Automatic transformation of HL7 v3 information models into equivalent UML models

David Ortiz, Antonio Villegas, Maria-Ribera Sancho and Antoni Olivé
Department of Service and Information System Engineering
Polytechnic University of Catalonia
Barcelona, Spain
Email: david.ortiz.lopez@est.fib.upc.edu
{avillegas, ribera, olive}@essi.upc.edu

Josep Vilalta
HL7 Education & e-Learning Services
HL7 Spain
Barcelona, Spain
Email: jvilalta@vico.org

***Abstract*—The Health Level 7 (HL7) v3 standard provides information models for the exchange, integration, sharing, and retrieval of electronic health information. Those models are defined in a graphical modeling language other than the UML standard. It involves the development of new tools and the need for specific training in order to support and understand the particularities within HL7 v3 models. We propose an automatic transformation process to obtain UML-compliant models from the MIF representation of the HL7 v3 models within the healthcare standard. The adoption of common UML models benefits the usage of existing modeling tools in the field of model-driven development such as code generators, model validators, and graphical visualizers, among others. We have tested our transformation approach with the models of the HL7 v3 2009 normative edition. The resulting UML models maintain the same semantics as the original v3 models but making use of standard and well-known UML elements.**

***Keywords*-HL7 v3, UML, M2M, model transformation**

I. Introduction

The Health Level 7 (HL7) v3 [1] is a standard for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice and the management, delivery and evaluation of health services. HL7 v3 introduces a model-based specification of messages and documents on the basis of a Reference Information Model (RIM) [2].

For generating an HL7 v3 message or document the information about the concepts involved are defined in a model. HL7 v3 comprises three types of models: RIM, D-MIMs and R-MIMs. The RIM describes the core classes for the health domain as well as the association between those classes and their specializations. A D-MIM is a refined subset of the RIM that includes a set of classes, attributes and relationships that can be used to create messages and structured clinical documents for a particular domain (a particular area of interest in healthcare). Finally, a R-MIM is a subset of a D-MIM that is used to express the information content for a message/document or set of messages/documents with annotations and refinements that are message/document specific. The content of an R-MIM is drawn from the D-MIM for the specific domain in which the R-MIM is used.

The HL7 v3 models are defined in a particular graphical modeling language maintained by the HL7 organization. The standard indicates the information each kind of message or document must contain and how it is structured. The way such messages/documents are shared and implemented is out of the scope of the HL7 standard. However, the v3 models are provided by the standard in two XML-based formats, XML Schema Definition (XSD) and Model Interchange Format (MIF [3]), which are not standard modeling technologies. There are few tools with v3 support and practically none of them reaches a sufficient maturity level to be used in real projects. This situation implies that most of the HL7 v3 developments are XML-centered using XSDs or MIFs to represent the models. It indicates a need to provide such models in a high-level standard modeling technology with enough tool support. Moreover, the particular modeling language in which HL7 v3 models are specified requires specific training in order to understand the characteristics and semantics of v3 models. These problems (and others) produced in HL7 v3 developments are studied in [4], [5].

The Unified Modeling Language (UML [6]) is a standardized general-purpose graphical modeling language in the field of object-oriented software engineering. UML is used to specify, visualize, modify, construct and document software artifacts of a system under development. The adoption of common UML models benefits the usage of existing modeling tools in the field of model-driven development such as code generators, model validators, model verbalizers, and graphical visualizers, among others. Furthermore, the visual notation UML provides is widely known by software development professionals. There exist several attempts to introduce UML for the specification of HL7 v3 models but up to now only the RIM appears to be a consolidated UML-compliant model due to the simplicity of the elements included and its reduced size.

The RIM Based Application Architecture working group (RIMBAA [7]) serves as a focus for those who are interested in using RIM-based information models, and to promote the development of HL7 v3-compliant applications. One of the RIMBAA issues[1] discusses the usage of UML as the modeling language for the HL7 v3 models. There, Spronk says that *'Looking at the experiences of v3 developers, [...] I'd personally suggest that it's probably time that HL7 creates (and publishes) an UML (enriched with OCL) specification of its static model artefacts as well as the data types. Having UML allows implementers to use tons of standard tools'* Also he remarks that *'this doesn't mean that MIF goes away, but just that there's an official transform with published UML files'*.

To address the RIMBAA issue we propose an automatic transformation process to obtain UML-compliant models from the MIF representation of the HL7 v3 models within the HL7 v3 standard. Our transformation is based on the ATL framework [8]. We have tested our transformation approach with the models of the HL7 v3 2009 normative edition. The resulting UML models maintain the same semantics as the original v3 models but making use of standard and well-known UML elements.

The structure of the paper is as follows. Section II introduces the characteristics of model-to-model transformations. Section III describes the required HL7 metamodel for the transformation and explains the usage of MIF files as input of our approach. Section IV presents the rules that are the core of our model transformation and Section V shows the results of the overall process. Finally, Section VI summarizes the conclusions and points out future work. A detailed version of this paper can be found in [9].

II. Model-to-Model Transformation (M2M)

In the context of Model Driven Engineering models are the main development artifacts and model transformations are among the most important operations applied to models. Model-to-model (M2M) transformations translate between source and target models, which can be instances of the same or different metamodels. A metamodel is a model that represents the language and elements from which to form models. There exist different model transformation approaches [10]. Our solution follows the ATL-based transformation infrastructure depicted in Fig. 1.

ATL[2] (ATLAS Transformation Language [8]) is a model transformation language and toolkit developed on top of the Eclipse platform that provides ways to produce a set of target models from a set of source models. ATL transformations are unidirectional, operating on read-only source models and producing write-only target models. During the execution of a transformation the source model may be navigated but changes are not allowed to it.

[1]http://wiki.hl7.org/index.php?title=PIIM

[2]http://www.eclipse.org/atl/

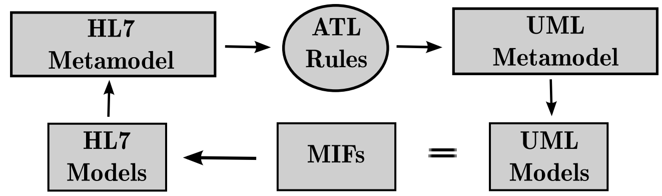

Figure 1. Overview of the automatic transformation process.

The M2M transformation process requires the specification of two metamodels (HL7 and UML) and a set of transformation rules to translate the elements of the first metamodel into elements of the second one. We have developed the HL7 metamodel and the ATL transformation rules. The UML metamodel is already specified in the Eclipse platform[3].

Our transformation starts with the MIF files of the HL7 v3. Those MIF files are processed and converted into instances of a HL7 metamodel, i.e., into HL7 models. Then, the ATL engine executes the transformation rules to translate from the HL7 metamodel into the UML metamodel for each of the initial HL7 models. As a result, the transformation automatically produces a UML model, which is an instance of the UML metamodel, for each HL7 model.

III. HL7 Metamodel and MIF Files

The HL7 v3 standard specification does not contain a model that describes the semantics of all the elements that are present in v3 models. Therefore, a HL7 metamodel is not explicitly included in the standard. We need a HL7 metamodel because it is required for the ATL infrastructure.

We have studied the HL7 documentation and performed a reverse-engineering study through all the v3 models in the standard in order to extract the general characteristics those models share. Since we want to translate v3 models into the standard UML, and we have experience with the UML metamodel, our primary goal was to develop a HL7 metamodel as similar as possible to the UML metamodel. This will simplify the transformation rules of the overall process.

A simplified version of our HL7 metamodel is presented in Fig. 2. Those elements like *Class*, *Property* or *Association* are directly extracted from the UML metamodel, with minor changes. However, there are new elements like *Choice*, *CMET* or *EntryPoint* that appear only in HL7 models. We have chosen the MIF files as the input of our model transformation because MIFs contain more information and are closer to the graphical representation of the v3 models than XSDs.

The effort to develop a HL7 metamodel allows us in the process of transforming the MIF files from the standard into HL7 models that are instances of our HL7 metamodel.

[3]http://www.eclipse.org/uml2/

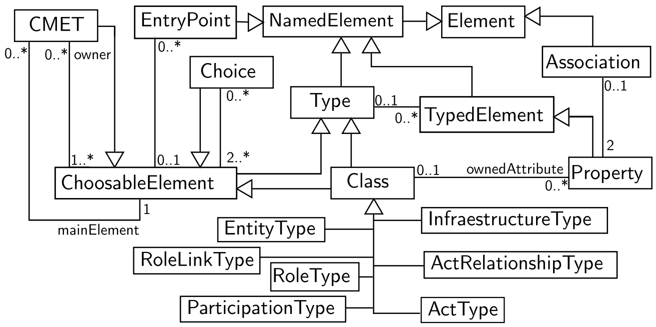

Figure 2. Simplified version of the HL7 metamodel.

We have developed the HL7 metamodel as an Eclipse EMF Ecore[4] model, which allows us to automatically generate a complete API to create instances of our HL7 metamodel. We have processed the MIF files and created the corresponding HL7 models through the metamodel API. Those HL7 models from the MIF files are the required input of the M2M transformation.

IV. Transformation Rules

Transformation rules are the key component of an ATL-based model transformation. Each rule deals with an element of the source metamodel and describes its equivalence in the target metamodel. Figure 3 indicates the effect of the transformation rules for the main elements of the HL7 metamodel.

A. Classes

A HL7 v3 class can be classified as an *Act*, *ActRelationship*, *Participation*, *Role*, *RoleLink*, *Entity* or as *Infraestructure* class. We transform each HL7 class into an UML class with the same name and attributes as the original but with an stereotype with the name of its kind of class.

B. Associations

An HL7 association is transformed into an UML association with the same name, participants, role names and multiplicities as the original one.

C. Entry Points

An entry point indicates the main element in a HL7 model. We transform each entry point into an abstract UML class with the stereotype `<<EntryPoint>>` and an attribute named *description* with the original information of the entry point.

[4]http://www.eclipse.org/modeling/emf/

D. Choices

A HL7 choice encloses two or more classes that are part of an inheritance hierarchy. Any associations connected to the Choice apply to all classes within it. We transform each Choice into an abstract class with the same name as the choice and the stereotype `<<Choice>>`. The classes within the Choice are transformed into regular UML classes with a generalization relationship with the choice class. Any association connected to the Choice is now connected to the abstract class representing the choice in UML.

E. CMETs

CMETs are pre-defined components that are re-used for several R-MIMs in order to avoid repetitions of common HL7 elements. We transform each CMET into an UML class with the name of the CMET and a read-only attribute named *identifier* whose value is the original identifier of the CMET. Also, such class is marked with the stereotype `<<cmet>>` and the stereotype of the main class of the original CMET.

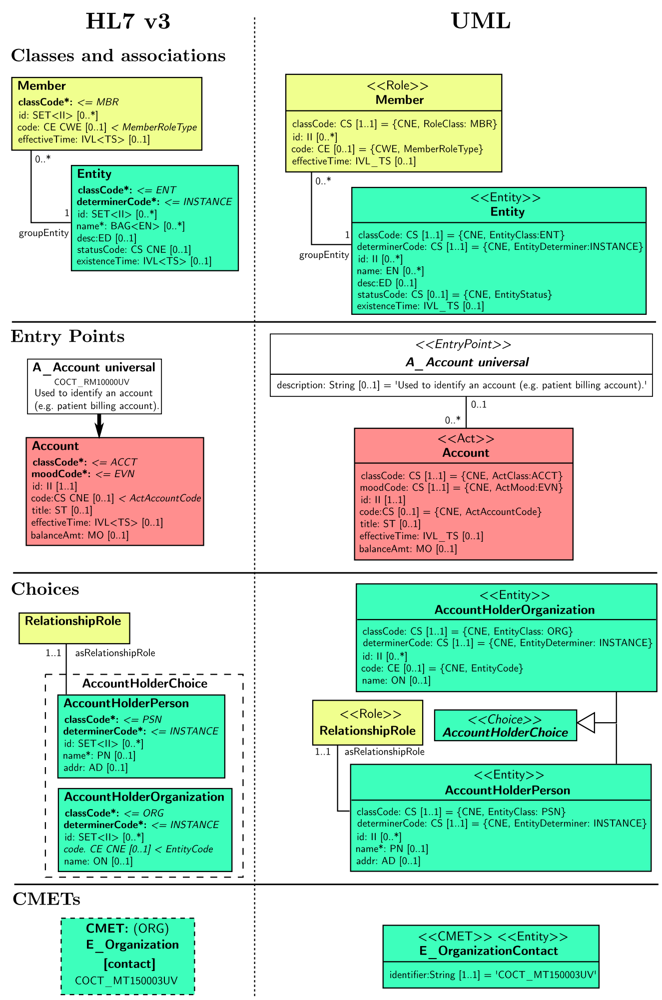

Figure 3. UML translation for the HL7 v3 complex elements.

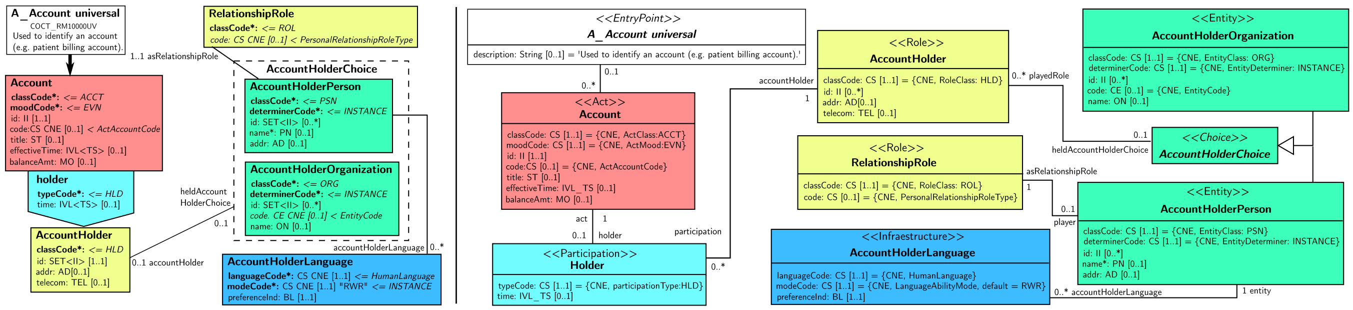

Figure 4. HL7 v3 (left) and UML (right) versions of the COCT_RM10000UV R-MIM model.

V. Transformation Results

We have tested our automatic transformation process with the HL7 v3 2009 normative edition, which contains 379 MIF files that have been translated into UML. The overall process took around 5 minutes in a computer with a Pentium 4 3GHz processor and 2Gb RAM. Since new HL7 v3 normative editions appear once a year, and preliminary versions appear once a month, the execution time of our transformation is acceptable. The 379 resulting UML models have no errors and are semantically identical than the original MIFs. Figure 4 shows a comparison between the HL7 v3 and UML versions of the COCT_RM10000UV R-MIM model. Both models are semantically identical.

VI. Conclusions and Further Work

HL7 v3 models are specified in a particular modeling language that requires special training and tool support. What is needed is an automatic process to transform the original HL7 v3 models into equivalent models specified by means of the standard UML. We have presented an ATL-based model-to-model transformation that easily translates the models of the healthcare standard into UML models. Input to our process is the set of MIF files that are converted into HL7 models that are instances of our HL7 metamodel. The core of the process are the ATL transformation rules that translate each element within the source HL7 models into elements of the target UML models. Our transformation allows the easy adoption of the HL7 v3 standard for those familiar with the UML, and the usage of a big amount existing tools to work with UML models.

We plan to continue our work with the study of the textual constraints that appear in the graphical view of the models represented in the MIF files. Such constraints are expressed in natural and formal language and should be corrected, normalized and rewritten in a standard constraint language in order to be easily included in our model transformation process. We are also open to suggestions to improve our work and make it of use for the HL7 community.

Acknowledgment

The authors would like to thank the collaboration of Diego Kaminker, HL7 Education WG co-chair and HL7 International Mentoring Committee co-chair. We would also like to thank the people of the GMC group for their useful comments to previous drafts of this paper. This work has been partly supported by the *Ministerio de Ciencia y Tecnología* under the project TIN2008-00444, *Grupo Consolidado*, and by Polytechnic University of Catalonia under FPI-UPC program.

References

[1] G. Beeler, "HL7 Version 3–An object-oriented methodology for collaborative standards development," *International Journal of Medical Informatics*, vol. 48, no. 1-3, pp. 151–161, 1998.

[2] G. Schadow, C. Mead, and D. Walker, "The HL7 Reference Information Model under scrutiny." *Studies in Health Technology and Informatics*, vol. 124, pp. 151–156, 2006.

[3] R. Spronk, "The HL7 MIF - Model Interchange Format," http://www.ringholm.de/docs/03060_en_HL7_MIF.htm, February 2010.

[4] B. Blobel, K. Engel, and P. Pharow, "Semantic interoperability–HL7 Version 3 compared to advanced architecture standards." *Methods of Information in Medicine*, vol. 45, no. 4, p. 343, 2006.

[5] R. Spronk, "Navigating the pitfalls: Implementing HL7 version 3," http://www.ringholm.de/docs/04100_en.htm, December 2006.

[6] OMG, *Unified Modeling Language: Superstructure, version 2.3*, Object Modeling Group http://www.omg.org/spec/UML/2.3/, May 2010.

[7] RIMBAA, "RIM Based Application Architecture working group," http://wiki.hl7.org/index.php?title=RIMBAA.

[8] F. Jouault, F. Allilaire, J. Bézivin, and I. Kurtev, "ATL: A model transformation tool," *Science of Computer Programming*, vol. 72, no. 1-2, pp. 31–39, 2008.

[9] D. Ortiz, "Transformación de modelos del estándar de salud HL7 a UML/OCL *(spanish)*," http://hdl.handle.net/2099.1/12477, June 2011.

[10] K. Czarnecki and S. Helsen, "Classification of model transformation approaches," in *Proceedings of the 2nd OOPSLA Workshop on Generative Techniques in the Context of the Model Driven Architecture*, 2003, pp. 1–17.