further subtyped and AssertionTopic, EvaluationResultTopic, and ProcedureTopic are the main branching points that we will cover in the next sections of this document. These are the branch points from which further topic subtypes will be created. The attributes inherited from FindingTopic and StatementTopic are shown as if they exist in AssertionTopic, EvaluationTopic, and ProcedureTopic.

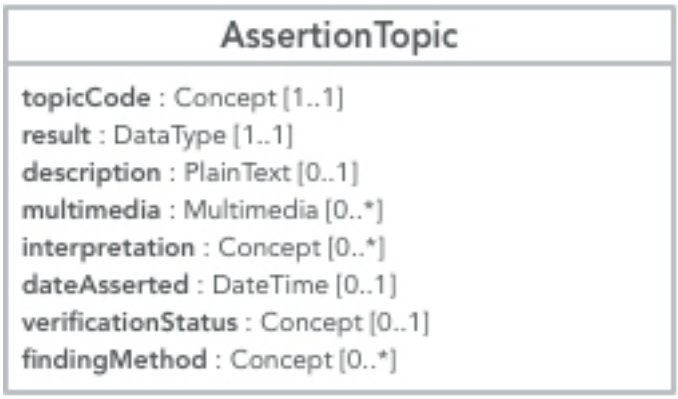
Figure 8.5. Topic Hierarchy



8.3.1. AssertionTopic

The first topic type we will describe is the AssertionTopic pattern with its included attributes, as shown in Figure 8.6, “AssertionTopic”. Many other topic patterns can then be subclassed from AssertionTopic. One example of this is ConditionTopic, shown in Figure 8.7, “ConditionTopic” which is a child of AssertionTopic and is used to represent a condition in a patient. ConditionTopic adds attributes such as clinicalCourse, severity, and diseasePhase that help to further describe conditions. If these additional attributes are unnecessary, then AssertionTopic can be used rather than ConditionTopic.

Figure 8.6. AssertionTopic



Note in the diagram, for simplicity, ConditionTopic is shown with the attributes it inherits from AssertionTopic.

Figure 8.7. ConditionTopic

The class AssertionTopic or ConditionTopic could be constrained as part of a ClinicalStatement to:

- assert the presence of chest pain.
- assert the absence of chest pain.

- assert the presence of edema.

The assertion pattern for a clinical statement is as follows:

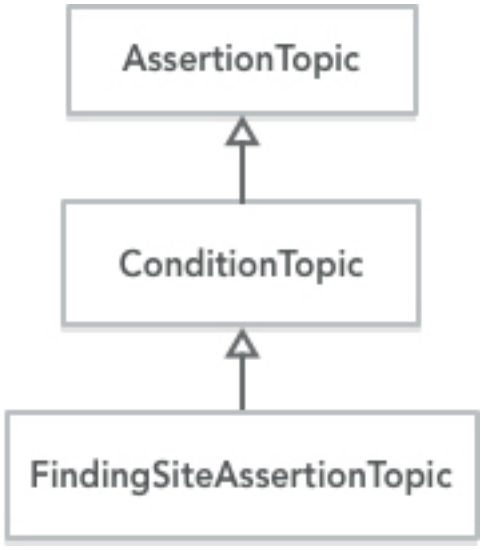
- topic.topicCode = a code meaning “assertion”.
- topic.result = a code representing what is being asserted e.g., a code for “rash”, “auto accident”, “hypertrophy”, etc.).

8.3.1.1. Assertion Hierarchy

The full hierarchy for AssertionTopic is shown in Figure 8.8, “Assertion Hierarchy”. AssertionTopic serves two important purposes: (1) it provides the core set of assertion attributes that are relevant in assertion of presence and absence; and (2) it is the parent type for the more specific assertions such as ConditionTopic and FindingSiteAssertionTopic. If additional attributes are identified as needed to properly model assertions, they would either be added to one of the existing assertion types or a new type could be created with these attributes. This modeling decision would be based on whether adding these attributes make sense for existing assertion types or whether they should be used to create a new subset of assertions. Typically an attribute is added to the parent class if that attribute is relevant in all the subclasses derived from the parent class. If an attribute is only relevant in some of the subclasses, then the attribute is introduced in those subclasses. This ensures that a class does not have an attribute that is incongruent and thus requires that attribute to be occasionally constrained out. For instance, it is viewed as bad practice to create an Animal class that contains arms, legs, and wings and then create a subclass of dog that constrains out wings since dogs do not have wings.

Note there are two ways to introduce an attribute that is not always used. A UML class specialization specifies a new class that has all of the attributes of its parent and may then specify additional attributes. An archetype may choose to use whichever class, parent or child, is appropriate. Or, the additional attribute may be added to the original class and the archetype may then use the attribute or "constrain it out" by setting its cardinality to zero. As previously stated, CIMI modelers prefer the first approach, extension through UML class specialization, that avoids the need to constrain elements out of archetypes.

Figure 8.8. Assertion Hierarchy



8.3.1.2. Assertions

Assertions affirm or deny the existence of clinical conditions, diseases, symptoms, etc. As just described, different varieties of assertion may extend an existing AssertionTopic class with any additional attributes

necessary to fully represent this new group of assertions. Example 8.1, “The patient has diabetes mellitus type 1 which was diagnosed at age 24” and Example 8.2, “The patient does not have diabetes mellitus type 1” show examples of clinical statements using the AssertionTopic class for the topic, and Example 8.3, “The patient has a femur fracture in the right leg” and Example 8.4, “The patient has a stage two pressure injury on the right ischial tuberosity” show examples of clinical statement using FindingSiteAssertion-Topic for the topic. These examples show the ‘topic.topicCode’, ‘topic.result’, and ‘context.contextCode’ for each, with the addition of any extra attributes from the chosen topic needed to describe the clinical statement. Context will be discussed in depth later in this document. For now, be aware the chosen context is a full class with many attributes but here we are only showing the context code attribute that is common to all context types.

Example 8.1. The patient has diabetes mellitus type 1 which was diagnosed at age 24

```
DiabetesMellitusAssert
    topic.topicCode: Assertion
    topic.result: Diabetes mellitus type 1 (disorder)
    topic.ageAtOnset: 24 years
    context.contextCode: Confirmed present (qualifier value)
```

Example 8.2. The patient does not have diabetes mellitus type 1

```
DiabetesMellitusAbsentAssert
    topic.topicCode: Assertion
    topic.result: Diabetes mellitus type 1 (disorder)
    context.contextCode: Known absent (qualifier value)
```


Note, in the CIMI alignment with the SNOMED CT concept model, the AssertionTopic pattern corresponds to the Finding hierarchy as inflected by the Situation hierarchy.

Note AssertionStatement.topic.topicCode is not part of this construction. It is modeled with the fixed term “assertion” and is as semantically inert as we can manage.

Other attributes may also inflect the semantics; e.g., an AssertionStatment.topic.findingMethod that would align with the concept model’s Finding.findingMethod.

8.3.1.3. Finding Site Assertions

A FindingSiteAssertionTopic is an assertion about a finding found on the body. This assertion is a “design by extension” assertion because it contains the additional attribute findingSite that is used to capture the body site affected by the condition. The FindingSiteAssertionTopic encourages post-coordination as shown in examples 3 and 4, and intentionally aligns with the SNOMED CT Clinical Findings concept model.

Example 8.3. The patient has a femur fracture in the right leg

```
FractureAssert
    topic.topicCode: Assertion
    topic.result: Fracture of bone (disorder)
```

```
    topic.findingSite.code: Bone structure of femur
    topic.findingSite.laterality: Right (qualifier value)
    context.contextCode: Confirmed present (qualifier value)
```

Example 8.4. The patient has a stage two pressure injury on the right ischial tuberosity

```
WoundAssert
    topic.topicCode: Assertion
    topic.result: Pressure ulcer stage 2 (disorder)
    topic.findingSite.code: Skin structure of ischial tuberosity
    topic.findingSite.laterality: Right (qualifier value)
    context.contextCode: Confirmed present (qualifier value)
```

8.3.2. Evaluation Result

The second topic pattern we will discuss is EvaluationResultTopic which is used to document a characteristic of a patient or a clinical value being observed. An EvaluationResultTopic may hold the code for a test in the 'topicCode' attribute (e.g., code for "heart rate evaluation", "serum glucose lab test", etc.) and the resulting value of the test in the 'result' attribute. Viewed another way, the EvaluationResultTopic topicCode holds a question (e.g., "what is the heart rate?", "what is the serum glucose?") and the 'result' holds the answer. Any clinical statement such as a laboratory test, a vital sign, or a questionnaire question that fits this pattern of a question and a resulting value is modeled with the EvaluationResultTopic pattern.

The evaluation result pattern for a clinical statement is as follows:

- topic.topicCode = what's being evaluated ("heart rate", "serum glucose", "breath sound", etc.).
- topic.result = the result of the evaluation ("72 bpm", "100 mg/dL", "rales")

The following is an isosemantic comparison of the evaluation result pattern to the previously described assertion pattern. In the previous section, we illustrated assertion models using rash, auto accident, and hypertrophy. Below we show what these assertion examples would look like if we hypothetically modeled them using the Evaluation Result pattern. Note, CIMI avoids creating models where the 'result' specifies "presence/absence" or "yes/no", so this is a clear indicator that the assertion pattern is preferred in these cases.

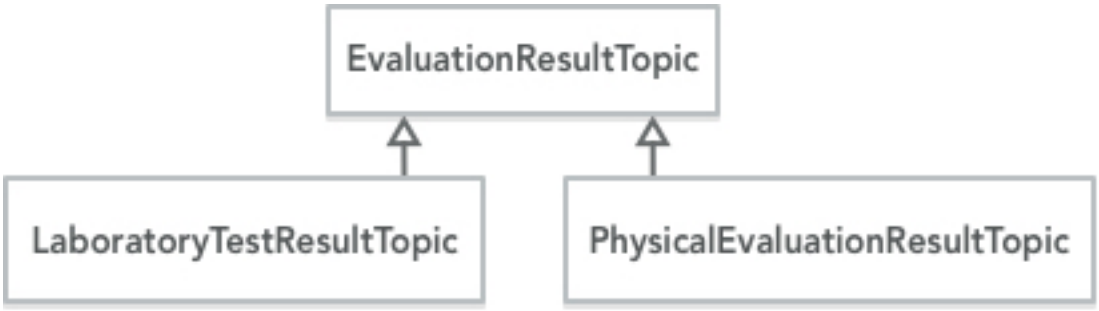
Assertion	• topic.topicCode = a code meaning "assertion" • topic.result = a code representing what's being asserted ("rash", "auto accident", "hypertrophy", etc.)
EvaluationResult (This is hypothetical)	• topic.topicCode = what's being evaluated ("rash", "auto accident", "hypertrophy", etc.) • topic.result = "present" or "yes"

Like Assertion, Evaluation Result corresponds to the SNOMED CT concept model. The EvaluationResultStatement.topic.topicCode attribute corresponds to the observation being evaluated.

8.3.2.1. Evaluation Result Hierarchy

EvaluationResultTopic currently has two subtypes; LaboratoryTestResultTopic (that includes additional attributes necessary to describe laboratory tests) and PhysicalEvaluationResultTopic.

Figure 8.9. Evaluation Result Hierarchy



8.3.2.2. Modeling in the Constraint Layer

This section will use LaboratoryTestResultTopic, which exists in the Reference Model Layer, to further describe modeling in the Constraint Layer. There are different categories of laboratory tests that differ in their resulting data type, such as quantitative labs and nominal labs, where the former would have a QUANTITY result and the latter would have a CODED_TEXT result. For the different lab categories, there is not a need for new named attributes, rather, only a need to constrain the result to the appropriate datatype. The modeler has a choice to make in this situation as the datatype could be constrained in a new class subtype in the reference layer or as an archetype in the constraint layer. Since a new named attribute is not required, the style CIMI has adopted as the constraint would occur in the constraint layer and an ADL Archetype would be created for both QuantitativeLaboratoryTestResult and NominalLaboratoryTestResult.

8.3.2.3. Evaluation Result Subtypes

LaboratoryTestResultTopic LaboratoryTestResultTopic contains attributes specific to the lab evaluation process. These include information about the physical process (e.g., specimen) plus process management information (e.g., status).

PhysicalEvaluationResultTopic PhysicalEvaluationResultTopic contains attributes specific to the clinical evaluation process. These include information about the physical examination process (e.g., patient position, body site).

Example 8.5. The patient’s skin turgor is friable

```
SkinTurgorEval
    topic.topicCode: Skin turgor (observable entity)
    topic.result: Fragile skin (finding)
    topic.evaluationProcedure: Inspection (procedure)
    context.contextCode: Confirmed present (qualifier value)
```

Example 8.6. The patient's systolic blood pressure is 120 mmHg

```
SystolicBloodPressureEval
    topic.topicCode: Systolic arterial pressure (observable entity)
    topic.result: 120
```

```
    unitsOfMeasure: Millimeter of mercury (qualifier value)
topic.evaluationProcedure: Auscultation (procedure)
context.contextCode: Confirmed present (qualifier value)
```

8.3.2.4. Guideline: Assertion versus Evaluation

In most cases the decision between using the evaluation result pattern and the assertion pattern is intuitive and straightforward. “Urine color”, for example, is clearly best modeled as an evaluation result because the attribute being evaluated is the color of the patient’s urine and the result of the evaluation is the set of codes representing the colors that may be observed. To model urine color as an assertion would require the creation of a large number of pre-coordinated concepts. The key would be “assertion” and result would be populated with a code from a set of codes such as “amber urine” (meaning “the patient has amber urine”), “clear urine”, etc.

However, this highlights that any evaluation model may be transformed into an assertion model. (Conversely, any assertion model may be transformed into an evaluation model.) In the case of urine color, the decision is intuitive. In other cases the decision is less clear.

For example, “heart rhythms” (bradycardic, tachycardic, etc.) may be modeled as multiple assertion models (bradycardia, tachycardia, etc.) or as a “heart rhythms” evaluation model whose data is constrained to a value set (containing “bradycardic”, “tachycardic”, etc.).

The general guideline is if it is natural to think of the concept as a noun, as a condition or a state that exists in the patient, model as an assertion or set of assertions. If the statement about the patient is thought of as a name/value pair (i.e., a noun representing the attribute and an adjective representing the value), such as “hair color” = (“black”, “brown”, “blonde”), then model it as an evaluation. However, it is important to note both styles are allowed and the true determinant of their use is whether a result for a given criteria other than true/false or present/absent is specified.

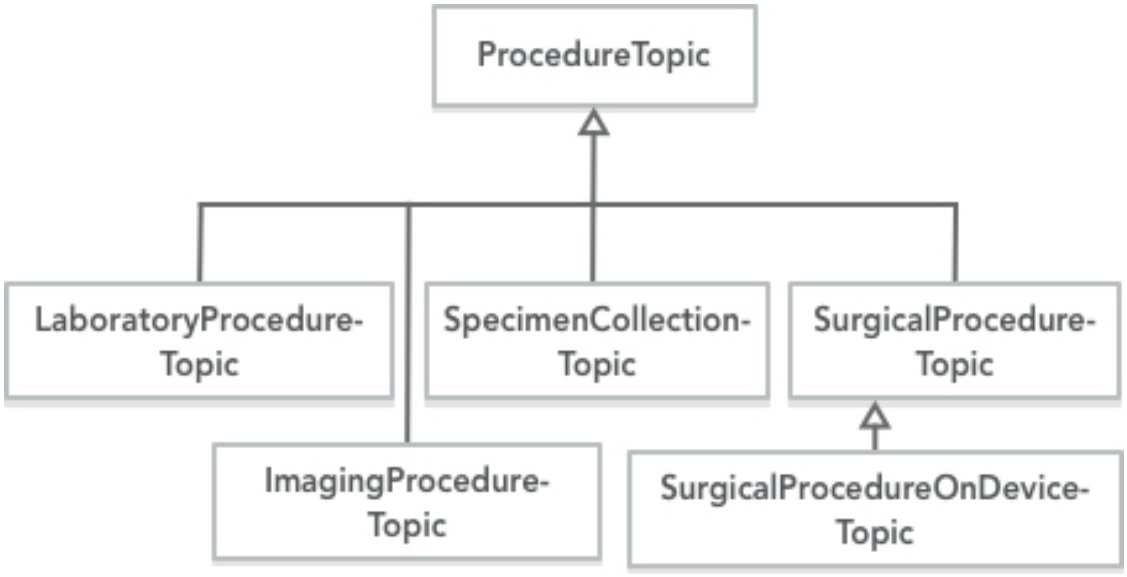
This discussion highlights the importance of isosemantic models. Even if one model or set of models can be agreed upon as the preferred storage model (e.g., assertion models for “bradycardia” and “tachycardia” instead of an evaluation model with “bradycardic” and “tachycardic” as values), inevitably there will be use cases (e.g., data entry, messaging, reporting, etc.) for the other model and a need to identify use cases where different modeling patterns describe semantically identical phenomena. These patterns are isosemantic. An essential (as of now unfulfilled) requirement is for a mechanism of identifying isosemantic models, managing isosemantic groups, and transforming between them. We expect a great deal of this work to be facilitated by the semantic underpinnings of the models supporting the ability to classify the content of two models and determine their logical relations (equivalent, subsumed, disjoint).

It should be noted the Assertion vs. Evaluation topic is solely concerned with the structure and schema pattern used to capture clinical information. Choosing Assertion vs. Evaluation patterns has nothing to do with whether the information being captured is subjective vs. objective.

8.3.3. ProcedureTopic

Procedure models are used to represent actions taken related to the care of a patient such as a cholecystectomy, peripheral IV placement, delivery of a warm blanket, dressing change, ambulation, patient education, etc. The CIMI ProcedureTopic, as shown in Figure 8.10, “Procedure Hierarchy”, is a base class for a number of specializations such as surgical, imaging, and laboratory procedures. The CIMI Procedure Model is aligned with the SNOMED CT Procedure Concept Model when such an alignment exists.

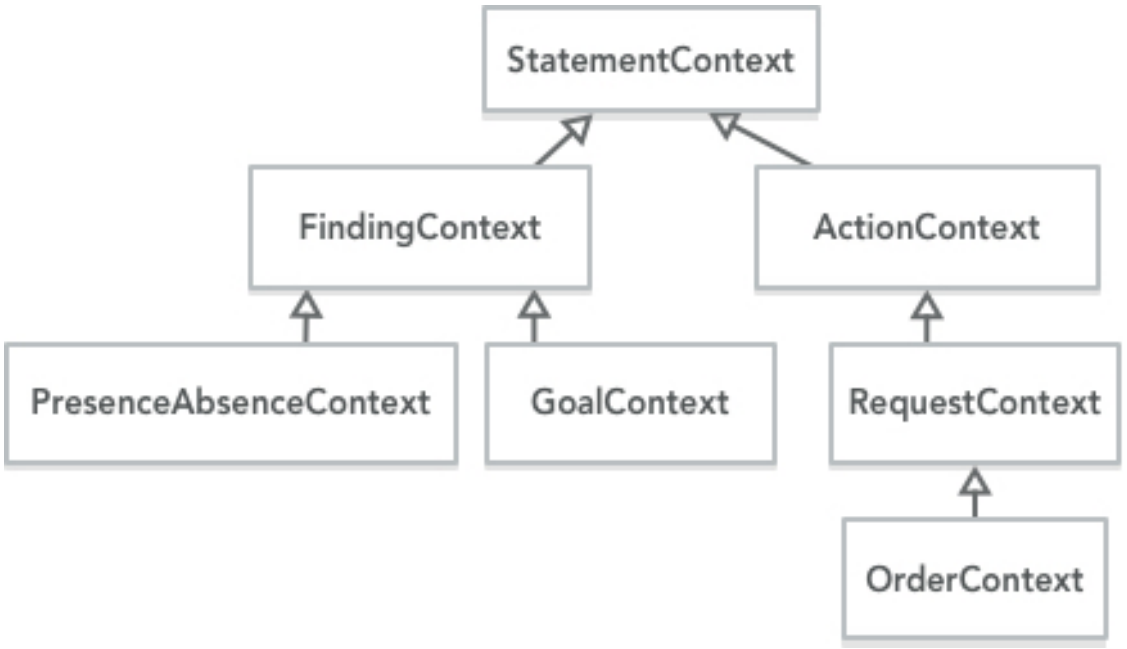
Figure 8.10. Procedure Hierarchy



8.4. Context Patterns

When a Clinical Statement is defined it will be modeled as a combination of a topic and a context. The ‘context’ describes the circumstances that form the setting in which the ‘topic’ should be evaluated. Specializations within the context hierarchy, shown in Figure 8.11, “Procedure Hierarchy”, add important attribution information for the situation being described.

Figure 8.11. Procedure Hierarchy

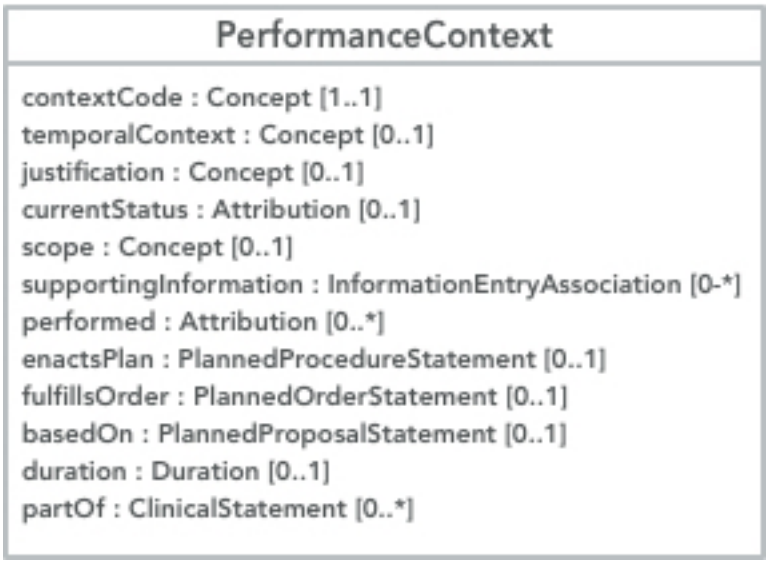


The StatementContext abstract class has the following three specializations:

FindingContext The FindingContext class aligns with the SNOMED Situation with Explicit Context for findings and provides the context for either the EvaluationResultTopic or AssertionTopic of a clinical statement. For instance, a context about a finding may state that the finding was present or absent.

ActionContext The ActionContext class aligns with the SNOMED Situation with Explicit Context for procedures and provides the context for the Act topic of a clinical statement. For instance, a statement about a procedure may specify the procedure has been proposed, ordered, planned, performed, or not performed. Each action context, in turn, has its own lifecycle. An example of the PerformanceContext class is shown in Figure 8.12, "PerformanceContext".

Figure 8.12. PerformanceContext



EventContext Specializations for EventContext have not been defined.

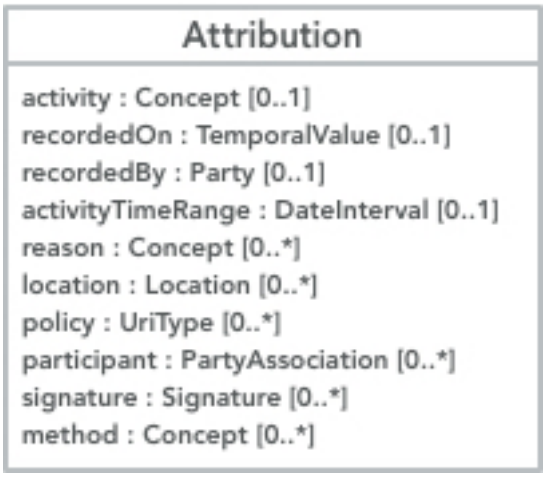
8.5. Metadata

The final division of the Clinical Statement pattern is the metadata which is a collection of attribution/provenance information regarding the topic/context being described by the clinical statement.

8.5.1. The CIMI Attribution/Provenance patterns

In the CIMI model, provenance information is represented by the Attribution class shown in Figure 8.13, "Attribution Class". The Attribution class provides a pattern for the capture of provenance information such as the what, who, when, where, why, and how associated with a particular activity – e.g., provenance attributes about the verification of a clinical statement (e.g. the provider performing the surgery in O.R. suite 6).

Figure 8.13. Attribution Class



CIMI currently includes two attribution patterns:

1. Attribution information as a part of the clinical statement – In this pattern, the ClinicalStatement pattern contains a number of attributes of type Attribution (e.g., ClinicalStatement.authored and ClinicalStatement.verified). This pattern provides a consistent way to capture attribution information that extends beyond simply the agent of an activity (e.g., the author). When attribution is part of the ClinicalStatement model, any change to the attribution for an activity will result in a version change.

2. Attribution information external to the clinical statement - CIMI allows the capture of provenance information external to the clinical statement through the Provenance class. The provenance class contains the Attribution class and provides pointers to one or more clinical statements (e.g., the Provenance.target attribute). This pattern allows the addition and modification of provenance information associated with a clinical statement without impacting its version.

8.6. Differences between ANF and CIF

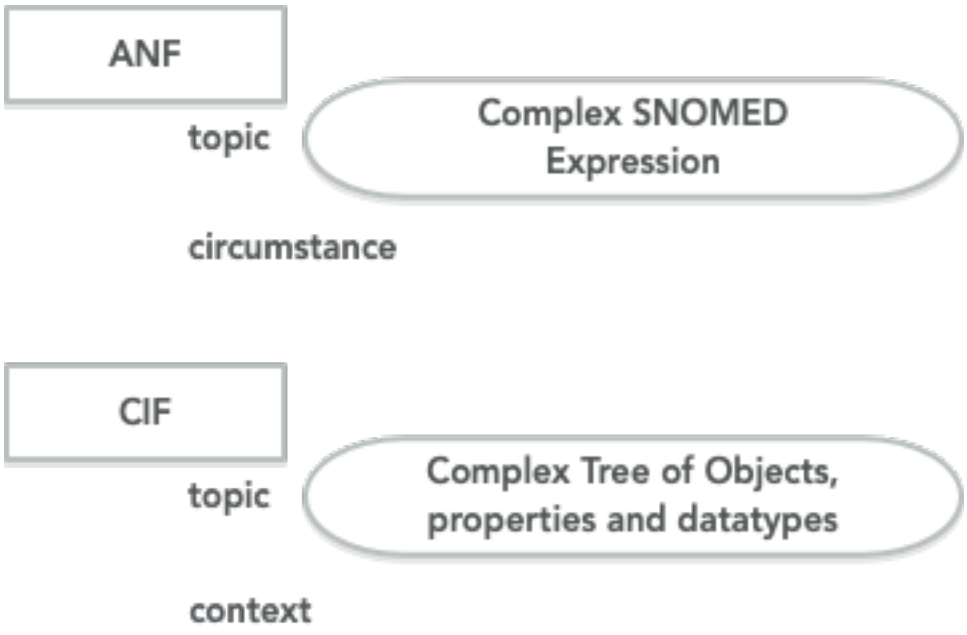
There are two fundamental differences between ANF and CIF. The first is the representation of topic, and the second is the representation of results.

1. The representation of topic.
2. The representation of results.

8.6.1. The Representation of Topic

In the ANF model, the topic is represented by a single field containing a simple to complex SNOMED expression whereas in the CIF model, all the pieces of information that make up the topic are broken out and structured as needed into multiple properties with property names and appropriate datatypes.

Figure 8.14. Topic Comparison



One implication of this is that the ANF is using two formalisms to represent the detailed clinical model. First it uses the formalism that represents the ANF reference model and constraints such as HL7's StructureDefinition syntax or OpenEHR's BMM/ADL syntax. Second, it uses SNOMED's syntax for post-coordinated SNOMED expressions. Tools for authoring and analysis would be required to parse and process both syntaxes.

The CIF model, on the otherhand, would be fully represented using the formalism that represents the CIF reference model and contraints such as HL7's StructureDefinition syntax or OpenEHR's BMM/ADL syntax.

8.6.2. The Representation of Results

In the CIMI CIF model, EvaluationResult and Assertion models are used to represent results. EvaluationResult has a topic representing what is being observed, and a result represented by a choice of datatypes. An Assertion on the otherhand, has simply a topic with a value of 'assertion', and a result stated what is being asserted.

In the ANF model, the topic represents what is being observed and the result may only be a range of either a count or quantity. No coded results are allowed.

In the CIF model, when creating a model with a numeric result, the choice is quite clear and the choice will be an EvaluationResult, such as a topic of 'SerumSodium' and result with a numeric quantity. In this case, the CIF and ANF models are very aligned, except for the fact that the ANF model will use a range representing the quantity.

But when a CIF model has a potential coded result, the choice between EvaluationResult and Assertion becomes muddied. For example, a model for Breath Sound could be an EvaluationResult with a topic of 'breath sound' and a coded result with the valueset shown below. Thus any of the breath sounds within the valueset can act as a result for this model. The other option, is that each of the breath sounds in the valueset is modeled as an Assertion with a topic of 'assertion', and a result of each particular code. To decide which model is better, usually we ponder how the clinician thinks about the data, or how it will be collected, or how it will be queried.

The ANF model cannot do an EvaluationResult style model as it doesn't allow coded results. Thus ANF is forced to make one and only choice, which is an assertion style where the particular breath sound is the topic, and the result will be a numeric count indicating presence or absence.

Example of Breath Sounds Valueset

- Absent
- Audible
- Clear
- Coarse Breath Sounds
- Coarse Crackles
- Crackles
- Diminished
- Expiratory wheezing
- Faint
- Fine Crackles
- Forced
- Inspiratory wheezing
- Left Ventricular Assist Device Noise
- Markedly Decreased
- Moderately Decreased
- Pleural Rub
- Prolonged Expiration
- Rhonchi
- Slightly Decreased
- Stridor
- Tubular Breath Sounds
- Upper Airway Congestion
- Wheeze

When querying instance data, the Assertion or ANF style is much more difficult for things like breath sounds. To query any breath sound instances, you must have knowledge of all the possible breath sound topics and query for each. With the EvaluationResult style, querying is simpler as you simply query for a topic of 'breath sound', and the coded result tells you what type of breath sound it is. Thus you do not have to know all the members of the valueset apriori to form the query.

8.7. Appendix A - Glossary

Table 8.1. Glossary

Term	Acronym	Definition
Archetype		A re-usable, formal model of a concept expressed as a computable constraint model defined in ADL
Archetype Definition Language	ADL	ADL is a formal language for expressing archetypes. It provides a formal, textual syntax for describing constraints on any domain entity whose data is described by an information model
Attribute		A field in any class
Clinical Information Modelling Initiative	CIMI	An initiative established to improve the interoperability of healthcare information systems through shared implementable clinical information models

Term	Acronym	Definition
Clinical Statement		Structured electronic communication made about a patient typically documented as an 'entry' in the patient record
Complex Clinical Statement		A statement that is composed of parts where each part can only be fully understood in the context of its parent
Compound Clinical Statement		A clinical statement composed of one or more clinical statements that may exist outside of the containing parent statement
Constraint Model		A formal specification used for describing constraints on an Underlying Reference Model. The Constraint Model is used to express clinical information models (i.e. archetypes)
Context		The circumstances that form the setting in which the ‘topic’ should be evaluated
Detailed Clinical Model	DCM	A relatively small, standalone information model designed to express a precise clinical concept in a standardized and reusable manner
Governance		The use of a set of processes, customs, policies, laws and institutions to direct the way people administer
Isosemantic Models		A model that, while different in structure, represents the same semantic content as a second model
Key		The main concept of interest in a clinical statement, about which the other attributes and relationships provide additional information
Metadata		Attribution information relating to the statement itself such as who authored, verified, recorded, or signed the statement. Metadata includes the who, where, why and when information
Terminology Binding		The assertion of a relationship between the information model and the terminology
Topic		The clinical entity described by the Clinical Statement, e.g. clinical assertions, evaluations results, and procedures
Topic Pattern		Attributes required to fully describe a clinical entity

9. KNART statement supports

KNARTs support the creation of statements through standardized questionnaires and order sets.

Part V. Assertional representation

Table of Contents

10. Assertions

10.1. Introduction

There are two types of patterns, which can be used to represent clinical observations. The assertional pattern and the evaluation result pattern. The general difference between those patterns is that the evaluation result pattern is best used, where a topic is evaluated with multiple different evaluation results. These evaluation results would be represented in a value set. The application of an assertional pattern is more useful, where assertions about the patient, e.g. an observation of a diagnosis, are made and the result of the observation is either “present” or “absent”. In those cases, the topic of the observation is represented as pre-coordinated concepts rather than value sets.

10.2. Assertional Pattern

10.3. Evaluational Result Pattern

10.4. Examples

Examples of Assertional Patterns:

• “Pneumonia” (present/absent)

• “Myocardial infarction” (present/absent)

Examples of Evaluation Results Patterns:

• “Ethnicity” – with a value set of all ethnicities

• “Pain” – with a value set of pain qualities, e.g. “aching”, “burning”, “stabbing” etc.

11. Solor Assertional Knowledge

11.1. Solor Representation

The Solor representation of Assertional Knowledge reaches beyond the patient as the subject of record and observations and evaluation results about the patient. It represents knowledge that can be applied to, e.g. the patients treatment or diagnostics.

11.2. Solor

Examples of Assertional Knowledge:

• “Aspirin treats pain”

• “Penicillin treats bacterial infections”

• “Myocardial infarction is associated with chest pain”

The Solor capability of associating statements enables the use of Assertional Knowledge to Clinical decision support applications, clinical pathways and general information (“info button”) that can be made available to users of EMR systems.

Part VI. Procedural representation

Table of Contents

12. Procedural Representation

12.1. Introduction

The Clinical Statement Type distinguishes between the performance of an action (“performed”) and a request for an action (“requested”). Both can be statements of procedures or interventions, which are requested (ordered) for a patient or have been performed on a patient in the past.

12.2. Performance of Action

12.3. Request for an Action

12.4. Statements of Procedures

12.5. Examples

Examples:

• Request for administration of medication

• Request for diagnostic imaging

• History of radiation therapy

12.6. Solor ANF Representation

In the context of the Solor ANF Model, the representation of procedures is comprised of the requested or performed procedure as the topic and a number of allowed circumstances, e.g. priority, frequency, duration and repetitions. Topic and circumstances are all codable concepts. Details about the ANF Model representation of procedures are discussed in Chapter xx

13. Procedural Knowledge Representation

13.1. Procedural Representation

As with Assertional Knowledge, the representation of Procedural Knowledge reaches beyond the documentation of procedures requested for or performed on the subject of record. Procedural Knowledge can pertain to, e.g.

• Standard ways of performing a procedure

• Treatment protocols for diseases

• Standard evidence-based Order Sets

Applied Procedural Knowledge can enable the use of Clinical Pathways, Clinical Decision Support and Knowledge Artifacts (KNARTs) that standardize patient documentation focused on clinical domains and patient situations.

13.2. Examples

Part VII. Solor Tooling

Table of Contents

14. Tooling for Solor

14.1. Introduction to KOMET

KOMET (Knowledge Management Tool) is the forthcoming tool suite that will be released for Solor. It will be published along with a user guide and appropriate downloads for installation at http://www.solor.io. [1]

14.1.1. KOMET

The VA's Foundational Informatics Architecture – which we call ISAAC – is an integrative logical architecture, which deliberately builds each new layer upon selected, compatible elements of its underlying components to build a coherent system. The Foundational Informatics Architecture builds primarily upon SNOMED CT, RxNorm, and LOINC by integrating their content and semantics, and normalizing the means to identify and version components, lexically search, logically define, semantically retrieve, and collaboratively extend. Support for evolutionary change is a critical feature of the Foundational Informatics Architecture (DERIVIATE), given that support for changes in knowledge over time is a critical aspect of health informatics.

The primary goal of the Foundational Informatics Architecture (DERIVIATE) is semantic operability (vs. interoperability). Semantic operability is the meaningful (semantic) use of data within the various components and uses of a single health IT system (vertical integration). Semantic operability is achieved by using a coherent integration of SNOMED CT, RxNorm, and LOINC as the primary building blocks upon which the foundational architecture is based. DERIVIATES's strict separation of concerns enables terminology components, as well as higher order derived structures such as clinical rules, to undergo evolutionary change without requiring changes to the architecture that it resides within. This allows for an agile environment with a stable architecture.

The lowest 'pad stone' layer of the DERIVIATE architecture is the integrated suite of standard terminologies including SNOMED CT, RxNorm and LOINC. Two higher layers build upon this terminology 'pad stone'. A Clinical Data layer uses standardized terminology to describe facts about a patient e.g., "John Doe has PNEUMONIA." A Procedural Knowledge layer uses standardized terminology to express biomedical and organizational knowledge, independent of any specific patient. For example, "Hydrochlorothiazide treats Hypertension" or "Myocardial Infarction elevates Troponin T Levels." CDS rules, order sets and documentation templates are also expressed in the "Procedural Knowledge" layer.

The VA must have tools to help knowledge workers create and maintain standards-based clinical decision support artifacts at enterprise scale. Tools must be able to produce CDS content that is standards compliant when such standards exist (e.g. HL7 CDS Artifact Specification DSTU 1.3). Tools must also build CDS artifacts that contribute to an ecosystem of semantic operability. This necessarily means building artifacts using standards-based "pad stone" building blocks of SNOMED CT, RxNorm and LOINC. CDS knowledge engineering content development tools must create artifacts in the layered approach described in DERIVIATE (i.e., tools must build more complex, standards-based artifacts by reusing less complex standards-based artifacts as components whenever possible). Tools supporting a highly collaborative, integrated and layered knowledge management environment must be carefully designed to be highly reusable within a common framework and use experience.

In the following sections we will first describe the common features required of all components in the integrated standards-based knowledge management tool suite and then document tool-specific requirements necessary for each artifact type.

[1] http://www.solor.io

14.1.1.1. Heuristic principles

To develop an optimal user interface (UI) for all of the editors that will be used to create and update clinical decision support artifacts, the Contractor shall execute an agile, User-Centered Design (UCD) and implementation process that addresses the stages shown in the figure below. Note that each stage is meant to be iterative and repeated as often as needed for each editor before moving to the next editor.

Part of user-centered design includes measuring usability.

Measuring usability starts with the five attributes of usability commonly referenced in the literature, shown below.

- Easy to learn (and re-learn)
- Efficient to use (performance)
- Effective to use (completion)
- Prevents errors (not cause harm)
- Satisfying to use (subjective impression)

To accomplish this, the usability testing strategy consists of two main components:

- 1) Formative Testing: Evaluating the usability of early designs of the user interface for the CDS editors and Governance Tool prior to and/or during software development.
- 2) Summative Testing: Measuring and testing system usability of the coded software that is stable and releasable in a test account with valid test data.

CDS and other knowledge artifacts share many similarities. Certain similarities are manifest in their basic components and structure. All CDS and knowledge artifacts should refer to standard terminologies (i.e., the lower layers of DERIVIATE) for the clinical entities and clinical actions comprising the higher order artifact. For example, a CDS rule that evaluates if a specific medication is being taken by a patient as a condition for execution should refer to RxNorm medications. DERIVIATE layers more complex artifacts on top of the terminology pad stone and on top of each other. In general, more complex layers each specify an artifact-specific syntax to orchestrate terminological and other less complex components into the desired higher level artifact. This means that more complex CDS artifacts may be composed of CDS artifacts of lesser complexity.

CDS artifacts of different levels of complexity may be developed by the same knowledge engineers. This has important implications for design and functionality of the user interface. The user interface must be consistent and provide an integrated view of knowledge artifacts at all levels of complexity. CDS artifacts must be searchable in clinically relevant ways, regardless of their final composition. An obvious example is finding artifacts containing identical or similar terminological concepts. Another is that basic editing functions must be consistent, easily learned and similar to typical editing conventions (e.g., copy, paste).

Knowledge engineers must collaborate to develop CDS artifacts in several ways. First, knowledge engineers may request review and critique of their work products by others, both informally during the build process and formally prior to release. Some CDS artifacts will be complex due their sheer size or because they are composed of collections of sub-artifacts. For example, a complex artifact for ordering clinical subspecialty consults may contain sub-artifacts of ECA rules, order sets and documentation templates. Knowledge engineers may take responsibility for portions of a complex knowledge artifact, divided by section or by sub-artifact type. Knowledge engineers must be able to request the development of sub-artifacts by other knowledge engineers and to track fulfillment. Tools must be able to support collaboration amongst knowledge engineers.

CDS artifacts share common metadata because they are formally dependent on each other (i.e., expression of asset to asset linkages). Artifacts must share common metadata regarding time stamps, editing, versioning and tracking. Other types of metadata are common because of overlapping requirements for linking the assets to the deploying organization, to the literature and to clinical work processes.

The fact that there are numerous similarities among CDS and other knowledge artifacts has important implications for the knowledge engineering tools used to create them. In short, CDS knowledge engineering tools share many common features and capabilities that will be described in the sections below. We acknowledge the work done by Zhou and colleagues regarding rule authoring environments requirements and reuse certain of their best practice requirements in this document.

14.1.1.2. Look and feel

14.1.1.3. Document template editor

The HL7 Version 3 Standard: CDS Knowledge Artifact Specification, DSTU (Draft Standard for Trial Use) Release 1.3, page 38, presents the following definition of documentation template from the HeD (Health eDecisions) Artifact Sharing Use Case:

"… a documentation template is a structured form for recording information on a patient into a set of predefined data slots. These templates are used to guide structured data entry within an EHR or other clinical information system."

The types of clinical documents that can be represented using documentation template artifacts include, but are not limited to, patient visit (encounter) summaries, procedure notes, consultation reports, patient-reported outcomes, and flowsheets.

A Documentation Template editor should be able to create documentation template artifacts representing a variety of clinical document types for use at both the VA local (facility) and enterprise level.

The HL7 Version 3 Standard: CDS Knowledge Artifact Specification, DSTU Release 1.3 includes Documentation Templates as a primary artifact type and Figure 4 on page 39 of the specification, provides a conceptual overview diagram of required and optional components.

The purpose of the Documentation Template editor tool is to support the creation of standardized Documentation Template knowledge artifacts. The Government requires a model-based Documentation Template editor that will allow the user to create documentation template knowledge artifacts that are based on SME-defined content that can be implemented within the VA's electronic health record (EHR) at the point of care and within the clinical workflow, and that conforms to the specifications in the HL7 Version 3 Standard: CDS Knowledge Artifact Specification, DSTU Release 1.3 (or later HL7 version or final standard if one is released).

The VA requires a Documentation Template Editor that can be used by the VA and non-VA end users to generate Documentation Templates of various types as may be needed in the full spectrum of medical practice. The Documentation Template Editor shall generate documentation templates that ultimately will be used by VA clinicians to manage patient care in a production environment.

The Documentation Template Editor shall be able to be used to create any type of documentation template as a structured collection of documentation concepts (also referred to as "form elements" or "observation items"). Per the HL7 Version 3 Standard: CDS Knowledge Artifact Specification, DSTU Release 1.3, "Each documentation concept … also can be thought of as a question to the user entering the data". Elements within the documentation concept serve a purpose to guide and constrain the user's responses -- for example, a list from which to choose an answer; whether an answer is a number, a date, or some other type; and the cardinality of the answer. (HL7 Version 3 Standard: CDS Knowledge Artifact Specification, DSTU Release 1.3, pages 38 and 39.)

Documentation concepts are contained in an action of type CollectInformationAction, enabling these concepts to be presented to the user conditionally (e.g., to ask questions appropriate to a patient's gender or to ask questions based on other responses), to compute responses for a concept based on previous responses or data from an EHR score (e.g., a risk score), and to bind the responses into expressions that can drive logic elsewhere in the documentation template (e.g., ask questions conditionally as described above). Thus, resulting documentation templates are capable of branching logic, and the forms created must be able to specify all the actions (such as action of type CollectInformationAction) within the HL7 Version 3 Standard: CDS Knowledge Artifact Specification, DSTU Release 1.3.

The documentation concepts in a template typically are organized hierarchically, into sections and subsections with the concepts themselves at the very bottom of the structure. In HeD Knowledge Artifact schema, these "sections" are called actionGroups - which in documentation templates may have behavior indicators associated with each actionGroup, e.g., whether a documentation concept must have a response.

14.1.1.4. Event condition action rule editor

The HL7 Version 3 Standard: CDS Knowledge Artifact Specification, Release 1.3, page 31, presents this definition of an event condition action (ECA) rule from the HeD (Health eDecisions) Artifact Sharing Use Case:

… an event condition action rule is an artifact with the general syntax "on event, if condition is true, then do action." The event triggers the invocation of the rule. The condition is a logical test that, if satisfied or evaluates "true," causes an action. The action part consists of a set of operations to execute. These actions may in turn cause further events to occur, which may in turn cause other ECA rules to fire…The action groups are the containers and organizers of the actions in an ECA rule. A rule typically has a single action group (top level section), but may have more. Conceptually, a set of actions in a rule could be considered a "mini order set" which is presented to a clinician at certain times and under certain conditions. As such, the actions may be structured hierarchically using action groups and behaviors to specify how the orders should be a shown to a provider, and to place restrictions on how a provider chooses from the available set of orders. It should be noted that this is just a conceptual example, and that not all actions are necessarily orders. For example, an action can be a creation of a new event that triggers another rule, a future encounter, or the creation of a state description of the patient.

"Efficient rule authoring tools are critical to allow clinical Knowledge Engineers (KEs), Software Engineers (SEs), and Subject Matter Experts (SMEs) to convert medical knowledge into machine executable clinical decision support rules."

An ECA Rules editor should be able to facilitate the user's ability to generate both local (i.e., for a single facility), VISN (i.e, for a group of facilities), and enterprise-level ECA rules that are standardized, sharable, interoperable, and extensible.

The VA requires a model-based ECA Rules editor that will allow the user to generate ECA Rule knowledge artifacts based on SME-defined content that can be implemented within VA's electronic health record (EHR) at the point of care and within the clinical workflow. The ECA Rules editor shall generate CDS knowledge artifacts with the general syntax "on event, if condition is true, then do action."

The VA requires an ECA Rules editor that will be used by government and non-government end users to generate ECA rules applicable to the full spectrum of medical practice, including generation of ECA rules that support the application of clinical practice guidelines and protocols in patient care as well as the dynamic management of these guidelines and protocols. The ECA Rules editor shall generate ECA rules that ultimately will be used by VA clinicians to manage patient care in a production environment.

The CDS knowledge artifacts generated by the ECA Rules editor shall conform to specifications defined in the HL7 Version 3 Standard: CDS Knowledge Artifact Specification, Release 1.3 (or later version or final standard if one is released).

14.1.1.5. Condition editor

14.1.1.6. Expression editor

14.1.1.7. Action editor

14.1.1.8. Order set editor

Clinical orders are used to initiate the majority of healthcare delivery activities in the US and thus are a major driver of cost, quality and safety. Orders are used in virtually all healthcare settings including (but clearly not limited to) medication prescribing, laboratory tests, imaging, procedures, consultations, encounters and hospital admissions. In the VA, for example, well over 1.2 million orders are entered every day and VistA contains billions of orders in aggregate.

Clinical orders' ubiquity and impact on healthcare delivery has made ordering a central focus of quality improvement efforts. Health Information Technology (HIT) was used to improve the ordering process when paper was the only available medium. Computerized provider order entry has taken clinical quality, safety and efficiency improvement initiatives to another level. Order related interventions are manifold and include allergy and interaction checking among medications and foods, appropriateness checks amongst all combinations of disease, drugs and labs; the establishment and enforcement of ordering prerequisites; and limitation of authority to place orders.

Order sets are an important category of order related interventions that enjoy widespread use because they have been shown to improve quality while enhancing the efficiency of the ordering provider (a rarity).

The HL7 Version 3 Standard: CDS Knowledge Artifact Specification, DSTU Release 1.3, includes order sets as a primary artifact and on page 34, presents this definition of an order set from the HeD (Health eDecisions) Artifact Sharing Use Case:

…an order set is a pre-defined and approved group of orders related to a particular clinical condition (e.g., hypertension treatment and monitoring) or stage of care (e.g., hospital admission to Coronary Care Unit). An order set is used as a checklist for the clinician when managing a patient with a specific condition. It is a structured collection of orders (or actions in the HeD schema) relevant to that condition and presented to the clinician in a computerized provider order entry system (CPOE).

Ordering providers use order sets as check lists, menus, and order construction shortcuts. Order sets are often embellished with clinical rationale and guidance about their proper use and literature references for the ordering provider.

The HL7 CDS specification provides a conceptual overview diagram of required and optional components.

An order/order set editor is needed that will be used to create knowledge artifacts for individual orders and order sets applicable to the full spectrum of medical practice and which conform to specifications defined in the HL7 version 3 Standard: CDS Knowledge Artifact Specification, Release 1.3 (or later version or final standard if one is released).

14.1.1.9. Aggregate artifact editor

The VA intends to use HL7 KNART artifacts for a variety of purposes in addition to documentation. In particular, we will include document templates as a core component for the ordering of specialty consults in combination with orders and order sets. We refer the class of artifacts that are composed of multiple KNART artifacts as "composite artifacts". We anticipate that there will be need for multiple types of composite artifacts in addition to specialty consultations.

The majority of the effort of creating specialty consults and other composite artifacts above and beyond construction of the subcomponents will be devoted to subcomponent integration into the desired consult. Other requirements can be met using editing environments for the individual subcomponents.

14.1.1.10. Presentation layer editor

The HL7 CDS Artifact specification is designed as an interchange format for CDS artifacts. This approach promotes the exchange of clinical decision support content because poorly shareable platform-specific implementation details are not included in the exchanged artifacts. While platform-specific implementation details are absolutely needed in order to execute the artifact in a given live HIT system, they might impede efforts to implement the clinical components of CDS artifacts in some other environment. The separation of CDS artifact interchange format from implementation format is an important step towards creating an ecosystem of shareable standards-based CDS on shareable standards-based data. As a result of these beneficial tradeoffs, HL7 CDS Artifacts must be transformed from an interchange format to an implementation format in order to be executed.

The purpose of presentation layer tools is to support the conversion of standardized interchange artifacts into implementable CDS artifacts. The scope included in this section includes any type of tool needed for CDS exchange artifact conversion. The initial tool to be constructed will support the conversion of HL7 CDS Documentation Templates with CQL into HTML5 templates with Drools DRL.

14.1.1.11. Governance workflow management

Achieving standards-based shared clinical decision support at the enterprise scale is a complex undertaking with many technical and organizational steps that require careful orchestration. Knowledge management tools supporting organizational processes are as important as technical tools for achieving wide-spread support, implementation and adoption of knowledge products such as clinical decision support rules, order sets and documentation templates. Numerous organizational challenges must be met at different phases of the CDS lifecycle, including problem identification, solution analysis, knowledge development, organizational vetting, impact assessment and periodic assessment (fig x in introduction).

CDS enterprise governance tools are designed to support organizational vetting and periodic review of enterprise knowledge artifacts. The desired end results are high quality knowledge artifacts that have been reviewed and approved for implementation by appropriate and authoritative bodies. Organizational vetting and periodic review involves various subject matter experts and governing bodies to perform the following functions:

- Critically assess and evaluate the proposed CDS artifact
- Document potential issues
- Decide to pursue or ignore identified issues
- Develop potential resolutions to those issues
- Approve of one or more resolutions and re-evaluate the proposed remediated artifact.

To complicate matters, different groups may be involved in vetting and periodic review of the same artifacts. Methods to integrate and harmonize or version and track different CDS artifacts are an essential feature of governance tools. Business requirements supporting these essential steps are described below.

14.1.1.12. Metrics and refactoring support

We need to think through the types of metrics and refactoring. Look at some of Fowler's books on these topics, and then come up with analogies for our domain.

14.1.1.12.1. Linguistic knowledge refactoring

Linguistic knowledge codifies the relationship between our words, and the shared concepts we hope they adequately represent. In ISAAC, we depend on several aspects of linguistics to make abstract concepts, initially defined with only a set of identifiers, sufficiently concrete to prove a shared understanding of the thoughts those concepts represent. For ISAAC, relevant aspects of linguistics include morphology (the structure of words), syntax (the structure of sentences), semantics (meaning), pragmatics (language in context), language variation (i.e., dialects), and language change over time.

14.1.1.12.2. Definitional knowledge refactoring

T-Box semantics

14.1.1.12.3. Declarative knowledge refactoring

A-Box semantics

14.1.1.12.4. Imperative knowledge refactoring

Bibliography

Motivation and Foundation

[interoperabilityprogress2018] Omar Bouhaddou, Sandra Mitchell, Chun Li, Russell Leftwich, Todd Turner, Matthew Rahn, Margaret Donahue, and Jonathan Nebeker. Copyright © null. . *Interoperability Progress and Remaining Data Quality Barriers of Certified Health Information Technologies*.

[allergen] Foster R. Goss, Li Zhou, Joseph M. Plasek, Carol Broverman, George Robinson, Blackford Middleton, and Roberto A. Rocha. Copyright © 2013. 27486010. 0168-8278. 10.1016/j.jhep.2016.07.033. 16000641. Journal of the American Medical Informatics Association. *Evaluating standard terminologies for encoding allergy information*.