Implementing an HL7 Version 3 Modeling Tool from an Ecore Model

Balázs BÁNFAI\textsuperscript{a}, Brandon ULRICH\textsuperscript{b,1}, Zsolt TÖRÖK\textsuperscript{a}, Ravi NATARAJAN\textsuperscript{c}, Tim IRELAND\textsuperscript{c}
\textsuperscript{a} B2 International Ltd., Budapest, Hungary
\textsuperscript{b} B2 International LLC, Pierre, South Dakota, USA
\textsuperscript{c} NHS Connecting for Health, Leeds, UK

Abstract. One of the main challenges of achieving interoperability using the HL7 V3 healthcare standard is the lack of clear definition and supporting tools for modeling, testing, and conformance checking. Currently, the knowledge defining the modeling is scattered around in MIF schemas, tools and specifications or simply with the domain experts. Modeling core HL7 concepts, constraints, and semantic relationships in Ecore/EMF encapsulates the domain-specific knowledge in a transparent way while unifying Java, XML, and UML in an abstract, high-level representation. Moreover, persisting and versioning the core HL7 concepts as a single Ecore context allows modelers and implementers to create, edit and validate message models against a single modeling context. The solution discussed in this paper is implemented in the new HL7 Static Model Designer as an extensible toolset integrated as a standalone Eclipse RCP application.

Keywords. HL7, Ecore, EMF, XMI, EMOF, standards, semantic interoperability

1. Introduction

Health Level Seven\textsuperscript{2} version 3 (HL7 V3) defines standards for messages that are exchanged within a typical healthcare workflow. HL7 V3\textsuperscript{3} relies upon object-oriented principles whereby all message models are derived from the Reference Information Model (RIM)\textsuperscript{4}, which expresses the data and semantic associations for archetypical classes. In addition to other parameters, the data defined in the message models are specified predominantly by their data type and vocabularies. These HL7 standard specifications (RIM, data types, and vocabulary definition) are currently serialized in XML syntax defined by the Model Interchange Format (MIF).

One of the main challenges to achieving interoperability using the HL7 V3 healthcare standard is the lack of clear definition and supporting tools for modeling, testing, and conformance checking. Currently, the knowledge defining modeling is scattered around in MIF schemas, tools and specifications or simply with the domain experts. To make matters even more complex, as the HL7 standard evolved numerous flavors and versions have emerged. Conformance can only be discussed in terms of a

\textsuperscript{1} Corresponding Author: Brandon Ulrich, B2 International LLC, 608 E Missouri Ave, Pierre, SD 57501, USA; E-mail: bulrich@b2international.com.
\textsuperscript{2} http://www.hl7.org.
\textsuperscript{3} http://www.hl7.org/library.
\textsuperscript{4} ISO HL7 21731 “Health informatics – HL7 version 3 – Reference Information Model”.
domain context that comprehensively encapsulates these HL7 concepts as well as a set of conformance rules to which the derived messages adhere.

One of the largest users of HL7 V3 messaging is the English National Health Service (NHS) Connecting for Health (CFH) program. As of March 30, 2009, over 180 million electronic prescriptions—each of which utilizes several HL7 V3 messages—have been processed [1].

The current tooling support for HL7 V3 message modeling has proved inadequate, and as the lack of tooling is a barrier to adoption [2], HL7 has consequently increased their commitment to tooling as a strategic imperative [3]. As the business case for industry tooling investment may be insufficient [4], CFH launched a six-month sponsored open-source project to deliver a next-generation toolset using open standards and built using the Eclipse framework.

2. Methods

The modeling tool was built upon the Eclipse platform—a favored framework for tool integration, particularly in the healthcare space. Standards Development Organizations (SDOs) like HL7 and the International Health Terminology Standards Development Organization (IHTSDO) have committed to tooling development using the Eclipse platform and standards adopters such as NHS CFH, Canada Health Infoway, and the US Veterans Health Administration are all building healthcare tooling on the Eclipse platform. A reason for this is the large number of open-source tools currently available built on the Eclipse framework.

The Eclipse Modeling Project provides various model-based development technologies, one of them being EMF, a modeling framework and code generation facility for building tools and other applications based on a structured data model. It is being developed and supported by a large and growing community, and a flourishing ecosystem has evolved around it. Being tightly integrated with Eclipse, EMF exploits its facilities to a great extent; it is the underlying foundation of many other modeling technologies. At the core of the EMF framework is the Ecore meta-model [5], which is similar to the Object Management Group (OMG) Essential Meta-Object Facility [6] (EMOF) standard. Ecore is the meta-model that describes models in EMF using building blocks such as EClass, EAttribute, and EReference. Ecore models can either be built from scratch or generated from existing XML schemas, UML models, and annotated Java interfaces. Models described by Ecore have built-in support for reflective editing, change notification, and serialization.

3. Results

3.1. HL7 Domain Model Expressed in Ecore

Numerous recent initiatives have shown that conceptual models can be expressed in different ways: Java, XML schema, or UML [7–9]. The common approach to these solutions is to base the conceptual model on the existing MIF schemas either by generating custom Java code from them, or transforming them to UML. EMF unifies Java, XML, and UML by providing an abstract, high-level representation. By
representing the core HL7 V3 context (RIM, data types and vocabulary) as well as the message models as Ecore models, the developers of the standard can edit these artifacts graphically or with a simple tree-based editor, serialize these artifacts in standard XMI [10] format and keep them in tight version control.

3.2. HL7 Modeling Tool Architecture

The architecture of the HL7 Static Model Designer is component-based and was designed with extensibility in mind, see Figure 1. The HL7 Static Model Designer is delivered as a standalone, self-contained Rich Client Platform (RCP) application. User documentation is integrated and provides a good starting point for new users.

3.2.1. Validation

Validation is an important feature for any modeling tool because it helps to reduce the time spent on finding and fixing human errors. The EMF Validation Framework\(^5\) makes use of the domain constraints built into the meta-model itself (e.g., cardinality), additional constraints are contributed via Eclipse extension points. Constraints are contributed to categories, so various constraint sets can be defined to accommodate the different flavors of the HL7 standard. Constraints can be defined in Java, OCL, or openArchitectureWare’s\(^6\) (oAW) Check language. Constraints can be enabled and disabled. All constraints are available for batch validation, which can be explicitly invoked by the user. The validation results are shown as markers on the diagram as well as in the Problems view, where they can be sorted and filtered. Some constraints

\(^5\) [http://www.eclipse.org/modeling/emf/?project=validation](http://www.eclipse.org/modeling/emf/?project=validation).

\(^6\) [http://openarchitectureware.org](http://openarchitectureware.org).
can be tied to a specific user action, for example adding a new class to a diagram. These constraints are checked when the associated action is performed, and any constraint violations are immediately shown to the user without explicitly invoking the validation.

3.2.2. Serialization

The default serialization format used by EMF forpersisting its Ecore models is OMG XML Metadata Interchange (XMI) 2.0. Using XMI, the EMF Persistence Framework can persist Ecore models as EMOF models as well, because the two meta-models are extremely similar. Both XMI and EMOF are widely adopted open standards, which enhances the interoperability with existing modeling tools. The serialized models can be saved, version controlled, or persisted in a database using Ten
eo.7

3.2.3. Model Transformation

Although our Ecore meta-model is MIF agnostic, transforms for importing and exporting were implemented for several versions of the MIF, including V1.0 and V2.0.x. The transform engine automatically determines the MIF schema version of the file and applies the appropriate transform script. MIF support has been implemented based on the MIF schemas, which are used as a meta-model by oAW’s XSD adapter. Support for new versions and flavors of the MIF can be added by contributing a new model-to-model (M2M) transform script via the provided Eclipse extension point.

3.3. Generating the Modeling Tool

The graphical editor itself is implemented using the Eclipse Graphical Modeling Framework8 (GMF). GMF provides both a code generation facility and a runtime platform to build graphical editors for models expressed in Ecore. Most of the editor code is generated based on the domain model, followed by manual customization via the Eclipse Graphical Editing Framework9 (GEF) to implement domain-specific behavior.

According to the standard, classes and associations are automatically named based on the naming information stored in the vocabulary core model. The naming algorithms and the format of the naming information were only described in informal documents. The naming information encapsulated in these documents was distilled into a formal Extended Backus Naur Form (EBNF) grammar, which was used to generate a parser implementation with ANTLR v310.

4. Discussion and Conclusions

The collection of distributed domain knowledge from tools, word processor documents, emails, and domain experts as part of this work led to the formal specification of HL7 business rules. Expressing these rules formally – using standards such as EMF, Ecore,
and EBNF – aids discussion of their intent within the HL7 community and allows future implementers of similar tools to reuse this work in their own applications.

Expressing the semantics of the HL7 domain in Ecore has enabled the cross-validation of interdependent model artifacts, which revealed many problems in the core artifacts describing the underlying standard. Resolutions of these problems were passed back to the HL7 community, resulting in new versions of the MIF and fixes to the core artifacts and individual models. Creating the Ecore meta-model, encapsulating the differences in separate components, and plugging in importer/exporter implementations and validation constraints enabled us to support various flavors of the HL7 standard. The Eclipse platform worked well as a cross-platform host and the model also proved a stable base to generate a platform-independent HL7 V3 modeling tool that incorporates validation of models, transformation to and from various serialization formats, and supports extension and regionalization.

The editor will become the standard HL7 modeling tool in NHS CFH in 2009, and will be the base of the future HL7 modeling tool [12]. Full binaries and source code for the modeling tool and associated Ecore models are available from the Open Health Tools site and tutorial videos are available at the B2 International site.

Acknowledgements. This work was supported and managed by NHS Connecting for Health. Thanks to the HL7 Tooling Committee for their prompt responses and collegial support.

References


11 https://staticmodeldesigner.projects.openhealthtools.org/.