Working with the HL7 metamodel in a Model Driven Engineering context

A. Martínez-García [a], J.A. García-García [b], M.J. Escalona [b], C.L. Parra-Calderón [a]

[a] *Technological Innovation Section, Virgen del Rocío University Hospital, Manuel Siurot Avenue, s/n, 41013 Seville, Spain*
[b] *Web Engineering and Early Testing Research Group, University of Seville, ETS Ingeniería Informática, Av. Reina Mercedes s/n, 41012 Seville, Spain*

Keywords:
HL7
Metamodel
Model Interchange Format
Model Driven Engineering
UML

A B S T R A C T

HL7 (Health Level 7) International is an organization that defines health information standards. Most HL7 domain information models have been designed according to a proprietary graphic language whose domain models are based on the HL7 metamodel. Many researchers have considered using HL7 in the MDE (Model-Driven Engineering) context. A limitation has been identified: all MDE tools support UML (Unified Modeling Language), which is a standard model language, but most do not support the HL7 proprietary model language. We want to support software engineers without HL7 experience, thus real-world problems would be modeled by them by defining system requirements in UML that are compliant with HL7 domain models transparently. The objective of the present research is to connect HL7 with software analysis using a generic model-based approach. This paper introduces a first approach to an HL7 MDE solution that considers the MIF (Model Interchange Format) metamodel proposed by HL7 by making use of a plug-in developed in the EA (Enterprise Architect) tool.

1. Introduction

HL7 (Health Level 7) International [1] is a non-profit organization that promotes and defines standards associated with health information systems. HL7 International members develop standards related to the exchange and model of Health information, with the objective of supporting clinical practice, management, development, and evaluation in Health services. This set of standards is known as HL7 standards, or simply, HL7.

A domain model is a conceptual model that describes concepts related to the problem domain [2,3]. It copes with concepts linked to the problem itself, instead of describing software system concepts. MDE (Model Driven Engineering) is a new paradigm that centers on creating and exploiting models [2,3]. Using MDE, productivity is increased because compatibility among systems is maximized (thanks to reutilization), thus simplifying the design process. Models act as system bases. This way, the conceptual definition of applications can be separated from the technology where they are executed. For this purpose, metamodel is a fundamental concept because it describes the concepts used in a specific model. There are many accepted notations to represent metamodels. In this case, we use UML (Unified Modeling Language)-class diagrams because they are the notations applied to both HL7 and UML.

HL7 has a metamodel called MIF (Model Interchange Format) [4]. This metamodel is not compliant with UML. In addition, HL7 International has developed its own graphic language to design the elements that compose its models. Considering the wide range of entities that MIF needs to cover in order to collect all the concepts necessary in a general health system, we must argue that MIF is very extensive and is presented in such an abstract way that, although it seems very interesting from the conceptual perspective, it can be difficult to manage.

HL7 International defines different domain models to explain each working problem or scenario that has been identified throughout the process. These conceptual schemes cover all areas that range from the information necessary to define system messages to the clinical documents themselves. All HL7 domain models can be modeled from MIF.

Considering that HL7 models are built in their own graphic language, and regarding the extension they present to cover all the entities necessary in a health system, we conclude that designing

Abbreviations: HL7, Health Level 7; MDE, Model Driven Engineering; UML, Unified Modeling Language; MIF, Model Interchange Format; NDT, Navigational Development Techniques; CDA, Clinical Document Architecture; OMG, Object Management Group; EA, Enterprise Architect; M2M, model-to-model transformations; ADL, Archetype Definition Language; AML, Archetype Modeling Language; RIM, Reference Information Model; HDF, HL7 Development Framework.

a software solution that can fulfill an HL7 standard is not an easy task for a software engineer. The fact that HL7 has a metamodel in a proprietary format produces much impact in the industry because lack of a commercial tooling support is identified, and a smaller knowledge field is produced simply because this notation is not a typical subject taught to software engineers at the university. In contrast, software engineers generally know UML and can design solutions through this standard. In addition, many MDE tools exist that perform a series of actions automatically, such as generating code or documentation, through a UML model.

Therefore, working to connect HL7 with software analysis has been relevant for us. Our long-term objective is for software engineers to design their solutions using the UML metamodel and the HL7 metamodel automatically. Consequently, we offer the capability of using standard MDE tools that need the problem to be modeled with UML modeling, apart from simplifying the solutions design.

This article lays the foundation for this research that we have recently started, and is motivated by our previous experiences, such as Diraya Specialized Attention project [5,6] and the eHealth project [7]. On the one hand, we performed a practical experience in the MDE context in the first project, which consisted in applying NDT (Navigational Development Techniques) Web Engineering methodology [8] when performing the Requirements and Analysis phases in a large-scale Web system focused on supporting Health information systems in Andalusia. On the other hand, the second project aimed to adapt the eHealth platform of the Virgen del Rocío University Hospital of Seville to a process-based SOA (Service-Oriented Architecture) to allow greater modularity, independence, maintainability, and usability for the development of functional modules that provide support to the clinical services of this hospital. For this purpose, we defined a model-driven proposal supported by automatic software tools.

These experiences concluded that MDE can reduce development time and identify possible errors or inconsistencies in early phases.

The main target of the research presented in this paper is to use the HL7 metamodel in the MDE context.

Fig. 1 illustrates the general process we aim to reach with this study.

Our secondary goals are as follows:

- To provide software engineers involved in the healthcare area with a solution that employs the benefits of the UML general proposed standard, standards recommended by HL7 International, and MDE existing tools.
- To take advantage of the potential of the existing tools that work with the new domain models exploitation paradigm: MDE.

This paper is structured as follows: After this introduction, Section 2 reviews and presents previous experiences. Then, Sections 3 and 4 explain the methodology used and the results obtained, respectively. Finally, Section 5 provides further discussion, and Section 6 states final conclusions.

2. Previous experiences

Some members of the HL7 International community have experienced the need of using a modeling standard instead of the modeling language that defines the domain models generated from MIF.

Previous experiences have studied the connection between HL7 v2.X and UML structures [9]. One of the first steps to use HL7 in the MDE context consists in implementing MIF in a computer-workable language. There are cases related to implementing computer-workable languages of a specific domain model, for example HL7 v3, but they do not cover the HL7 metamodel completely [10].

Researchers from the Polytechnic University of Catalonia have conducted an experiment in this domain. They identified some weaknesses while using the HL7 modeling language, and proposed a translation of the HL7 domain models to UML nomenclature in order to overcome such weaknesses. The researchers even implemented a translation from the HL7 v3 domain model to UML models [11]. Finally, they concluded that the HL7 International community could not find the UML model sufficiently suitable to replace the original MIF, and therefore, they could reject its adoption.

Since 2012, Sparx Systems has sponsored the HL7 Tooling Challenge, a yearly contest aiming to encourage the development of

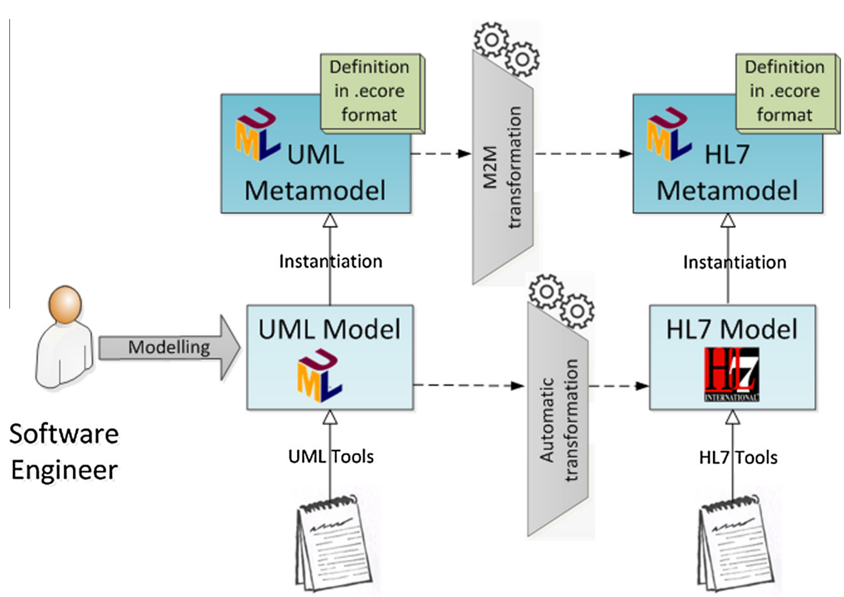

Fig. 1. Solution using the HL7 metamodel in the MDE context.

HL7 tools. The 2012–2013 edition of this contest aimed to generate a set of tools in order to support developers in the implementation of MIF-based HL7 v3 standards. The winners of this edition were the aforementioned researchers from the Polytechnic University of Catalonia. The 2013–2014 edition aimed to produce the design specifications of a tool capable of generating RIM-based information models. The winners of this edition have not been released, yet.

There is an open source project called Model-Driven Health Tools that has developed a common modeling framework in order to improve interoperability in the healthcare infrastructure, thus allowing the creation of computable models of the CDA (Clinical Document Architecture) templates in UML, which is based on MDE. Currently, this project has already built models from some existing HL7 specifications (continuity of care document, public health case reports, personal healthcare monitoring report, etc.), and some researchers have successfully used this set of tools [12,13]. The difference between this experience and our proposal is that they centered their effort on one of the HL7 primary standards, CDA. However, our proposed solution covers the entire spectrum of concepts that exists in the HL7 metamodel.

No previous studies that intend to use HL7 in the MDE context making a correspondence between the elements of HL7 and UML metamodels are found, in which the software engineer can use UML directly (being able to get help from all existing UML tools in the MDE context) and work automatically on the HL7 metamodel.

3. Materials and methods

3.1. The HL7 metamodel

All HL7 domain models are simply HL7 metamodel instances. Consequently, such metamodel represents the general knowledge of all HL7 domain models.

The elements that characterize the HL7 metamodel are described separately in the document offered by HL7 International to standardize such metamodel [14]. The most important parts of each of the four diagrams applied to the present study are indicated below:

- Information Model diagram. Defines the content of the messages that can be exchanged through HL7. The main elements of this information model are classes, connections, attributes, and states. Classes provide objects abstractions; semantic relationships between classes are expressed through connections; attributes represent the facts that affect the class objects; and states capture the changes that events caused on classes.
- Data Types and Vocabulary Domains diagram. Defines the structures and vocabulary domain relationships used to outline the information model attributes and message design models.
- Use Cases and Interaction diagram. Defines the elements necessary to design use cases and interaction diagrams.
- Messages Design diagram. Defines message specification model maps, and information content models, and are used to cover message particular specifications. This diagram includes the message information model, which is an essential information model subset used to define a message set, a message hierarchical description, and a message element model type.

3.2. The UML metamodel

Within the OMG (Object Management Group) proposals that cover MDE, UML is the most prevalent conceptual modeling language. Conceptual modeling languages allow defining the entities and relationships presented by an information system. This way, they allow designing the software system at the Design phase of the software lifecycle. UML helps make specifications and create documentation of the software system domain models from the structure and design perspective.

UML models are simply UML metamodel instances, and consequently, such metamodel represents the general knowledge of all models defined with UML.

In addition, the UML metamodel is described in two documents: infrastructure [15] and superstructure [16]. A description of all the elements of the UML metamodel can be read in the documents of infrastructure and superstructure included in the OMG official website.

Some of the parts of this metamodel that are considered relevant for the present research are highlighted below in order to study their correspondence with the HL7 metamodel:

- The Types diagram within the Core::Basic chapter of the UML infrastructure (described in clause 10.1) considers the definition of elements, comments, and different types of elements that can be found.
- The Classes diagram within the Core::Basic chapter of the UML infrastructure document (described in clause 10.2) refers to the definition of classes and attributes.
- The Expressions diagram within the Core::Constructs chapter of the UML infrastructure document (described in clause 11.2) involves the definition of the values given to these elements.
- The Classes diagram within the Core::Constructs chapter of the UML infrastructure document (described in clause 11.3) identifies the definition of associations.
- The Generalizations package within the Core::Basic chapter of the UML infrastructure document (described in clause 9.2) analyzes the definition of generalizations.
- The Constraints diagram within the Core::Constructs chapter of the UML infrastructure document (described in clause 11.5) studies the definition of constraints.
- The Profiles package within the Core::Profiles chapter of the UML infrastructure document (described in clause 12.1) outlines the use of extensibility mechanisms to UML, including the definition of stereotypes.
- The Packages diagram within the Core::Constructs chapter of the UML infrastructure document (described in clause 11.9) copes with the definition of packages.
- The State Machines diagram in the BehaviorStateMachines package within the State Machines chapter of the UML superstructure document (described in clause 15) represents the definition of states.
- The UseCases diagram in the Use Cases chapter of the UML superstructure document (described in clause 16) represents the definition of use cases, actors, and relationships.
- The Interactions diagram in the BasicInteractions package within the Interactions chapter of the UML superstructure document (described in clause 14) covers the definition of interactions.

3.3. Comparative analysis between both metamodels

Currently, we are working on reaching a solution in the MDE context by evaluating the possibility of associating both the UML and HL7 metamodels. Focused on this goal, we have first conceptually analyzed the UML and HL7 metamodels; second, we have theoretically studied the correspondence among their elements. Table 1 lists the result of this analysis.

As expected, there are many elements in the HL7 metamodel that we could not identify with UML. This is mainly because UML is a general standard, whereas HL7 is a specific standard (for

Table 1
Correspondence between both metamodels.

HL7 metamodel diagram	HL7 metamodel element	UML metamodel element
Information model	Class	Class
	Attribute	Property
	Structural_attribute	*Direct correspondence not found*
	Association	Association
	Composite_aggregation	*Direct correspondence not found*
	Generalization_relationship	Generalization
	State	State
Data types and vocabulary domain	Data_type	Type
	Data_type_category	*Direct correspondence not found*
	Attribute_type	TypedElement
	Vocabulary_concept	*Direct correspondence not found*
	Domain_version	*Direct correspondence not found*
	Code_system	*Direct correspondence not found*
	Coded_term	*Direct correspondence not found*
	Concept_relationship	*Direct correspondence not found*
Use cases and interaction	Actor	Actor
	Use_case	UseCase
	Use_case_relationship	DirectedRelationship
	Storyboard	*Direct correspondence not found*
	Interaction	Interaction
	Application_role	*Direct correspondence not found*
Messages design	Design_information_model	*Direct correspondence not found*
	DIM_class_row	*Direct correspondence not found*
	DIM_attribute_row	*Direct correspondence not found*
	DIM_relationship_row	*Direct correspondence not found*
	DIM_state_row	*Direct correspondence not found*
	Hierarchical_message_description	*Direct correspondence not found*
	HMD_class_row	*Direct correspondence not found*
	HMD_attribute_row	*Direct correspondence not found*
	HMD_relationship_row	*Direct correspondence not found*
	Message_type	*Direct correspondence not found*

healthcare) that defines specific elements for needs associated with the health area.

Then, it is necessary to find out a solution that uses all the HL7 metamodel conceptual richness in our MDE solution.

3.4. Stereotypes as solution to cover all HL7 richness

We propose using the extensibility mechanisms of UML in order to define a new UML-based language that models those aspects not previously covered by UML. This way, the notation and semantics of this language can be broadened.

Extending UML (i.e., specializing its concepts and establishing semantic constraints on them, but always preserving the UML elements' original semantics) is known as UML Profile and allows defining a collection of UML extensions that together describe some particular modeling problems, and facilitate modeling constructions in that domain. This type of extensibility mechanisms is defined in clause 12.1 in the UML infrastructure document [15].

UML extension mechanisms are based on: (i) stereotype, which defines the elements of a specific domain to extend UML classes from them; (ii) tagged value, which adds additional meta-attributes to any stereotyped element defined in the Profile; and (iii) constraint, which identifies the conditions of the stereotyped elements to create well-defined models and extrapolate constraints to the profile.

In this paper, we define an UML Profile using previous extension mechanisms to model HL7 domains. Our UML Profile is not explained in detail because this paper would become too extensive, but we describe a particular case as an example. In particular, we describe how the HL7 classes, attributes, and structural_attribute metaclasses are defined in our UML Profile. The first metaclass is an object abstraction of a set of real-world things, considering that all objects must have the same characteristics, and all instances must conform to the same rules. The second metaclass (i.e., the «HL7Attribute» metaclass) represents data captured on classes, and allows taking values that are independent for each HL7 class. The third metaclass constitutes a structured attribute that establishes a link between the class and the coding applied to represent the instances of these classes (set of allowed codes).

We have followed several steps to define these metaclasses in the UML Profile.

On the one hand, it is necessary to create three stereotypes to represent the HL7 class, attribute, and structural_attribute metaclasses. These stereotypes are named «HL7Class», «HL7Attribute», and «StructuralAttribute» in our UML Profile, and they are respectively extended from the UML «Class» metaclass, «Property» metaclass, and «Property» metaclass using the UML extension mechanism, as indicated in Table 1. Although Table 1 does not identify a direct correspondence for utilizing a structural_attribute element (HL7 metamodel) through a UML metamodel element, we have decided to use the UML «Class» metaclass because it has a similar structure.

On the other hand, each attribute defined in the HL7 standard for the HL7 selected metaclasses have been mapped using tagged values in each stereotype.

The «HL7Class» stereotype has four tagged values: (i) description, which is a short informative statement; (ii) history, which is a compound data type that serves to track the history of any HL7 metaclass (e.g., the «HL7Class» and «HL7Attribute» metaclasses); (iii) isAbstract, which indicates whether this class can be instantiated; and (iv) name, which specifies a unique name for the HL7 class.

The «HL7Attribute» stereotype has seven tagged values: (i) description, which covers a short informative description of the attribute; (ii) history, which is a compound data type that serves to track the history of any HL7 metaclass (e.g., the «HL7Class» and «HL7Attribute» metaclasses); (iii) inclusion, which indicates

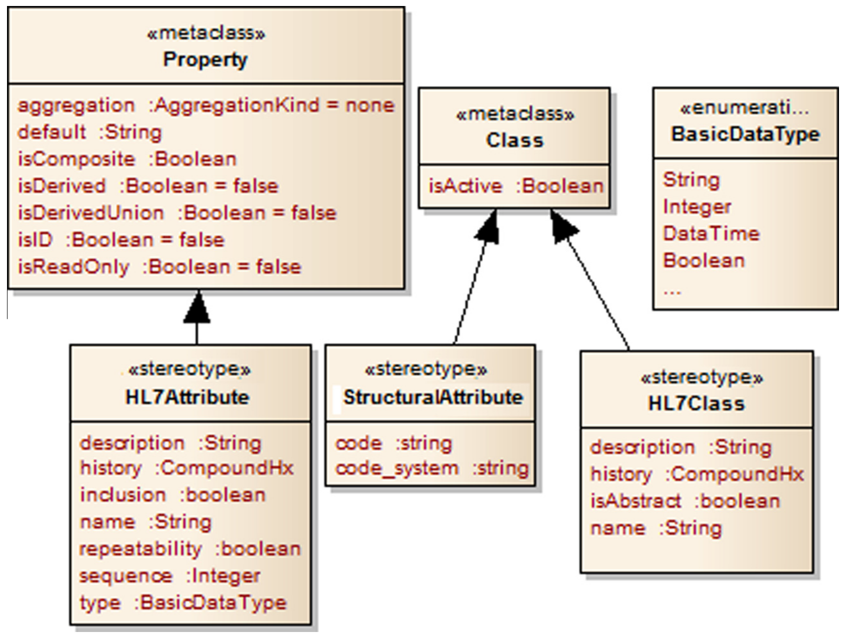

Fig. 2. Fragment of our UML Profile related to the «HL7Class», «HL7Attribute», and «StructuralAttribute» stereotypes.

whether inclusion of the attribute is mandatory; (iv) name, which specifies a unique name; (v) repeatability, which covers the repeat characteristics when this attribute is included in a hierarchical message description; (vi) sequence, which identifies the relative sort order of the attribute within the containing class; and (vii) type, which specifies the data type of the attribute (the data types allowed are formalized through the «BasicDataType» enumerations on our profile).

The «StructuralAttribute» stereotype has two tagged values defined according to the HL7 standard: (i) code_system, which means the code system that defines the set of allowed codes used in the structured attribute; and (ii) code, which means the specific code assigned to the attribute in an instantiation.

Fig. 2 shows how the «HL7Class», «HL7Attribute», and «StructuralAttribute» stereotypes extend respectively from the UML «Class» and «Property» metaclasses in our UML Profile.

3.5. Transformation rules

After laying the theoretical foundations in the previous section, we use the transformation techniques in the MDE context in order to establish a systematic protocol with which to obtain the HL7 model from a specific UML model. For this purpose, we have defined a set of model-to-model transformations (M2M) rules that are formalized using QVT (Query/View/Transformation [17]). All our QVT rules are not explained here because they are outside the scope of this paper, and would make the paper too extensive. Nevertheless, as an illustrative example, Table 2 describes the QVT rule used to obtain the «HL7Class» metaclass (HL7 metamodel) from the «Class» metaclass (UML metamodel). This QVT rule is formalized at the «Class» metaclass using the QVT directive «mapping» that allows mapping concepts from the source metamodel (UML) to the target metamodel (HL7). In addition, such rule initializes all properties of the «HL7Class» metaclass and resolves all relationships between this metaclass and the «HL7Attribute» and «HL7Relationship» metaclasses (the latter has not been mentioned before; here, we represent a relationship between HL7 classes).

Table 2
QVT rule to obtain the «HL7Class» metaclass.

```
mapping UMLMetamodel::Class::toHL7Class (): HL7Metamodel::HL7Class {
  description:= "DESCRIPTION";
isAbstract:= self.isAbstract;
name:= self.name;
  history:=
  has += self.Property→forAll (p: Property | p.resolveone(HL7Metamodel::
   HL7Attribute);
  relationship += self.Property → forAll (a: Association |
a.resolveone(HL7Metamodel::HL7Relationship);
}
```

3.6. Tool developed on Enterprise Architect

One of the most important aspects to ensure the applicability of our model-driven proposal in real environments is to design and develop tools that support our theoretical framework. In this sense, these tools should allow defining HL7 models (using our UML Profile described in Section 3.4) in a friendly way, and allow automatic execution of the transformation rules.

In this context, we have chosen to integrate our proposal within the EA (Enterprise Architect) tool in order to provide a real environment that covers the proposal in a practical context. However, this decision has not been an easy task. We have considered several market studies. One [18] is the market study by our research group (IWT2, Web Engineering and Early Testing of University of Seville, Spain) in collaboration with the Andalusian Regional Ministry of Culture, Education and Sport (Spain).

This study compares nine modeling tools in order to identify the best tool to perform our research with regard to technology transfer issues. This comparative study is conducted considering the following points: (i) to provide UML2.0 extension mechanisms (i.e., a tool must allow developing and designing UML Profiles); (ii) to incorporate MDE processes (i.e., a tool must support mechanisms or algorithms in order to systematically generate models from other models); (iii) to allow generating documentation in an automatic, flexible, and agile manner; (iv) to allow managing big project lifecycles; (v) to be useful for the entire software lifecycle (i.e., the phases that cover the feasibility study phase, requirements, analysis, design, implementation, testing, and finally, maintenance); and (vi) to be compatible with UML2.0.

In the end, we conclude that EA offers the best quality–cost relationship in the evaluated points. Another reason that led us toward using this tool instead of another is the fact that EA is used widely in a high number of IWT2 projects, and it is known by most of its customers and partners. Some examples include the AQUAWS project [19] performed with EMASESA, CALIPSOneo project [20] performed with Airbus Military, THOT project [21] performed with Public Works Agency of Junta of Andalusia, and SICATA project performed with Andalusian Health Service, among others.

All these reasons made us finally choose EA as our base modeling tool, but it is important to emphasize that our solution can be implemented by any tool that manages UML2.0 and extension mechanisms if the tool provides import and export options structured in the XMI format.

3.7. Our solution

After choosing our modeling tool base, we integrated our solution on it. In this sense, we implemented our UML Profile through EA and each QVT transformation rule through a plug-in in EA.

Fig. 3 shows the graphical user interface for our solution within EA, and how several HL7 classes were modeled (Section 4 describes this example in detail). The (A) mark shows the EA Toolbox associated with our UML Profile; and the (B) mark shows the modeling area where users can represent their HL7 models.

4. Results

We present a first approach to the HL7 MDE solution without discarding the metamodel proposed by HL7 International, through a plug-in developed with the EA tool.

This research is currently addressed as the subject of a PhD thesis. Nevertheless, the present work is based on previous practices, such as the Diraya Specialized Attention project previously mentioned, which constitutes the first previous experience on this research topic.

The short-term objective considers designing a solution in the MDE context in order to connect both the UML and HL7 metamodels, but focuses on the requirements and analysis levels.

We would also like to test these results in the NDT methodology context. This methodology, basically developed in the software project context, has been applied in biosanitary fields related to HL7 and offers a suitable framework on which to center the practical assessment of our results. In addition, we execute a proof of concept in this line in the OncoInves project context (code PI-0116-2012) by designing its scenario based on the EHR functional model, the clinical research profile for HL7 [22], and therefore, follow the HL7 metamodel underlying any model.

In order to show that a correspondence between the HL7 and UML metamodels is feasible, this section describes an example based on the Care Plan reference model. The HL7 Care Plan defines the management action plans for various problems identified for patients. In this structure, the care planning for each professional can be organized, planned, and checked for completion, thus allowing the monitoring of unperformed activities and unmet goals for later follow up. Within the Care Plan Model, Plan and «HealthGoal» classes have been selected and extracted. The «Plan» HL7 class represents a generic care plan that contains all relevant components to support different types of care plans. The «HealthGoal» HL7 class represents the goals that include a specific Plan. Fig. 4 shows the relationship between these HL7 classes according to HL7 specification.

The above classes can be represented using our UML Profile. However, we did not model all classes in order to not overextend this paper. The undefined classes are secondary classes in our example, and are colored in dark in order to identify them.

Both the «Plan» and «HealthGoal» HL7 classes are represented by the «HL7Class» stereotype. These HL7 classes have three types of attributes: (i) basic attributes, which represent simple values such as logical values, strings, and integers; (ii) list of values, which represents a fixed list of allowed values; (iii) and structured attributes, which represent a data type composed of other attributes. As Table 1 indicates, the first and second attribute types have a

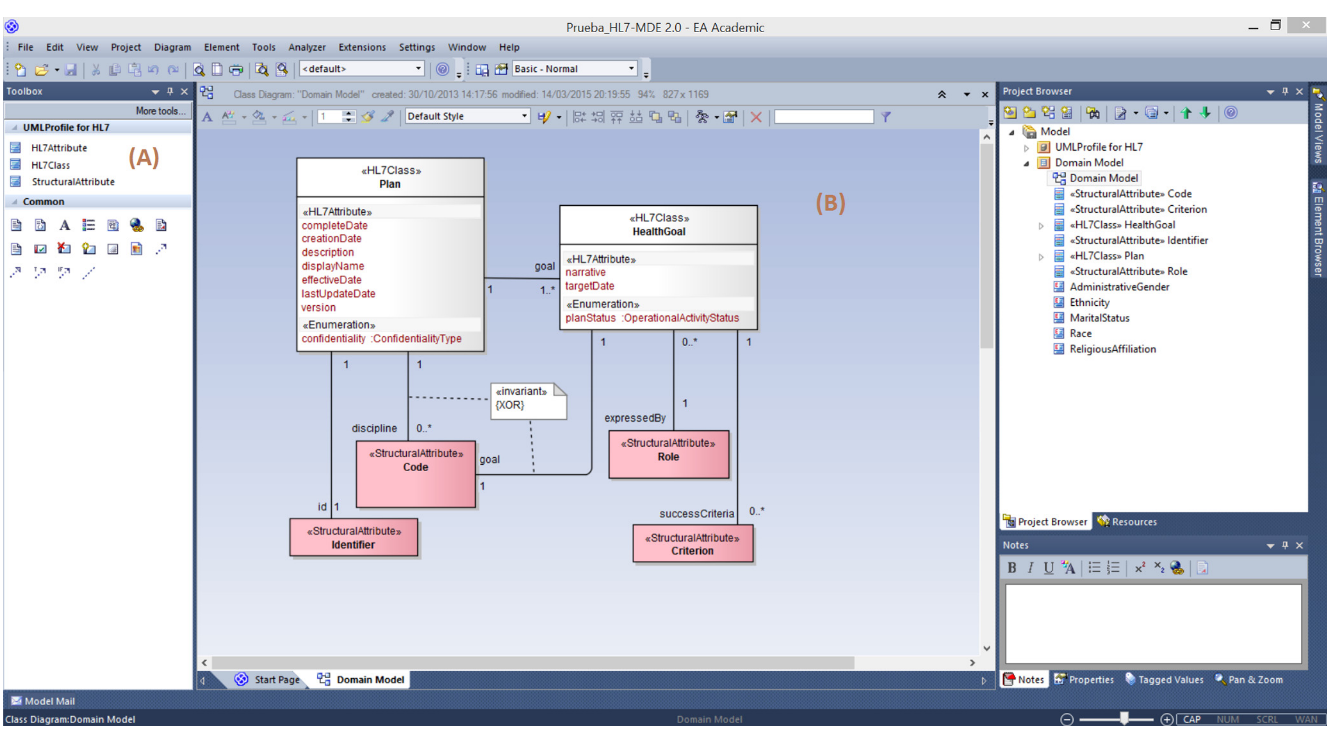

Fig. 3. View of our solution integrated into EA.

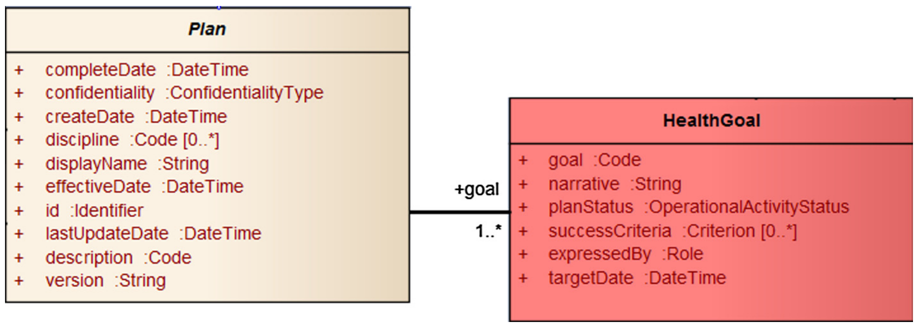

Fig. 4. «Plan» and «HealthGoal» classes extracted from the Care Plan reference model.

direct correspondence with the UML Property and Enumeration metaclasses, respectively. For example, the «Plan» HL7 class has the confidentiality attribute defined as a value from the list of values named ConfidentialityType.

The third attribute type does not have direct correspondence with UML. Section 3.4 demonstrates our solution based on extending the «Class» metaclass of UML and using the «StructuralAttribute» stereotype (Fig. 2). For example, the «Plan» HL7 class has an attribute named discipline that is a set of Code objects. The Code concept is an HL7 object itself with its own attributes. To model this situation in UML, we define the Code concept (with its own attributes) using the «StructuralAttribute» stereotype. In addition, we establish a relationship between the Code and Plan HL7 concepts. In this relationship, we indicate that Plan HL7 can have a set $[0 \ldots *]$ of Codes.

Finally, Fig. 5 shows how the «Plan» and «HealthGoal» HL7 classes are represented using our UML Profile.

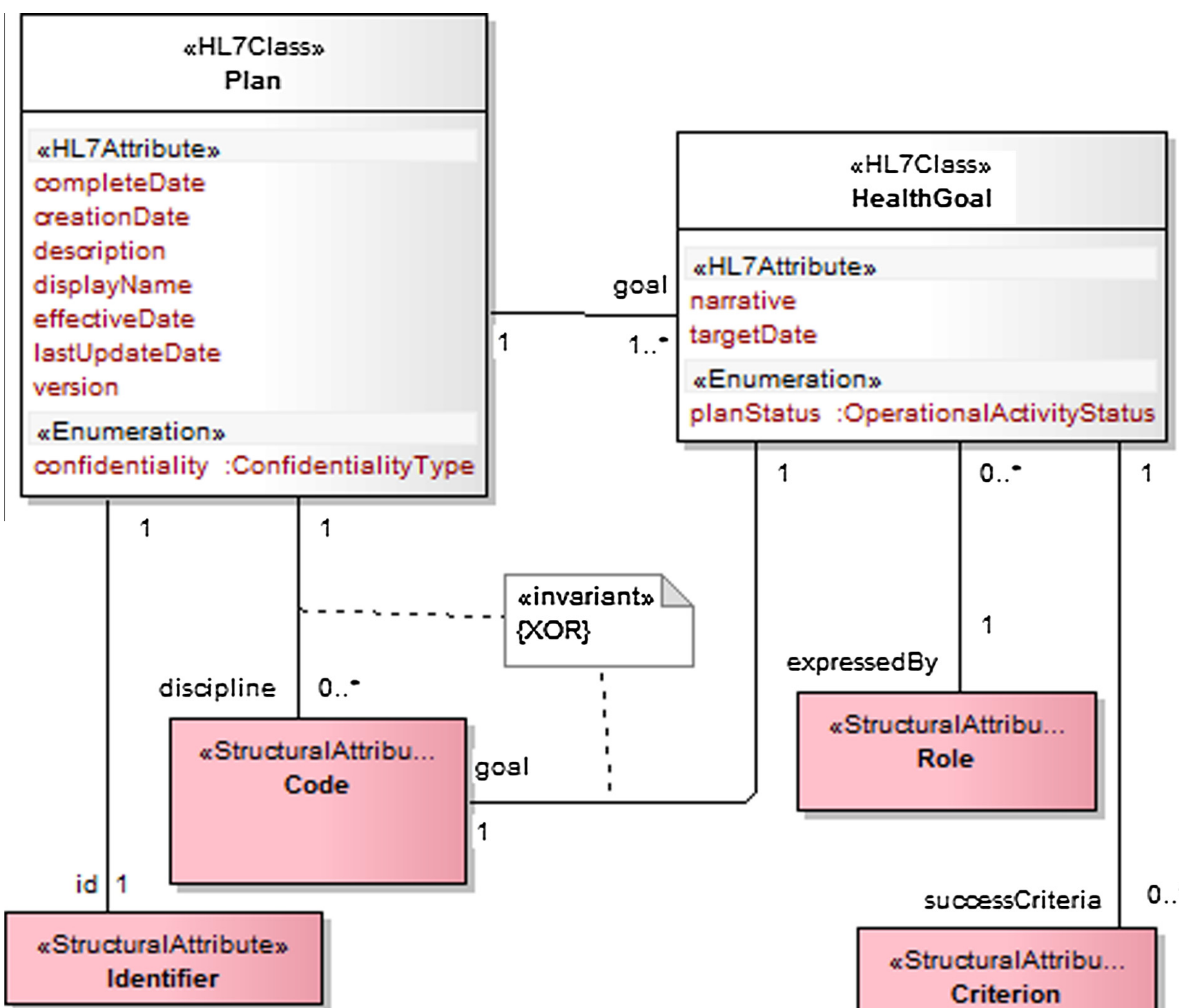

Fig. 5. UML-based representations of «Plan» and «HealthGoal» HL7 classes.

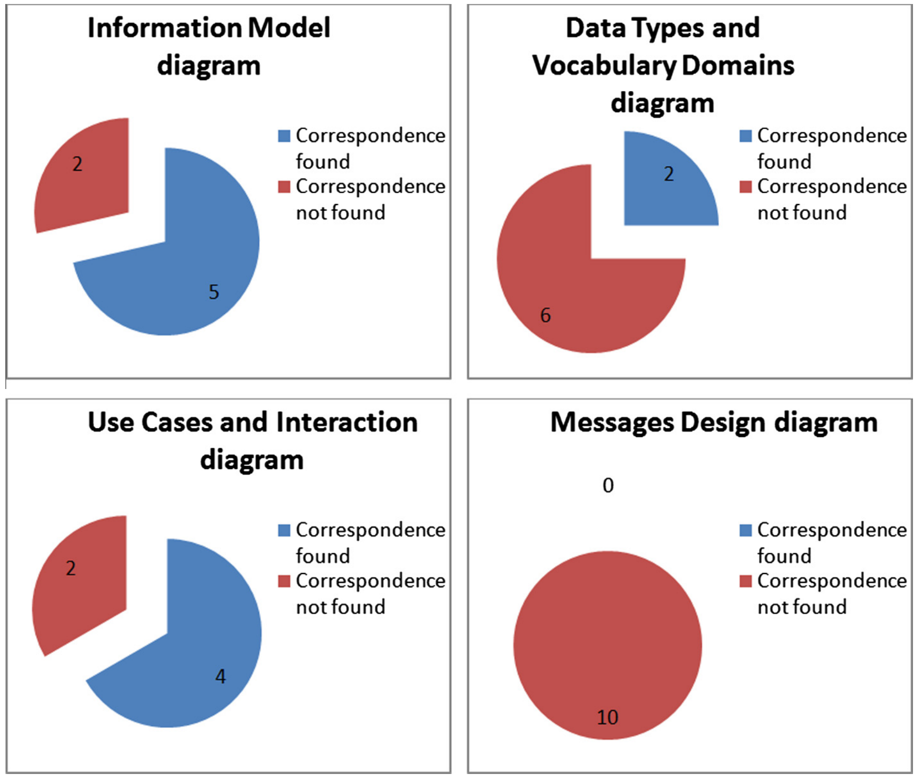

Fig. 6. Correspondence between HL7 and UML metamodels.

5. Discussion

As presented in Section 3.1, the HL7 metamodel is composed of four diagrams. Fig. 6 shows the relationships among the concepts identified (or not) in the UML and HL7 metamodels by classifying these relationships in terms of HL7 metamodel diagrams using the information included in Table 1.

Globally, approximately 35% could be identified, against 65% that could not be identified.

Despite the fact that it obvious that all the elements that correspond to the Messages Design diagram of the HL7 metamodel cannot be represented with UML elements, this is not understood as a problem because these elements are necessary for designing a system, and the present research is focused on the Requirements and Analysis phases exclusively. Discarding the part of the HL7 metamodel that covers the Messages Design Diagram for this reason, we obtain the following percentages: approximately 52% could be identified, against 48% that could not be identified.

As a solution for using all the HL7 metamodel conceptual richness in our MDE solution, we employ stereotypes, as explained in depth in Section 3.4. With our solution, we expect to fully cover the HL7 metamodel (with the exception of the Messages Design diagram). Therefore, using our solution, we can design HL7 models taking advantage of UML commercial tools.

Using model transformation techniques in the MDE context, we can ensure that the results of these transformations are consistent with the source models. This fact will reduce effort and errors. When an application implements an HL7 standard, this implementation is conditioned to the interpretation applied by the technical staff. Using our proposal, if different HL7 applications are modeled with our tool, although this different HL7 applications use a different subset of the HL7 standards, will be more compatible and their interoperability will be increased.

HL7 International involves many standards' development lines. Many researchers, as part of the scientific community, are working with HL7 standards in the field of healthcare information systems [23–26].

The approach explained in this paper affects all MIF-based artifacts from HL7 International. Several HL7 International subgroups exist that are conducting the work that covers a domain model based on MIF utilization, and therefore all these domain models are based on the same metamodel. The HL7 domain models provide a standard interoperability framework that allows an unambiguous information exchange, thus providing maintenance and extension abilities in order to achieve a common interoperability framework as a driver for the semantic interoperability among health information systems. When developing a new HL7-based interoperability use case, software engineers face the generation of a new domain model according to HL7 specifications. Based on MDE and relying on the HL7 metamodel common to all domain models, our proposal provides software engineers with a tool that allows them to generate these domain models through UML-based modeling, thus being transparently compliant to HL7 specifications while shortening development time, avoiding errors, etc.

HL7 International is working on introducing a new proprietary representation in order to reduce the complexity of HL7 reference models in the context of the Fast Healthcare Interoperability Resources specification. This issue highlights our reflection on HL7 complexity: we are walking in the same direction in order to design HL7 systems in an easier way.

We find it interesting to lead our work toward the working lines listed below, once the correspondence between UML and HL7 metamodels has been studied and the transformation between models has been implemented:

- To design software system models through the UML standard in order to obtain HL7 correspondence automatically. The learning curve for software engineers who would like to design these systems in accordance with HL7 would be shorter if they were they to model directly in UML. Most software engineers know UML because it is the most commonly used conceptual model language.

- To use UML-based tools. There are several UML-based tools, both in open-source and private markets, that allow designing a system with UML. In addition, there are several tools that can use UML diagrams that allow software engineers to work easier, such as NDT.
- To align, in a simple way, the concepts used in HL7 models with a system of concepts in a health scenario: ISO 13940. Studying the correspondence among concepts used in the HL7 metamodel is the only requisite to align all HL7 models with a standard system of concepts because all HL7 models are based on the same metamodel.
- To spread this work through the analysis of the conformance between metamodel UML and other information domain formats, even allowing the ontologization of HL7 domain models.
- To connect our solution to the emerging ADL (Archetype Definition Language) specification, the language in which openEHR archetypes are expressed. This proposal from the openEHR Foundation is an abstract syntax with the features of being human readable and computer processable (i.e., it can be edited manually using a normal text editor). To cover this connection between our solution and ADL, we consider using AML (Archetype Modeling Language) from OMG, a UML profile for modeling archetypes that cover the transformation of AML models to an instance of ADL.
- To certify the compliance of HL7 domain models with UML information system design models. A tool that reports errors found will be created once UML models have been automatically transformed into HL7 models. Thanks to M2M, we will be able to validate models because we will specify the metamodel with which the source model must comply. It may be interesting for HL7 International to have a tool that can validate a UML requirements model that should include the system requirements for the HL7 functional model in order to initiate a validation process of the existing systems.

Moreover, our research must confront a challenge: We expect for software engineers to plan systems with UML, and align them automatically with the HL7 metamodel. Nevertheless, this raises the following question: What would happen if we wanted them to be aligned with a specific HL7 model, for example, with the Reference Information Model (RIM)? We will have to find a solution to this challenge, and perhaps we may obtain it by aligning them with HDF (HL7 Development Framework).

6. Conclusions

This paper presented research with the aim of using the HL7 metamodel in an MDE context.

UML is a generic conceptual modeling language that intends to support any scenario in the domain model design. Thus, its metamodel is very generic.

In contrast, HL7 International is an organization that defines and promotes standards focused on healthcare domains. Its metamodel is less generic than the UML metamodel because it is restricted to the healthcare area.

HL7 standards propose very remarkable domain models from a conceptual perspective, but they are very complex at the same time. For this reason, software engineers who attempt to design systems that follow these standards may find some difficulties when managing them, and encounter a long learning curve.

Using HL7 standards in the MDE context remains an unexplored area, from which many benefits and research areas can be obtained. This will provide software engineers who attempt to design Health information systems with solid support.

This work uncovered a problem when searching a solution to the use of HL7 in the MDE context: there are elements in the HL7 metamodel that do not correspond to elements of the UML metamodel. An experiment was performed using a UML metamodel extension through stereotypes to solve this problem, and it was concluded that it is possible to completely cope with the elements present in the HL7 metamodel.

The authors will continue working on this research in order to find a solution in the MDE context that automatically links the HL7 metamodel to the UML metamodel, mainly focusing on the Requirements and Analysis levels. To cover all the richness that the HL7 metamodel represents, a systematic method is being used by defining stereotypes and transformation rules class by class through this metamodel.

Conflict of interest

The authors have no conflict of interest.

Acknowledgments

This research has been partially supported by the MeGUS project (code TIN2013-46928-C3-3-R) of the Spanish Ministry of Science and Innovation, and the Carlos III Health Institute within the call Strategic Help in Health (PITeS ISA project, code PI12/01571).

The authors are thankful to the Network for Innovation in Medical Technologies and Health ("Plataforma ITEMAS" in Spanish, code PT13/0006/0036) promoted by the Carlos III Health Institute. Finally, we would also like to thank Universia foundation for supporting PhD students through research grants.

References

[1] HL7 International Official Web Page, 2015. <http://www.hl7.org/>.

[2] D.C. Schmidt, Model-driven engineering, IEEE Comput. 39 (2) (2006) 25, http://dx.doi.org/10.1109/MC.2006.58.

[3] R. Van Der Straeten, T. Mens, S. Van Baelen, Challenges in model-driven software engineering, Models Softw. Eng. (2009) 35–47, http://dx.doi.org/10.1007/978-3-642-01648-6_4.

[4] R. Spronk, C. Ringholm, The HL7 MIF – Model Interchange Format, 2010. <http://www.ringholm.com/docs/03060_en_HL7_MIF.htm>.

[5] M.J. Escalona, C.L. Parra, J. Nieto, Diraya project. The power of metamodels in real experiences with web engineering, in: XI International Congress on Project Engineering (AEIPRO), 2007, Proceedings, p. 167.

[6] M.J. Escalona, C.L. Parra, F.M. Martín, J. Nieto, A. Llergó, F. Pérez, A practical example from model-driven web requirements. Information system development, Chall. Pract. Theory Educ. 1 (2008) 157–168.

[7] A. Martínez-García, M.J. Escalona, J.A. García-García, G. Aragón, A. Moreno-Conde, C. Parra, Using model-driven engineering and health standards for improving healthcare processes management, in: 26th Medical Informatics Europe Conference, MIE2015, Madrid, España, 2015.

[8] M.J. Escalona, G. Aragón, NDT a model-driven approach for web requirements, IEEE Trans. Softw. Eng. 34 (2008) 377–390, http://dx.doi.org/10.1109/TSE.2008.27.

[9] F. Oemig, B. Blobel, A formal analysis of HL7 version 2.x, Stud. Health Technol. Inform. 169 (2011) 704–708, http://dx.doi.org/10.3233/978-1-60750-806-9-704.

[10] B. Bánfai, B. Ulrich, Z. Török, R. Natarajan, T. Ireland, Implementing an HL7 version 3 modeling tool from an Ecore model, Stud. Health Technol. Inform. 150 (2009) 157–161, http://dx.doi.org/10.3233/978-1-60750-044-5-157.

[11] D. Ortiz, A. Villegas, M.R. Sancho, A. Olive, J. Vilalta, Automatic Transformation of HL7 v3 Information Models into Equivalent UML Models, HL7 Spain Technical Report, 2011.

[12] S. Renly, R. Altamore, L. Nelson, A. Orlova, K. Patterson, S. Quaynor, L. Reed-Fourquet, J. Timm, A new model for collaboration: building CDA documents in MDHT, in: AMIA Annual Symposium Proceedings, American Medical Informatics Association, 2012, pp. 763–769.

[13] A. Farkash, J.T. Timm, Z. Waks, A model-driven approach to clinical practice guidelines representation and evaluation using standards, Stud. Health Technol. Inform. 192 (2012) 200–204.

[14] HL7 International, HL7_V3_Meta-Model, Version Number 1-16, 2002. <http://www.hl7.org/library/committees/mnm/hdf/met0116l.pdf>.

[15] UML, UML Infrastructure, 2011. <http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF>.
[16] UML, UML Superstructure, 2011. <http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/>.
[17] Object Management Group, Documents Associated with Meta Object Facility (MOF) 2.0 Query/View/Transformation, 2008. <http://www.omg.org/spec/QVT/1.0/>.
[18] IWT2 Research Group, 2014. <http://www.iwt2.org/>.
[19] C.R. Cutilla, J.A. Garcia-Garcia, M. Alba, M.J. Escalona, J. Ponce, L. Rodriguez, Model-driven engineering applied in functional testing: the practical experience of the AQUA-WS project, in: Information Systems and Technologies (CISTI), 6th Iberian Conference, 2011, IEEE, pp. 1–5.
[20] M.J. Escalona, J.A. Garcia-Garcia, F. Mas, M. Oliva, C. Del Valle, Applying model-driven paradigm: CALIPSOneo experience, Proc. Ind. Track Conf. Adv. Inform. Syst. Eng. 1017 (2013) 25–32.
[21] M.J. Escalona, J.A. García-García, F.J. Dominguez-Mayo, I. Ramos, Technical tool surveys and comparative studies: a systematical approach, in: V. Strahonja, N. Vrček, D. PlantakVukovac, C. Barry, M. Lang, H. Linger, C. Schneider (Eds.), Information Systems Development: Transforming Organisations and Society through Information Systems (ISD2014 Proceedings), Faculty of Organization and Informatics, Varaždin, Croatia, 2014. ISBN: 978-953-6071-43-2.
[22] HL7 International, HL7 EHR Clinical Research Functional Profile (CRFP), Release 1, 2014. <http://www.hl7.org/implement/standards/product_brief.cfm?product_id=16>.
[23] C.H. Lin, I.C. Lin, J.S. Roan, J.S. Yeh, Critical factors influencing hospitals' adoption of HL7 version 2 standards: an empirical investigation, J. Med. Syst. 36 (3) (2012) 1183–1192, http://dx.doi.org/10.1007/s10916-010-9580-2.
[24] T. Al-Enazi, S. El-Masri, HL7 engine module for healthcare information systems, J. Med. Syst. 37 (6) (2013) 1–8, http://dx.doi.org/10.1007/s10916-013-9986-8.
[25] G.B. Laleci, M. Yuksel, A. Dogac, Providing semantic interoperability between clinical care and clinical research domains, IEEE J. Biomed. Health Inform. 17 (2) (2013) 356–369, http://dx.doi.org/10.1109/TITB.2012.2219552.
[26] W. Kim, S. Lim, J. Ahn, J. Nah, N. Kim, Integration of IEEE 1451 and HL7 exchanging information for patients' sensor data, J. Med. Syst. 34 (6) (2010) 1033–1041, http://dx.doi.org/10.1007/s10916-009-9322-5.