Towards a Semantic Interoperability Environment

Isela Macía
GE Global Research
Rio de Janeiro, Brazil
bertran@ge.com

***Abstract*—The variety of Electronic Health Records (EHRs) makes interoperability a global trend in healthcare, which is of a paramount interest to the Brazilian Ministry of Health. In particular, semantic interoperability receives a special attention since it ensures that different health information systems make the same interpretation of the exchanged information. Although several standards have been documented to support interoperability (e.g. HL7 and IHE), achieving the semantic one is still a challenge. In this context, this paper represents a step towards supporting seamless semantic interoperability by combining different health standards (OpenEHR, IHE and HL7). It describes a software architecture that illustrates the role of different health standards in a semantic interoperability environment. Moreover, it introduces a process aiming at supporting the sematic validation of clinical documents. Finally, it documents several findings, such as benefits of the combined use of OpenEHR and IHE profiles.**

Keywords—Interoperability health informatics standards; OpenEHR; IHE; HL7; software architecture.

I. INTRODUCTION

Electronic Health Records (EHRs) are essential or at the heart of the IT application in healthcare. They manage all of a patient's medical history from one practice, supporting providers on both diagnosis and treatment processes. The variety and relevance of EHRs resulted in the emergence of methodologies to develop these health records such as OpenEHR [15], making interoperability a global trend in healthcare. Standards to represent clinical information as well as to support interoperability are of a paramount interest to the Brazilian Ministry of Health - e.g., Resolution No 2073 [12].

Interoperability refers to the ability of health information systems (e.g. EHRs) to exchange information and to use the information that has been exchanged [6]. At the semantic level, interoperability implies the concern to ensure the same interpretation of the exchanged information by different health information systems. Interoperability is crucial to the Brazilian government given its well-known benefits [6], such as: (i) coordination and continuity of care, (ii) support and continuity of the clinical, administrative, educational and research used by organizations, (iii) increase effectiveness attention, (iv) decrease of inefficiencies and inequities, and (v) improve of patients' health and care quality.

Exchanging information in a standard way is crucial to achieve interoperability among health information systems. The advent of different health standards, such as *Health Level Seven* (HL7) enables health information systems to integrate by communicating standard information [13]. Other health standards, such as the *Integrating the Healthcare Enterprise* (IHE) defines integration guidelines based on established data standards (e.g., HL7) [14]. Its main goal is to integrate health standards for effective interoperability and efficient workflow.

However, the use of existing standards to achieve semantic interoperability is still a challenge in healthcare due to several reasons. First, interoperability standards [13][14] provide limited mechanisms to validate the information to be exchanged (e.g. constraints of clinical concepts). Second, they do not focus on ensuring the same interpretation of the exchanged information from one health information system to another. Third, the adoption of health interoperability standards is not trivial since it requires high effort, technical expertise as well as clinical domain knowledge. Finally, the combined use of standards to achieve semantic interoperability is a field of research [3][5][9].

This paper represents a step towards supporting seamless semantic interoperability among health information systems (e.g. EHRs), by adopting OpenEHR, HL7 and IHE in the same environment. In particular, it describes a software architecture that: (i) defines the roles of different health informatics standards in a semantic interoperability environment and (ii) shows how they can co-exist. Second, a process is presented aiming at supporting the sematic validation of clinical documents and therefore, at ensuring their same interpretation by different health information systems. It is important to highlight that existing interoperability standards (e.g. IHE and HL7) do not support this kind of semantic validation. The described process is generic, allowing health professionals to specify the validation of clinical documents.

This paper is organized as follows. Section II introduces the health informatics standards used in this research to achieve semantic interoperability. Section III describes the proposed software architecture in terms of its main components. Section IV details each step of a process to support the semantic validation of clinical documents. Section V discusses lessons learned when using different interoperability standards. Finally, Section VI concludes and presents future work.

II. BACKGROUND

The focus of this research is on supporting semantic interoperability, as different health information systems should interpret clinical information in the same manner. In this context, this section briefly describes the health informatics standards leveraged to that end.

978-1-4799-6644-8/14/$31.00 ©2014 IEEE

A. OpenEHR

OpenEHR is one of the emerging methodologies for developing EHRs [15]. It consists of a generic information reference model, application-specific archetypes [1] and context-specific templates. The reference model defines generic data structures for the health care domain (e.g. observation, evaluation and history). The archetypes [1] allow the definition of clinical concepts (e.g. blood pressure), which are expressed as views on the generic reference model. Archetypes also define semantic rules that guarantee the correctness and completeness interpretation of the clinical concept (e.g. blood pressure constraints). Finally, templates package the archetypes in terms of forms (or dialogues) relevant to particular situations of care. In this paper, we use OpenEHR (e.g. model and archetypes) to validate clinical documents shared among different EHRs (Section V).

B. Health Level Seven – HL7 Messages

Health Level Seven (HL7) is a not-for-profit organization focused on the development of informatics interoperability standards in the health care domain [13]. These standards aim at supporting the exchange, integration, sharing and retrieval of electronic health information. HL7 messaging standards define the language, and data structure required for seamless information integration among health information systems. HL7 messages (versions 2.1-2.6) use ASCII, non-XML encoding syntax based on segments (lines) and one-character delimiters. Segments are composed by fields separated by a delimiter (e.g. |, ~). Each segment contains one specific category of information. For instance, in every message the first segment defines the message type and, thus, the remainder expected segments in the message. This research leverages HL7 messages to establish the communication among health information systems (Section III).

C. Integrating Health Enterprise - IHE

The Integrating the Healthcare Enterprise (IHE) [14] is an initiative by healthcare professionals and industry to improve the way computer systems in healthcare share information. IHE promotes the coordinated use of established standards such as HL7 to address specific clinical needs in support of optimal patient care. Systems developed in accordance with IHE communicate with one another better, are easier to implement, and enable care providers to use information more effectively. In order to enable seamless communication among international standards, IHE proposes Integration Profiles. These profiles provide precise definitions about the implementation of health standards to meet specific clinical needs (e.g. IHE Infrastructure Technical Framework). In the context of this research, we use three specific IHE profiles to implement the basis for an interoperability environment – i.e. *Patient Identifier Cross-Referencing* (PIX), *Patient Demographics Query* (PDQ) and *Cross-Enterprise Document Sharing* (XDS). Next section details the use of these profiles in this research.

III. System Architecture

This section describes the software architecture implemented in order to achieve semantic interoperability in health care domain. We use HL7 and IHE profiles to support a heterogeneous communication among health informatics systems. Additionally, we use OpenEHR to represent clinical information and validate clinical documents. The goals of the architecture are to show how: (i) semantic interoperability is achieved by using OpenEHR archetypes and terminology repositories, and (ii) different document standards can co-exist in the same environment.

A. Overall Description of the Proposed Architecture

Figure 1 presents the overall design of the proposed architecture, which relies on the Client-Server style [2]. This style allows multiple health information systems, acting as clients, to be served simultaneously. It promotes flexibility since new health information systems can be easily integrated in the environment. As it can be seem, health information systems (e.g. OpenMRS) send patients' information to a centralized server by means of web services. These services, located at the server side, receive HL7 messages as parameters. These messages in turn encapsulate clinical information associated with a specific operation. For instance, when the OpenMRS system updates a patient's information, it then calls the *PIX*, *PDQ* and *XDS* services to reflect the patient update in the server side. The following sections discuss these layers, their main components and relationships in details.

B. Application Layer

The Application layer is composed by health information systems that share and feed patients' information. Figure 1 illustrates two kinds of these systems, which are part of our current implementation. The first system, *Health Portal*, corresponds to a web interface that supports the registering, searching and recovering of patients' information, directly related to the *Integration* Layer (Section III.C). The second system, *OpenMRS*, is an open source EHR [16]. We extended the implementation of this system in order to support communication with the *Integration* layer. When a patient is registered or his/her information is updated in the *Application* layer, the web services (located in the *Integration* Layer) are called and HL7 messages are exchanged between both layers aiming at synchronizing the patient's information. Other kinds of health information systems can be also integrated with our interoperability environment as long as they successfully call the web services located in the *Integration* layer.

C. Integration Layer

The *Integration* layer comprises the implementation of IHE profiles to support interoperability (Section II.C). We decided to focus on the Patient Identifier Cross-Referencing (PIX), Patient Demographics Query (PDQ) and Cross-Enterprise Document Sharing (XDS) profiles. The reason is they specify basic functionalities to support interoperability among health information systems.

The PIX, PDQ and XDS components implement the transactions defined in the PIX, PDQ and XDS profiles, respectively [13]. In particular, the PIX component manages: (i) patients registration, (ii) requests and responses about a list of patient's identifiers in direct relation with the MPI component (Section III.D), and (iii) updates on patient's

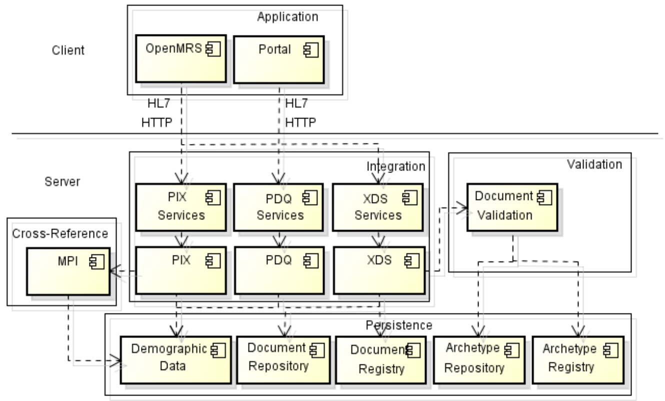

Fig. 1. Component diagram of the proposed architecture.

identifier cross-reference associations. The PDQ component manages requests and responses about patient's demographic and visit information. Finally, the XDS component manages the sharing and searching of clinical documents in relationship with the *Validation* layer (Section III.E).

Moreover, the *Integration* layer comprises the PIX Services, PDQ Services and XDS Services components. They provide web services to access the functionalities of the PIX, PDQ, XDS components, respectively. These web services define the interface and the data contracts (HL7 message) of the access over HTTP protocol. Additionally, they validate the structure of the HL7 messages according to the corresponding operation (e.g. patient registering and clinical episode update). Once a HL7 message is successfully validated, its information is transformed into business entities and the functionalities of the PIX, PDQ, XDS components are invoked. Otherwise, a message is sent back to the source of the request (health information system located in the *Application* layer), indicating the reason of the associated error.

D. Cross-Reference Layer

The *Cross-Reference* layer comprises the Master Patient Index (MPI) component, which ensures patients have a unique identifier and constant demographic across health information systems (*Application* layer). The unique patient identification allows the system to provide a clear and complete view of a patient's clinical history.

In order to ensure such unique patient identification, the component relies on a probability-based-matching algorithm. A probability value is assigned to all sort of demographic data (e.g. birth date, address), which is defined in the system configuration file. This value depends on the relevance and uniqueness of the data to the Brazilian Ministry of Health. For instance, the CPF (individual registration) is a unique serial number for all Brazilians; thus, it receives a high probability. In case of names, probabilities are assigned to phonemes according to their popularity in Brazil. For instance, the "*Mar*" phoneme receives lower probability than others like "*Ken*", since there are more names in Brazil starting with the former than with the later (e.g. Marcio, Marcelo and Marta).

When the PIX component is invoked, it calls the MPI component to ensure the unique patient representation in the interoperable environment. The MPI then verifies whether the patient, P_1, already exists by matching his/her information with those patients' information stored in the database (*Persistence* layer). This matching generates a result value. If this value is higher than a given threshold, then P_1's identifier is associated with P_2's identifier and the information of both patients is merged into the existing P_2 record. Otherwise, if there are no P_1 is stored in the database as a new patient - *Persistence* layer (Section III.F).

E. Validation Layer

The *Validation* layer comprises the *Document Validation* component, which ensures that clinical documents are corrected syntactically and semantically. This validation allows that the health information systems in the interoperable environment interpret the exchanged information in the same way. Note that health interoperability standards (e.g. IHE, HL7) only achieve syntactic validation of clinical documents. They do not support different kinds of semantic validations such as ordering of clinical contents.

The algorithm used in this research to validate of clinical documents emerged as a process result (Section V). The Validation component parses the document and compares its structure against a predefined XSD file (*XML Schema Definition*) in order to ensure its syntactic correctness. The semantic validation goes beyond. It is based on OpenEHR model – i.e. OpenEHR model and archetypes. Once the document is syntactically valid, for each of its structures, the component verifies whether the structure under analysis is in conformance with its corresponding archetype, directly related to the *Persistence* Layer. This semantic validation involves the checking of several types of OpenEHR structures (e.g. Observations, Extracts, History, Events and Elements) against their archetype constraints (e.g. cardinality, order, value), terminologies and dependencies with other archetypes.

F. Persistence Layer

The *Persistence* layer comprises components, which store and recover health information. We decided to store clinical and demographics information in two different PostgreSQL databases for security reasons. This model does not allow knowing the patient's clinical information by only recovering its demographics and vice versa.

Once a clinical document is considered to be valid (semantic and syntactic), its content and metadata (including references to existing archetypes) are persisted in the database. By having the content of clinical document well-organized in a database, the system is able to perform an efficient recovery of the patient's clinical history. Therefore, it is not necessary to mine a huge amount of documents every time the patient's clinical history is visualized. Finally, the *Persistence* layer manages the audit trails of all actions performed (e.g. add and update). This ability makes possible to track who performed a particular action at a given time.

IV. Validation of Clinical Document in Details

The following sections describe our process that supports the validation of clinical documents. This validation benefits clinicians and organizations because they can exchange documents ensuring the preservation of both syntax and semantic of documents. This means that the document sender and the document receiver will interpret the document in the same manner. Additionally, this validation benefits patients since their diagnosis will be conducted more effectively.

A. Validation Process Overview

Figure 2 presents an overview of the validation process, which relies on four main steps. The figure also emphasizes the inputs and outputs specific to each step.

Step 1 – Structure Selection consists of analyzing the structures in the OpenEHR information model used to validate clinical documents at a first stage of the interoperability environment implementation. The set of structures should be incrementally increased until achieve the validation of all the OpenEHR model structures. In addition, this step comprises the analysis of clinical terminologies that will be used to support the validation of clinical documents. The goal is to select the suitable terms to represent the clinical information that will be exchanged in the system. The selection of terminologies is a critical task since any inconsistency in the document may lead to misunderstandings of the clinical information.

Step 2 - Archetype Selection comprises a deep analysis of which clinical concepts will be exchanged in the system through documents at a first stage of the interoperability environment implementation. Similarly to the previous step, the desirable stage corresponds to the validation of any kind of clinical concept. Additionally, it is necessary to define which constraints will be used to validate these concepts. Finally, we need to select the set of archetypes that will be used to validate the clinical concepts and, thus, the documents to be interchanged. The selection of archetypes to be used is a critical task since any inconsistency in the archetype definition may lead to misunderstandings of the clinical information.

Step 3 – Validation Algorithms are implemented in this step taking into consideration: a set of structures that represent the clinical documents; archetypes that define the constraints of such structures, XSD files (*XML Schema Definition*) to support the syntactic validation and terminologies. The implemented architecture provides mechanisms to include new structures, constraints to be validated as well as the overriding of the supported algorithms (e.g. parsing of clinical documents).

Step 4 – Validation algorithms are applied on the clinical documents when they are exchanged in the interoperability environment by the *Document Validation* component, producing an acknowledge code. This validation process is original because it is agnostic to the type of clinical document, so it can be applied to validate different clinical documents such as CDA and OpenEHR.

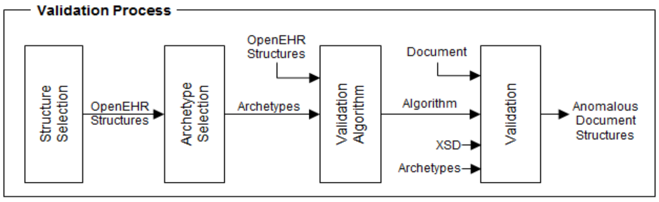

Fig. 2. Clinical Document Validation Process. *(Boxes are steps and arrows connect the inputs and outputs of each step).*

In the following sections detail each step using a common pattern: input, output and description. Also, some steps are illustrated by implementation details.

B. Step 1 - Structure Selection

This step does not have **input** and as **output** it provides a list detailing the OpenEHR structures that will be used to represent the clinical information and to be validated.

The first step of the validation process deals with selecting which OpenEHR model structures will be used to validate clinical documents. We perform the identification in an iterative way. For each primary care procedure: we manually extract all key concepts, check which data structures are used to represent these concepts in existing clinical documents and analyze their dependencies with other structures. In this process, we select data structures that have different semantics and complexity levels such as *single value*, *table of values*, *cluster*, *history*, *event*, *observation* and *extract*.

The step then focuses on the analysis of existing terminologies. Its goal is to ensure that all entities in the architecture consistently rely on the same set of terms. Several clinical terminologies have been documented (e.g. LOINC [8] and SNOMED-CT [11]). We selected *SNOMED CT* as a reference terminology in our current implementation because it covers all areas of clinical data and defines a hierarchical representation to define the relationships among concepts.

C. Step 2 - Archetype Selection

The step receives as **input** a list of OpenEHR structures that will be semantically validated and as **output** it produces a list of archetypes to be used in the document validation.

This step deals with selecting a set of archetypes and constraints that will be used to validate the exchanged clinical information. To this end, we first analyze the international Clinical Knowledge Model repository [15] and select those archetypes that correspond to the clinical information (e.g. procedures) to be exchanged. Then, we manually inspect the selected archetypes and check whether they only rely on the previously selected OpenEHR data structures (Step 1). At the end of this step, we build a list of OpenEHR archetypes to be used in the validation of clinical documents (Section III.E).

D. Step 3 – Validation Algorithm

The step receives as **input** OpenEHR structures to represent the clinical information, and archetypes to validate clinical concepts. It produces as **output** an algorithm to validate clinical documents that are exchanged in the system.

The first step towards the clinical document validation is the development of the selected OpenEHR data structures in an existing programming language – i.e. Java [17]. Its goal is to support the management of clinical documents. Then, we developed a parser to detect syntax errors in clinical documents. We also developed a set of classes to manage the archetype structure. We then developed a semantic parser based on both OpenEHR and archetype structures. For each structure in the clinical document, this parser checks whether the structure content is in conformance with the rules specified in the related archetype. In case there is no archetype associated with the structure under analysis, the parser only checks the terminology of such structure. Finally, an acknowledge code is returned as result of the parsing process. In particular, if the document is considered to be invalid, a list is also returned detailing the set of errors encountered. In the following we detail this development process.

OpenEHR Data Structures: The development of OpenEHR structures was performed on top of an open source initiative [4]. This initiative provides the basis for the implementation of several OpenEHR structures and data types. We extended the open source code by either adding: (i) classes to represent new OpenEHR structures or (ii) new methods in the existing classes to make possible the semantic validation. For instance, we added the getItems(..) method that returns the list of document structures that correspond to a given archetype path. Also, we included the equals(..) method, which returns whether two OpenEHR structures reference the same object. Other added methods (e.g. accept) will be introduced later.

Archetype Model: The development of the archetype information model [1] relies on an open source initiative [4]. As depicted in Figure 3, the RMArchetype class represents a generic archetype element, which is specialized in two subclasses: StructureArchetype and DataArchetype. These classes aim at defining the basis for managing the archetype content. Other classes such as ObservationArchetype, EventArchetype and HistoryArchetype also make this hierarchy. We also extended the existing implementation in order to: (i) recursively load the content of the archetype elements, (ii) process its cardinality and (iii) return its value. Additionally, we added classes to include the processing of new archetype elements such as Cluster, Event, Interval, Ordinal, Ratio and Uri.

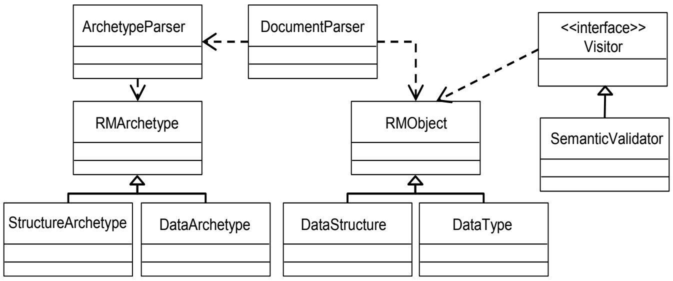

Fig. 3. Implementation slice of the validation algorithm.

Validation: Our implementation rely on XSDs for both CDA and OpenEHR documents to validate the correctness of clinical documents. The comparison of clinical documents with their corresponding XSD structures guarantees that documents are syntactically corrected. The DocumentParser class is in charge of performing such comparison (Figure 3). If the comparison is successfully performed, this class loads the document content into OpenEHR structures and types. Additionally, the archetype elements referred in the document are then loaded into the *archetype* class hierarchy. To perform this operation, we implemented the ArchetypeParser class. At the end of this operation, the built OpenEHR structure is returned if no syntax errors are detected. Otherwise, an acknowledge code is returned with a list detailing which document elements presented errors and the type of such errors. The validation process stops at this point.

Once the document is syntactically correct, the algorithm proceeds with the semantic validation, which is implemented using the Visitor design pattern. The SemanticValidator class starts analyzing the OpenEHR structure returned by the syntax parser. The validator class checks whether the structure content is in conformance with the rules specified in the corresponding archetype (i.e. archetype structures). The validation process follows a recursive implementation since some structures are composed by others. Therefore, composite structures (e.g. Observation) are valid if: (i) they are in conformance with the corresponding archetype specification in terms of terminology, cardinality, order and type of the children structures *and* (ii) their children are valid (e.g. History). A leaf structure is valid if it is in conformance with the rules specified in the corresponding archetype. The parser only checks the terminology if there is no archetype associated with the structure under analysis. This implementation can be extended to support the validation of new structures and archetypes.

E. Step 4 - Validation

The step receives as **input** the clinical document to be validated, validation algorithm, XSD, which defines the syntax structure of clinical documents, and archetypes, which define the structure and semantic of clinical concepts. It generates as **output** a list of document elements whose content is not in conformance with the specified syntax or semantic rules (i.e. those defined in XSD and OpenEHR archetypes).

We automatically execute the validation algorithm on the exchanged clinical documents. This execution can follow a runtime or a batch process. Calling the developed validation algorithm is straightforward, using the API provided by the Document Validation component as shown below.

```
 DocumentParser parser = new DocumentParser();
List<RMObject> rms  = parser.parser(document);
outputfile.println(parser.getErrors());
processSyntaxValidationOutput();
Visitor v = new SematicVisitor(archRepository);
for(RMObject rm : rms)
   rm.accept(v);
outputfile.println(v.getSemanticErrors());
processSemanticValidationOutput();
```

V. Lessons Learned

Mixing of OpenEHR and Interoperability Standards. Our implementation showed how semantic interoperability can be achieved by using simultaneously OpenEHR and interoperability standards. The use of an archetype repository helps to reduce misunderstandings of clinical documents by ensuring the correctness of the clinical information. Specifically, our implementation supports different levels of validation, such as type of structures, mandatory structures, dependencies among structures (e.g. the existence of a given structure implies the existence of another), cardinality and ordering of structures. Our research revealed how many simple errors and inconsistencies in clinical documents (e.g. cardinality, ordering of structures), which are not detected by interoperability standards, can be easily pointed out by including archetypes in the interoperability environment.

Contrast between HL7/IHE and OpenEHR. There are more HL7 and IHE implementations than OpenEHR ones. This seems to indicate that HL7 and IHE are widely adopted standards in industry to support health interoperability. On the other hand, successful industrial cases of OpenEHR are currently scarce. We believe this may occur due to the several factors. First, there is a lack of stable and complete OpenEHR implementations. The open source community has developed several software solutions [4], which represent proof of concepts of the OpenEHR standard. However, they only focus on particular parts of the standard and are not integrated to support semantic interoperability.

Second, there is no strong research evidence about the feasibility of OpenEHR to support large health information systems. We develop an algorithm to semantically validate clinical documents based on OpenEHR. However, our algorithm relies on a subset of OpenEHR structures at its first stage. Other structures need to be included in the future. There are some studies available in the literature [3][5][7][9], but they are limited in terms of clinical concepts and information to be interchanged. Also, there is no empirical knowledge about the cost of implementing a health systems based on OpenEHR. There is no evidence about the performance of OpenEHR when dealing with huge amount of data. This is an important point since the validation of clinical documents against a repository of archetypes as well as the archetype dynamic loading are processes that might demand high consumption of computational resources. Finally, the integration of OpenEHR with different document standards should further explored.

Need of Creating a Federated Clinical Repository. One of the key aspects of our validation process is the definition of a common archetype repository. We used the Clinical Knowledge Manager (CKM) at the first stage of the development. It defines archetypes for hundreds of clinical concepts. However, it was observed that the already published archetypes need to be carefully reviewed to attend the requirements of different Brazilian health programs. For instance, adaptations need to be made in the terminologies, measurement units and clinical concepts to attend these requirements. Also, new archetypes had to be created in order to define observations and evaluations for local diseases.

These observations highlight the need of creating a selected committee of specialists, representatives of different health insurance and government needs to support the development of a semantic interoperable environment. This committee would be in charge of defining and maintaining a federated repository of clinical concepts and terms to be used among the different health information systems. The need of creating a federated repository of clinical information is relevant given the demand of the Brazilian Ministry of Health to adopt different health standards, including OpenEHR.

VI. Concluding Remark

The proposed software architecture revealed that combining OpenEHR with interoperability standards helps to increase the consistency of clinical documents. Our design showed that a number of new validations (those based on archetype constraints) can be supported, without impacting the behavior of the existing interoperability standards. It indicates that both types of standards can coexist towards achieving semantic interoperability. Moreover, we illustrated how the proposed validation process can be extended to other systems. We also revealed potential improvements aiming to increase the use of OpenEHR by the community. In general, a higher number of OpenEHR implementations should be available in order to assess its benefits and foster its use in industrial scenarios. Moreover, we identified the integration of OpenEHR with other clinical documents (e.g. CDA) should be further explored by researchers. In addition, there is also a need to perform case studies of OpenEHR implementations, involving a high number of clinical documents and archetypes. As future work we plan to compare our system with existing ones.

References

[1] Beale, T. “Archetypes: Constraint-based domain models for future-proof information systems”. In: OOPSLA pp. 1-18, 2002.

[2] Clements, P.; Bachmann, F.; Bass, L.; Garlan, D.; Ivers, J.; Little, R.; Nord, R.; Stadord, J. “Documenting Software Architectures: Views and Beyond”. Addison-Wesley Professional, 2002.

[3] Dentler, K.; Teije, A.; Cornet, R.; Keizer, N. “Semantic integration of patient data and quality indicators based on openEHR archetypes”. International Conference on Process Support and Knowledge Representation in Health Care pp. 85-97, 2012.

[4] Frade, S.; Miranda Freire, S.; Sundvall, E.; Patriarca-Almeida, J.H. “Survey of openEHR storage implementations”. In IEEE 26th International Symposium on Computer-Based Medical Systems pp. 303-307, 2013.

[5] Garde, S.; Knaup P.; Hovenga, E.; Heard, S. “Towards Semantic Interoperability for Electronic Health Records”. Methods Inf Med, 46: pp. 332–343, 2007.

[6] Gibbons P.; Arzt, N.; Burke-Beebe, S.; Chute, C.; Dickinson, G.; *et al.* “Coming to Terms: Scoping Interoperability in Health Care. Final” Health level Seven EHR Interoperability Work Group, 2007

[7] Heymans, S.; McKennirey, M.; Phillips, J. “Semantic validation of the use of SNOMED CT in HL7 clinical documents”. Journal of Biomedical Semantics 2(1), 2011.

[8] LOINC https://loinc.org/. [accessed 03/05/2013]

[9] Menárguez-Tortosa, M., Fernández-Breis, J. “Validation of the openEHR Archetype Library by using OWL Reasoning”. Studies in Health Technology and Informatics 169, 789 (2011).

[10] Silva-Ferreira, P.R; Patriarca-Almeida, J.H.; Vieira-Marques, P.M.; Cruz-Correia, R.J. “Improving expressiveness of agents using openEHR to retrieve multi-institutional health data”. In 7th Iberian Conference on Information Systems and Technologies, pp. 1-5, 2012.

[11] SNOMED Clinical Terms User Guide. International HealthTerminology Standards Development Organisation, Copenhagen, 2009.

[12] The Brazilian Ministry of Health Resolution: http://bvsms.saude.gov.br/bvs/saudelegis/gm/2011/prt2073_31_08_2011.html [accessed in 02/21/2013]

[13] The HL7 http://hl7.org/. [accessed in 04/03/2013].

[14] The IHE http://ihe.net/. [accessed in 04/03/2013].

[15] The OpenEHR. http://openehr.org/ [accessed in 05/13/2013]

[16] The OpenMRS. http://openmrs.org/. [accessed in 06/21/2013].

[17] The Java language: http://www.oracle.com/technetwork/java/index-138747.html [accessed in 09/12/2012]