

A Methodological Proposal and Tool Support for the HL7 Standards Compliance in the Development of Health Information Systems

A. Martínez-García¹, M.A. Olivero², A. Suárez-Bote², J.M. Sánchez-Bejines²,
F.J. Domínguez-Mayo², M.J. Escalona², M. Mejías², C.L. Parra-Calderón¹

¹Group of Research and Innovation in Biomedical Informatics, Biomedical Engineering and Health Economy. Institute of Biomedicine of Seville, IBiS / Virgen del Rocío University Hospital / CSIC / University of Seville, Seville, Spain
Av. Manuel Siurot, s/n, 41013, Seville (Spain)
{alicia.martinez.exts, carlos.parra.sspa}@juntadeandalucia.es

²Web Engineering and Early Testing Group, Computer Languages and Systems Department, University of Seville
ETSII, Avda. Reina Mercedes S/N, 41012, Seville (Spain)
{almudena.suarez, miguel.olivero, juan.sanchez}@iwt2.org
{fjdominguez, mjescalona, risoto}@us.es

Abstract

Health information systems are increasingly complex, and their development is presented as a challenge for software development companies offering quality, maintainable and interoperable products. HL7 (Health level 7) International, an international non-profit organization, defines and maintains standards related to health information systems. However, the modelling languages proposed by HL7 are far removed from standard languages and widely known by software engineers. In these lines, NDT is a software development methodology that has a support tool called NDT-Suite and is based, on the one hand, on the paradigm of model-driven engineering and, on the other hand, in UML that is a widely recognized standard language. This paper proposes an extension of the NDT methodology called MoDHE (Model Driven Health Engineering) to offer software engineers a methodology capable of modelling health information systems conforming to HL7 using UML domain models.

Keywords: HL7, UML, Model-Driven Engineering, health information systems, MoDHE

1.0 Introduction

Today, health information systems are increasingly complex [1]. Developing quality, maintainable and interoperable products is a challenge for software development companies wishing to find a market in healthcare systems. The need for a shared clinical history at the global level is a reality [2]. For this, it is essential to use health informatics standards that allow the establishment of standards for the exchange of clinical information [3].

There are proposals such as NDT [4], which are included in the paradigm of model-driven engineering, covering the requirements and analysis phases of web systems. However, NDT has a generalist character, therefore, in this paper we present in detail the methodological proposal that we have named MoDHE (Model Driven Health Engineering) because of its close relation with the MDE paradigm, the health field, and because it is based on the NDT methodology. Nowadays, this methodology focuses on the requirements definition phase, since it is the most critical phase in clinical projects because there are many HL7 standards that apply to that specific phase of software engineering, standards that must be applied to obtain software Interoperable health. Once the requirements phase is solved, the rest of the life cycle phases of the health software could be developed in the way equivalent to any other non-sanitary software project. In any case, the methodology has been developed with the capacity to be extended to the rest of the life cycle phases.

This methodology allows the software engineer to systematically model health information systems by working on UML diagrams and ensures compliance with HL7 standards including extension mechanisms that allow for any existing standards in HL7.

The MoDHE methodology is based on 3 main pillars. The first pillar, the methodology, offers a procedure that allows designing HL7 domain models as part of the development of a health information system. The second pillar, the HL7-based modelling language, extends to UML to model health information systems conforming to the full spectrum of HL7 standards. The third pillar, the derivation mechanisms, makes possible the interoperability between standards, facilitating the maintainability and extension of the systems. At the time of develop a first version of the methodology of MoDHE, it was decided to work with 3 of the main standards of HL7: v3, CDA and v2.x.

To automate this frame of reference, a support tool has been developed, registered as MoDHE Suite. The main objective of the tool in which we focus on this paper is to support the development of health information systems conforming to HL7, through a framework of reference that allows approaching the standards of HL7 and the modelling language of general purpose UML, using the MDE paradigm. The MoDHE methodology makes real this framework. Section 2 of this article describes the HL7 standard and its relation to UML, as well

as what is NDT. In section 3 we talk about our proposed NDT extension, MoDHE. In section 4 we present the tool that supports this methodology, MoDHE Suite, concluding with the conclusions in section 5.

2.0 Related work and context

This article proposes the joint use of two initially independent worlds of wide relevance in software engineering applied to the health environment: UML [5] (Unified Modeling Language) and HL7 (Health Level Seven).

2.1 UML y HL7 (CDA, v3 y v2.x)

UML [6] is the standard modelling language proposed by the OMG (Object Management Group) [7], an organization that promotes the use of object-oriented technologies by creating and maintaining guidelines, standards and specifications. A domain model is a conceptual model that describes the entities, attributes, roles, relationships, and constraints related to the domain of the problem [8, 9]. Instead of describing concepts of a software system, it describes the concepts of the reality of the problem itself. On the other hand, MDE (Model Driven Engineering) is a paradigm that focuses on the creation and exploitation of domain models, allowing software engineers to become independent of representation and to focus on concepts [8, 9]. In this way, a metamodel describes the concepts used in a particular domain model [8, 9]. When representing metamodels, there are many accepted notations. One of the most commonly used notations is UML.

On the other hand, HL7 defines domain models in each of its standards [10], ranging from information necessary to define messaging between systems, to the clinical documents themselves, in order to represent each of the problems or work scenarios that HL7 has gone identifying over time [11]. The HL7 standards have a common metamodel, called MIF (Model Interchange Format), from which all HL7 domain models can be modelled [12]. The MIF is formally defined in one of the HL7 standards [13]. It should be noted that the MIF is so extensive and presented in such an abstract way that, although it is very interesting from a conceptual point of view, it can cause much difficulty in its management and learning.

Each HL7 standard has an underlying metamodel, which specializes and extends the MIF. In some cases, the metamodel is not explicitly defined in a diagram, but is defined verbally in different documents. In other cases, this metamodel is explicitly defined in diagrams that use a graphic language of HL7. Given that the metamodels of some HL7 standards are described verbally in large documents, and that other HL7 standards are modelled using their own graphical language, we think that it is not easy for a software engineer to design the domain model of a software solution conforming to a specific HL7 standard. Unlike HL7 standards, software engineers in general are comfortable with more general modelling languages, such as UML.

The HL7 CDA and HL7 v2.x standards are the most used in Spain [14]. The HL7 v3 standard is the reference model where, by refinement of a subset of RIM (Reference Information Model) elements, domain models such as the HL7 CDA itself are generated. Therefore, when deciding which standards to include as part of the MoDHE framework, it was decided to work on these 3 standards. The HL7 v3, Version 3 Product Suite, consists of a set of RIM-based specifications that provide implementers with the necessary resources to work with messages, data types, and terminologies. It is considered a more robust standard than HL7 v2.x, since it reduces the semantic ambiguity and improves the processes, by having an underlying information reference model (this reference model is the RIM).

HL7 CDA is a document marking standard that specifies the structure and semantics that any clinical document must meet to be exchanged between health care providers and / or patients. This standard works on the R-MIM information model, which is a subset of RIM.

2.2 NDT

NDT is a methodology based on model-driven engineering that provides formal and complete support for software lifecycle management (feasibility study, requirements, analysis, design, implementation, maintenance, testing) [4]. Using NDT, we can cover the phases of the software engineering life cycle in a structured way, reducing errors and redundancies.

The NDT methodology extends the UML metamodels, supporting the design of models in each phase of the software life cycle, representing these models using UML diagrams. To support the limitations identified in each phase of the software life cycle, it defines constraints. In addition, it defines transformations between models, allowing to automatically generate the model of a specific phase, taking the information previously modeled from diagrams or models of previous phases.

Most HL7 standards are framed in the NDT Requirements Engineering phase (DRS phase). Therefore, of all the phases covered by NDT, in this paper we focus on the DRS phase. In addition, of all phases of the software life cycle covered by the NDT methodology, the DRS phase is the most complete because NDT covers a large percentage of the needs presented by this phase of the software life cycle. The final objective of the DRS phase is to model a catalog of requirements that define the needs of the system, establishing these requirements cataloged per their typology, without entering aspects related to development.

In this paper, efforts have focused on three HL7 standards that can correspond to the software information requirements (as part of the DRS phase of NDT), defining the static or structural part of the system. These standards are HL7 v3, HL7 CDA and HL7 v2.x. These 3 standards can be matched with elements modeled in the

'Data Storage Requirements' (RA) of the NDT DRS phase. Therefore, to encompass these 3 HL7 standards, the NDT metamodel corresponding to the DRS phase has been extended, adding the proper elements of each standard.

3.0 The Model Driven Health Engineering (MoDHE) framework

The MoDHE methodology enables the software engineer to systematically model health information systems by working on UML diagrams, and ensures compliance with HL7 standards including extension mechanisms that allow for any existing HL7 standards to be included. This methodology is part of one of the three pillars defined in the previous section. Thanks to this methodological proposal once the software engineer has modeled the information requirements according to HL7 using the methodology of MoDHE, you can automatically generate the analysis phase of the system and the later phases of the software life cycle as if of a project Non-health software. In this way, we can systematize the development process.

The MoDHE methodology extends to the NDT methodology, extending the metamodels that cover the elements of the software life cycle, and contemplating the metamodels of the HL7 standards, thus allowing a formal and complete framework that allows modeling a system of Health information according to HL7 in a systematic way using UML models. The MoDHE methodology allows a software engineer to model requirements using the UML language, defining in a transparent and systematic way the HL7 requirements. The following figure (Figure 1) shows the overall process.

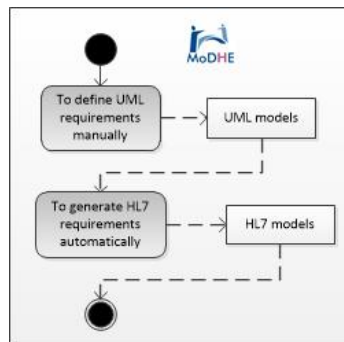


Figure 1: Global process

The NDT methodology extends the UML metamodels, supporting the design of models in each phase of the software life cycle, representing these models using UML diagrams. To support the limitations identified in each phase of the software life cycle, it defines constraints. In addition, it defines transformations between models, allowing to automatically generate the model of a specific phase, taking the information previously modeled from diagrams or models of previous phases.

Using the MoDHE methodology, the software engineer, when constructing a sanitary software system, would perform the same phases as when using the NDT methodology (EVS, DRS, DAS, DDS, Construction, DPS, DMS), with the difference That in the DRS phase, in addition to defining non-health information requirements (and other types of requirements, such as objectives, actors, new natures, etc.), it would define health information requirements conforming to HL7. It should be noted that, considering that the NDT methodology uses UML notation, the learning curve of the software engineer using NDT for the first time is minimal.

The MoDHE methodology extends the DRS metamodells of NDT. The DRS phase of NDT has as its final objective to model a catalog of requirements that defines the needs of the system, establishing these requirements catalogued according to their typology, without entering into aspects related to development. Specifically, MoDHE extends the storage requirements (RA) metamodel to include elements of the HL7 standards. Within the RA metamodel, the MoDHE methodology focuses on information requirements. Thanks to this feature, a software engineer can define the catalog of health requirements according to HL7.

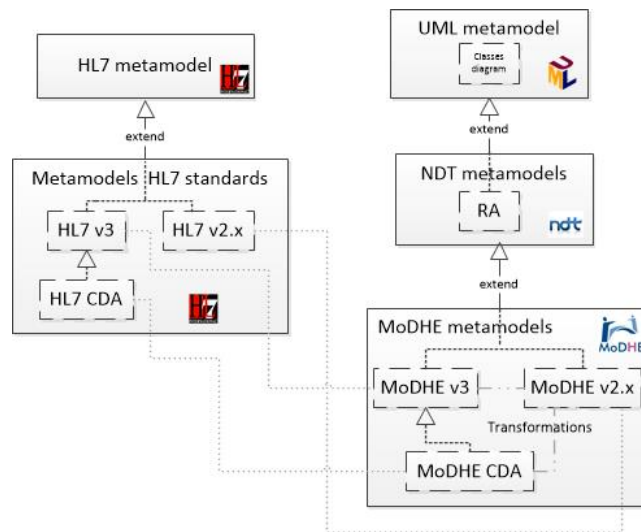


Figure 2: Formal definition of the approach

Health software, from the point of view of the software life cycle, runs the same process as any other software development. Each HL7 standard provides guidelines and recommendations focused on a specific phase of the software lifecycle [15]. Therefore, the MoDHE methodology has been developed as an extension of the NDT methodology, supporting the development of all phases of the health software lifecycle as can be viewed on Figure 2. It should be noted that, focusing on the modelling of health requirements, it is not mandatory to design the models according to all 3 HL7 standards, the standards to be used will depend on the concrete scenario.

Considering that the HL7 v3, HL7 CDA, HL7 v2.x standards have the common characteristic that they cover the definition of information requirements of a health system, common elements in those standards are identified. These common elements allow you to define simple transformations between entities from one standard and another, allowing you to partially create the structure of the model based on a standard taking as input the model based on another standard. , Taking as reference the formal definition of the metamodels that has been realized in other studies, these can be analysed and formalized the established semantic relations, by means of which the target model can be obtained following a specific source model. This process, considering such semantic relations, establishes a rule-based transformation mechanism to obtain the final model. Thanks to this transformation process, a traceability between the two metamodels is established, allowing the automation of development, as well as improving the quality and consistency of the models. For example, if you have modelled a model based on the MoDHE v2.x metamodel, you can automatically generate part of the MoDHE CDA model with the same information previously modelled by the MoDHE v2.x metamodel.

By implementing these transformations between the different MoDHE metamodels (each representing an HL7 standard), the MoDHE methodology allows reuse of models already created in an HL7 standard when modelling requirements per another HL7 standard, reusing the information, removing duplication, redundancy, and reducing errors. In addition, based on the NDT methodology, the MoDHE methodology covers the entire software lifecycle, including modelling non-health requirements in the requirements engineering phase.

4.0 MoDHE Suite: A tool to support the framework

The MoDHE framework, as we have seen, allows the design of HL7 domain models based on UML, using techniques based on the MDE paradigm. This framework is composed of the methodology and metamodels needed to make this design, as well as the transformations necessary to create models by taking the information previously modeled in other models. To make possible the practical use of this theoretical framework that allows us to generate UML models conforming to HL7, it is necessary to have a CASE support tool. This tool, which we have called MoDHE Suite has been implemented in C # as a plugin for Enterprise Architect.

To develop the MoDHE Suite tool, this research has been used as an EA basis, an already existing modeling tool that provides extension mechanisms through plugins, as well as because this modeling tool is widely known by companies and organizations in the that MoDHE Suite can be validated and evaluated once it is set up.

The definition of the specific syntax has been made using UML profiles, more specifically, UML version 2.5 [5] has been used. A UML profile is a formal extension of the UML language itself with the objective of defining new concepts from existing UML constructors, to provide them with a more precise and concrete semantics. It has been chosen to use a UML extension as a mechanism to define the concrete syntax of the UML and HL7 metamodels, since there is no problem with the use of a new specific language.

The UML extension protocol is based on 3 mechanisms:

- Stereotypes. Thanks to the stereotypes it is possible to define each of the elements of a specific domain that in turn will extend specific UML metaclasses.
- Tagged value. Tagged values allow you to add particular properties to any defined stereotype within the profile.
- Constraints. The constraints define the semantic conditions that apply to the stereotypes of the profile and that condition the instantiation of the metamodel.

In developing all UML profiles for MoDHE, all stereotypes extend the UML Class metaclass. It has been chosen to extend this UML metaclass in question because this metaclass aims to specify a classification of objects and to specify their properties (attributes, operations, associations, etc.), characterizing the structure and the context of these objects. Thus, when modeling an element of HL7, attributes, associations, etc. can be defined using a nomenclature similar to that used by the UML class metaclass, thus reducing the learning curve in the use of the methodology and the tool one, and improving the usability of these.

Considering that the MoDHE methodology extends the NDT methodology, the MoDHE Suite tool is based on the implementation of the NDT-Suite tool, which in turn uses the software architecture provided by EA. One of the most important aspects in the development of the MoDHE Suite tool is the translation of the concrete syntax within EA. For this, the MDG Technologies module of EA has been used. In the case of the MoDHE Suite tool, since it is an extension of the NDT-Suite tool, an MDG Technology project has not been created from scratch, failing which the 5 existing MDG Technology projects have been used for the implementation of the tool NDT-Suite, corresponding to the following phases of software development: requirements, analysis, design, testing, maintenance.

For each MDG Technology project, EA creates 2 packages by default:

- Toolbox package. It contains the set of stereotypes that make up the UML profile, along with its tagged values. Each of these stereotypes must be linked to the appropriate UML metaclass through a "extend" relationship. Each of the contemplated stereotypes is defined by means of a set of tagged values, which correspond to the attributes of the stereotypes defined in the UML profile.

- Profile package. It contains all those EA artifacts necessary to define the creation of diagrams according to certain stereotypes defined in the previous package, and previously selected. From this set of artifacts, the user is given the ability to model following a UML profile.

When configuring our plugin, and running EA, we have a new menu that allows modeling in UML health information requirements conforming to HL7 standards. It should be noted that the MoDHE Suite toolbox for modeling information requirements, in addition to including the elements of HL7 standards, includes the necessary elements to model non-health information requirements (subsystems, storage requirements, new natures) elements of the NDT-Suite tool.

The EA plugin for MoDHE Suite includes the profile implemented in EA, and implements the methods necessary to verify that the constraints specified in this section satisfy. It also implements the transformations defined for this tool. These transformations have been previously and theoretically defined in QVT language [16].

The QVT language is a standard language proposed by the OMG for the definition of M2M transformations. This language was born at the end of 2005 as a common proposal of several research institutions and companies. For the definition of the structure and syntax of the metamodels, this language is based on the specifications of the MOF [17] and OCL [18] standards proposed by the OMG. The use of QVT against other transformation languages such as ATL (Atlas Transformation Language) is due to the proposal presented is an extension of the NDT methodology, which uses specifications of transformations in QVT, therefore, using QVT will improve the Compatibility between both sets of transformations (those of NDT, and those of MoDHE).

These transformations have finally been implemented in C # language. These transformations could have been implemented alternatively in any other general purpose language such as Java, Python, etc.

To cover this development, first, the concrete syntax of the metamodels has been defined, and secondly the code that models the concepts of the metamodels and the transformations previously defined in QVT has been implemented.

5.0 Conclusions

In this paper, we have presented the Model Driven Health Engineering (MoDHE) framework. This proposal is an extension of NDT that is based on 3 main pillars. The first pillar, the methodology, offers a procedure that allows the design of HL7 domain models as part of the development of a health information system. The second pillar, the HL7-based modelling language, extends to UML to model health information systems conforming to the full spectrum of HL7 standards. The third pillar, the derivation mechanisms, make possible the interoperability between

standards, facilitating the maintainability and extension of the systems. At the time of developing the methodology of MoDHE, it was decided to work with 3 of the main standards of HL7: v3, CDA and v2.x

To automate this reference frame, a support tool has been developed, registered as MoDHE Suite. This tool allows the design of HL7 compliant domain models using a UML-based interface. It also allows the generation of models of a specific standard based on existing models of another specific standard. This tool has been validated in a real case study extracted from a project in which the Technological Innovation Group of Virgen del Rocío University Hospital participated, demonstrating that the solution developed is very useful.

The present paper proposes the development of a reference framework that facilitates the design of HL7-compliant domain models using a UML-based interface.

6.0 Acknowledgments

This research has been partially supported by the POLOLAS project (code TIN2016-76956-C3-2-R) of the Spanish Ministry of Science and Innovation, the KNOWBED project (code PIN-0213-2016) founded by the Andalusian Regional Ministry of Health, and Carlos III Health Institute within the call Strategic Help in Health (PITeS TliSS project, code PI15/01213), and FEDER funds.

The authors are grateful to Carlos III Health Institute for promoting the Network for Innovation in Medical Technologies and Health ('Plataforma ITEMAS' in Spanish, CODE PT13/0006/0036). Finally, we would also like to thank Universia Foundation for awarding PhD students with research grants.

7.0 References

- 1 Kushniruk, A. W., Borycki, E. M., Kuwata, S., & Kannry, J. (2010). Emerging approaches to usability evaluation of health information systems: towards in-situ analysis of complex healthcare systems and environments. *Studies in health technology and informatics*, 169, 915-919.
- 2 Garde, S., Knaup, P., Hovenga, E. J., & Heard, S. (2007). Towards Semantic Interoperability for Electronic Health Records--Domain Knowledge Governance for open EHR Archetypes. *Methods of information in medicine*, 46(3), 332-343.

- 3 Sinha, P. K., Sunder, G., Bendale, P., Mantri, M., & Dande, A. (2012). Electronic health record: standards, coding systems, frameworks, and infrastructures. John Wiley & Sons.
- 4 Jose Escalona, M., & Aragón, G. (2008). NDT.A model-driven approach for web requirements. *Software Engineering, IEEE Transactions on*, 34(3), 377-390.
- 5 OMG (2015, marzo). *OMG Unified Modeling Language, v2.5*. Retrieved March, 2017, <http://www.omg.org/spec/UML/>
- 6 UML (2017, March). Object Management Group. Retrieved March, 2017, <http://uml.org/>
- 7 OMG (2017, March). Object Management Group. Retrieved March, 2017, <http://www.omg.org/>
- 8 Van Der Straeten, R., Mens, T., & Van Baelen, S. (2009). Challenges in model-driven software engineering. In *Models in Software Engineering* (pp. 35-47). Springer Berlin Heidelberg.
- 9 Schmidt D.C. (2006). Model-Driven Engineering. *Computer*, 39(2), 25-31.
- 10 HL7 Standards (2017, March). Section 3: Clinical and Administrative Domains. Retrieved March, 2017, http://www.hl7.org/implement/standards/product_section.cfm?section=3
- 11 HL7 Wiki (2017, March). Domain Analysis Model. Retrieved March, 2017, http://wiki.hl7.org/index.php?title=Domain_Analysis_Model
- 12 Spronk, R., & Ringholm, C. (2010, febrero). The HL7 MIF-Model Interchange Format.
- 13 HL7 International (2011). *Model Interchange Format, Release 1. Informative Document - August 2011*
- 14 HL7 International (2016). *Affiliate Spotlight: HL7 Spain*. The official publication of Health Level Seven International News. Page 19.
- 15 HL7 Reference Information Model (2017, March). Retrieved March, 2017, <http://www.hl7.org/implement/standards/rim.cfm>
- 16 OMG (2015, February). *Documents Associated with Meta Object Facility (MOF) 2.0 Query/View/Transformation, v1.3*. Retrieved October, 2016, <http://www.omg.org/spec/QVT/>.
- 17 OMG (2015, junio). *Meta Object Facility (MOF) Core, v2.5*. Retrieved October, 2016, <http://www.omg.org/spec/MOF/>.
- 18 OMG (2014, February). *Object Constraint Language, v2.4*. Retrieved October, 2016, <http://www.omg.org/spec/OCL/>.