Implementing reusable software components for SNOMED CT diagram and expression concept representations

Balázs BÁNFAI a,1, Roland PORCIÓ a, Tibor KOVÁCS b
a B2i Healthcare, Budapest, Hungary
b Budapest University of Technology and Economics, Hungary

Abstract. SNOMED CT is a vital component in the future of semantic interoperability in healthcare as it provides the meaning to EHRs via its semantically rich, controlled terminology. Communicating the concepts of this terminology to both humans and machines is crucial therefore formal guidelines for diagram and expression representations have been developed by the curators of SNOMED CT. This paper presents a novel, model-based approach to implementing these guidelines that allows simultaneous editing of a concept via both diagram and expression editors. The implemented extensible software component can be embedded both desktop and web applications.

Keywords. SNOMED CT, diagram representation, Compositional Grammar, software modelling, reusable implementation, Snow Owl

Introduction

SNOMED Clinical Terms (SNOMED CT) [1] is considered the most comprehensive, multilingual clinical healthcare terminology in the world. It is maintained by the International Health Terminology Standards Development Organization (IHTSDO) [1] with the primary purpose to ensure interoperability and to provide the semantics to an interoperable electronic health record. SNOMED CT can be characterized as a multilingual thesaurus with an ontological foundation structured as a multi-axial subtype hierarchy. The ontology is composed of rational statements in the form of Concept1 - Relation - Concept2 where multiple relation types can connect the concepts. Concept relationships can express hierarchical or other types of relationships enabling not only the hierarchical grouping of the related concepts but also logical reasoning via programmatic classification.

As communicating the concepts of SNOMED CT is a very important aspect of the development of the terminology, IHTSDO has developed specifications for representing concepts both as diagrams [2] as well as textual expressions [3] defined by a formal compositional grammar. Both representations are meant to formalize communication while eliminating ambiguity; the diagrammatic representation is targeted for humans whereas the expression is designed primarily to be machine-
In addition to conformance to the specifications, the authors aimed to satisfy the following requirements:

- Concepts simultaneously editable via the expression and diagram editors while keeping both representations in sync
- Editing possible both locally and remotely over the internet
- Support for concurrent modification in case of a distributed deployment
- Embeddable into both desktop and web applications
- Extensible to accommodate future extensions of the specifications or additional user requirements

1. Methods

Each SNOMED CT concept encapsulates a unique clinical meaning, which is identified by a unique and immutable numeric identifier. Concept identifiers do not carry implicit meaning; the meaning of the concept is defined by its relationships to other concepts within the terminology. Each concept can have multiple human-readable descriptions including the unique and mandatory Fully Specified Name. For example:

**CONCEPT**

- Concept ID – 22298006
- Fully Specified Name - Myocardial infarction (disorder)

**DESCRIPTIONS**

- Preferred Term - Myocardial infarction
- Synonyms - Myocardial infarct, MI - Myocardial infarction, Infarction of heart, Cardiac infarction, Heart attack

**RELATIONSHIPS**

- Myocardial infarction **IS A**
  - Myocardial disease (disorder)
  - Necrosis of anatomical site (disorder)
  - Ischemic heart disease (disorder)

The defining **ATTRIBUTES** are

- Associated morphology - Infarct (morphologic abnormality)
- Finding site - Myocardium structure (body structure)

1.1. SNOMED CT Compositional Grammar and Diagramming

SNOMED CT concepts can be described as a SNOMED CT Compositional Grammar expression. The syntax of SNOMED CT Compositional Grammar [3] is specified in Augmented Backus-Naur Form (ABNF). A concept definition minimally consists of the concept identifier, optionally followed by a human-readable description of the concept. Multiple parentage is expressed by concatenating the parents using the plus sign and refinements to the concepts can be appended after a colon following the concept:

```
57809008|Myocardial disease| + 414545008|Ischemic heart disease| +
609410002|Necrosis of anatomical site|:
116676008|Associated morphology| = 55641003|Infarct|,
363698007|Finding site| = 74281007|Myocardium structure|
```
The diagramming guidelines define the graphical notation for representing a concept definition where the concept being defined is shown at the top left connected to a series of expressions via operators. It is allowed to omit the defined concept. Hollow arrows represent parentage by pointing to the parent concept, whereas other types of relationships are represented by a rounded rectangle between two regular arrows pointing from source to target.

1.2. Solution Approach

The authors have selected the Eclipse Modeling Project as the technological foundation for implementing the Model-View-Controller (MVC) [4] pattern required by the editors. At the core of the Eclipse Modeling Project lies the Eclipse Modeling Framework (EMF) [5], a modeling framework and code generation facility built around its Ecore meta-model. Ecore is an implementation of the Object Management Group (OMG) Essential Meta-Object Facility (EMOF) standard [6]. Ecore models are extensible by definition and can either be built from scratch or generated from existing XML schemas, UML models, Xtext domain specific languages (DSLs) or annotated Java interfaces. Models described by Ecore have built-in support for reflective editing, change notification, validation and serialization. The authors have defined a common model as an Xtext DSL [7] by adopting the SNOMED CT Compositional Grammar syntax. Xtext then generates an EMF model for the DSL, which serves as the common target for both the diagram and expression editors.

User modification of the diagram or textual expressions generates events that are caught by listeners. These listeners notify the editor to perform the necessary modifications on the model. Grammar changes are converted using an Xtext DSL Parser that is also responsible for validating the input. As long as the expression is invalid, the graphical editing is disabled. Notifications about model changes have been developed via EMF Adapters [5] that implement the Observer design pattern [4]. For creating the textual representation of the model, an Xtext Serializer is configured which is capable of converting the given model and formatting the output based on the grammar rules. Figure 1 shows the implemented architecture and the communication between the main components of the library.

![Figure 1. Editing and notification.](image)
1.3. Desktop and Web Clients

For desktop editing the authors have selected the Eclipse Graphical Editing Framework (GEF) [7] as it tightly integrates with EMF. The actual diagram elements have been developed as custom figures. For the expression entry the StyledText component was used. For the web editing the Vaadin framework [8] was selected. The framework is suitable for creating efficient and desktop-like applications and it supports the creation of customized Google Web Toolkit (GWT) [9] components. The textual editor is a customized RichTextArea widget, while the graphical editor implements the Canvas component. The client side (web browser) is responsible for the representation of the data and user event creation, while the server side handles the events from the client, updates the model based on the events, informs the editors if the model was changed and performs the model conversion and validation.

2. Results

The implemented editors do not make assumptions whether the model has been accessed locally or remotely which makes it suitable for a standalone environment as well as for client-server architectures. Via the notification mechanism, both editors are kept in sync. Validation of the user modifications is carried out automatically and the errors are shown both in the expression and in a pop-up window. The invalid part of expression is highlighted and underlined with a red squiggle and error messages are shown in the lower right corner in a pop-up window.

A demo web application is shown in Figure 2. In addition to the web application, the created desktop component was also embedded into Snow Owl [10] (Figure 3), a commercial medical ontology authoring solution. The selected concept in the left side concept navigator is visualized. The expressions or the diagrams can be exported or printed directly from both the web and desktop components.

![Figure 2. Component in a web application.](image-url)
3. Discussion

Java components have been implemented to simultaneously view and edit SNOMED CT concepts both as a diagram and as an expression. The solution conforms to the relevant IHTSDO specifications. The heavily utilized Eclipse modeling software libraries proved to be an excellent choice to fulfill the requirements as the implemented common EMF model supports concurrent editing, change notification, validation and serialization. The Java library developed can be embedded both into desktop as well as into web applications. The introduced approach and the developed model can also be extended to accommodate future requirements of the evolving standards. The authors are not aware of any implementation at present that conforms to the formal specifications published by IHTSDO while fulfilling the non-functional requirements stated in this document.

References