© 2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

Towards Semantic Interoperability of Electronic Health Records

Idoia Berges, Jesús Bermúdez and Arantza Illarramendi

***Abstract*—Although the goal of achieving semantic interoperability of Electronic Health Records (EHRs) is pursued by many researchers, it has not been accomplished yet. In this paper we present a proposal that smoothes out the way towards the achievement of that goal. In particular our work focuses on medical diagnoses statements. In summary the main contributions of our ontology-based proposal are the following: First, it includes a canonical ontology whose EHR-related terms focus on semantic aspects. As a result, their descriptions are independent of languages and technology aspects used in different organizations to represent EHRs. Moreover, those terms are related to their corresponding codes in well-known medical terminologies. Second, it deals with modules that allow obtaining rich ontological representations of EHR information managed by proprietary models of health information systems. The features of one specific module are shown as reference. Third, it considers the necessary mapping axioms between ontological terms enhanced with so-called path mappings. That feature smoothes out structural differences between heterogeneous EHR representations, allowing proper alignment of information.**

***Index Terms*—Electronic Health Record, Semantic Interoperability, Ontology.**

I. Introduction

In 2009 the European Community presented a long-term research and deployment roadmap that provides the key steps for achieving semantic interoperability in the area of healthcare[1]. The incorporation some years ago of Electronic Health Records to the healthcare institutions may be seen as the first step towards the achievement of the goal, since, apart from local advantages over manual records such as avoiding legibility problems, they favour a fast exchange of clinical data between different organizations. However, the fact that most healthcare institutions have developed their health information systems in an autonomous way has resulted in a proliferation of heterogeneous health information systems, each one with its own proprietary model for representing and storing EHR information, which hinders the task of interoperating with each other.

In many areas, the adoption of knowledge representation standards stands out as the most usual approach to solve interoperability problems. This happens also in the healthcare area, where some standards such as openEHR[2], ISO 13606[3] and HL7-CDA[4] are under development for this purpose. All three follow a dual model-based methodology for representing EHR information: the Reference Model defines basic structures such as List, Table, etc., while the Archetype Model defines knowledge elements (such as Respiration Rate) by using and constraining the elements of the Reference Model. Although the idea of using a standard may seem suitable for the considered goal, we think that interoperability does not mean to have a unique representation but a semantically acknowledgeable equivalent one. This would relieve healthcare institutions from being forced to use one standard in the representation of their knowledge and moreover, since several standards are being developed for the same purpose, the interoperability problem will remain unsolved unless these standards merge into a single one. Currently, some research is being done on the latter issue[5].

In this paper we present a proposal to move towards the notion of full semantic interoperability of heterogeneous EHRs, which states that when one particular system receives some EHR information from another healthcare institution, the received information can be seamlessly integrated into its underlying repository because the differences in the language, in the representation of the information and in the storing systems do not cause any misunderstanding[1]. Two general approaches for interoperability among systems are described in [6]: Using a canonical model to which the particular models are linked or aligning the particular models two by two. The proposal presented in this paper is sustained in the former approach. More precisely, it is an ontology-based approach where OWL2[7] ontologies are used as representation models. In general, ontologies have been considered relevant for several purposes such as: enabling reuse of domain knowledge, allowing the analysis of domain knowledge and sharing common understanding of the meaning of information[8]. Our approach benefits from the latter advantage and additionally it provides the following ones:

- It favors the notion of semantic interoperability: The use of a formal ontology as canonical conceptual model allows to focus on aspects that are independent of the languages or technologies used to describe EHRs.
- It favors the notion of extensibility to different models: The framework comprises two kinds of ontologies which represent the definitions of clinical terms that appear in EHRs at different levels of abstraction. The canonical contains ontological definitions of EHR statements and the application ontologies contain specializations of the definitions of the canonical ontology according to the standards mentioned previously or according to proprietary models of healthcare institutions.
- It decreases the need of human intervention: The frame-

All authors are with the Department of Languages and Information Systems, University of the Basque Country, Donostia-San Sebastián, 20018 Spain. e-mail: {idoia.berges, jesus.bermudez, a.illarramendi}@ehu.es.

Manuscript received XX; revised XX.

work relies on a reasoning mechanism that, using axioms stated in the ontology, infers knowledge that allows the discovery of more relationships among the heterogeneous models used by the different health information systems.

Dealing with ontologies, one relevant aspect is the features of the terms that are part of them. In our scenario those terms are related to EHRs. Different kinds of information can be found in an EHR. OpenEHR divides this information into 5 subtypes[9] and we also have adopted that division in the definition of our canonical ontology: *Observations* comprise the data that can be measured in an objective way, such as the age of a patient, his respiration rate, etc. *Evaluations* represent the evidence obtained from observations, for example the diagnosis of an illness. *Instructions* represent actions to be performed in the future such as the prescription of a medicine or the request of a laboratory test. *Actions* are used to model the information recorded due to the execution of an instruction and finally there is one last type to record *administrative* events such as admission or discharge information. In this paper we just focus on one type of evaluations, namely the diagnoses, but similar ideas to those that will be explained here for diagnoses could be also applied to the other types of information. Moreover, the terms of the application ontologies are obtained from the particular health information systems and then linked to the terms in the canonical ontology by using ontology mappings.

A certain number of works related to ours can be found at present. With regard to the benefits of taking semantics into account, some works discuss the convenience of using semantic technologies in several heathcare related issues. In [10], the handicaps for widespread adoption of semantic technologies within a care records system are pinpointed. In [11] the challenges to be addressed in order to be able to use the so-called Smart Internet to enable reforms on healthcare information systems are discussed. Lastly, in [12] the triplespace paradigm is suggested as semantic middleware to support pervasive access to electronic patient summaries. The works mentioned next also rely on semantic technologies for interchanging data, as opposed to other formats such as XML, which are structure-based. More specifically, related to the topic of facilitating semantic interoperability between heterogeneous health information systems, the following works deal only with the interoperability between standard-based health information systems: [13] provides a solution to achieve semantic interoperability between systems that have been developed under the HL7 reference model and which requires that the source system has some prior knowledge about the target system. In [14] ontology mappings are proposed between pairs of archetype-based models. In [15] a model-driven engineering approach that transforms archetypes of the ISO 13606 standard into OWL models is presented. Finally, authors in [16] describe an approach to translate from the Archetype Definition Language (ADL[17]) to OWL, they also present some techniques to map archetypes to formal ontologies and show the convenience of using semantic rules on the resulting representation in order to guide the execution of primary care guidelines. In this paper we present a wider approach since apart from the interoperability of standard-based systems we deal also with interoperability considering proprietary models. Some other works that tackle the problem of semantic interoperability of EHRs from a different perspective are the following: In [18] a semantic conceptualization model for an EHR system is presented. This still early work is more oriented towards the accessibility, use and management of the EHR at a local level, but it also aims at providing a base in order to solve the interoperability problem from a semantic point of view. In [19] the hypothesis that semantic technologies are potential bridging technologies between the EHRs and medical terminologies –as well as a possible representation of the combined semantics of systems to be integrated– is raised and some experimental study is made on this issue. We also promote the connection between the semantic representation of EHR statements and their codes in well-known terminologies. Finally, in [20] authors discuss how advanced middleware, such as Enterprise Middleware Bus, and semantic web services can assist in solving interoperability issues between eHealth systems.

The rest of the paper is divided as it follows: In Section II the global architecture of the framework is presented, and extensive details about the canonical ontology and the auxiliary modules DB2OntoModule and MappingModule are given. The feasibility of the solution is shown in Section III, and finally, conclusions are discussed in Section IV.

II. Global architecture

In Fig. 1 the three-layered architecture of the solution can be found. The *lower layer*, contains the particular underlying repository of each healthcare institution, where the information of the EHRs is stored. Associated to each kind of underlying repository, there is some kind of file (e.g. database schema, set of ADL files) where information about the structures in the repository can be found. Then, the *middle layer* contains one application ontology for each information system, built on top of the underlying repository. These application ontologies are created semi-automatically from the underlying repositories by some auxiliary modules (e.g. DB2OntoModule and ADL2OntoModule), or imported from an ontology repository, and describe semantically each underlying repository. Moreover, they are linked to their corresponding repositories by some Σ links that specify how to transfer information from each of the representations to the other. Finally the *upper layer* contains one canonical ontology. This ontology will contain the necessary classes and properties to represent the different types of information that can be found in an EHR and is linked to each of the application ontologies by some integration mappings defined by a MappingModule. Each particular healthcare institution will have only a partial view of the global framework, since with our proposal there is no need for that institution to know anything but its underlying repository, its application ontology, and the canonical ontology.

The proposed framework allows one healthcare institution to interpret on the fly clinical statements sent by another one – even when they use proprietary formats. We support our claim on the following techniques:

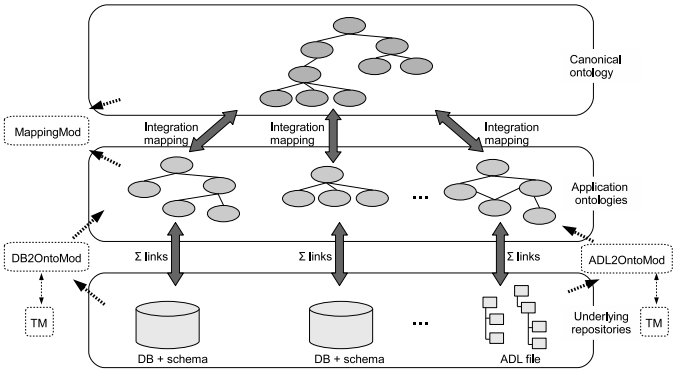

Fig. 1. Global architecture of the solution

- *Logic-based descriptions:* Representations of diagnoses considered by particular health information systems, described using standards as well as proprietary models, are expressed in our approach by using OWL2 ontology axioms. Moreover, terms in those axioms are related with canonical ontology terms that focus their descriptions on language and technology independent aspects. This approach increases the opportunities of solving the interoperability issue since it relies mainly on semantic aspects.
- *Automated reasoning:* All ontology descriptions, as well as the mappings among elements of the ontologies, are expressed in the same formalism OWL2. This uniform representation allows the use of well-known reasoners in order to derive new statements from the existing ones. Furthermore, the mismatch problem is avoided and automatic integration is facilitated.
- *Transfer mechanism:* A process, guided by the previous two items, is implemented to transform a particular clinical statement from a healthcare institution into a corresponding clinical statement for another healthcare institution. So-called path mappings play a crucial role during the transfer process, smoothing out the structural differences between EHR representations.

Finally, we are aware that the messiness of real world EHRs may sometimes hinder the task of fitting them into the presented proposal, but in our opinion this does not invalidate the advantages it can provide in many situations.

In the following subsections, the canonical ontology, the *DB2OntoModule* and the *MappingModule* are described thoroughly.

A. Canonical ontology: Representing diagnoses in OWL

The canonical ontology contains the necessary classes and properties to represent the different types of information that can be found in an EHR. Following openEHRs classification of EHR entries, we have defined in the ontology five classes to represent the general categories: `Observation`, `Evaluation`, `Instruction`, `Action` and `Admin`. Moreover, these five classes have been specialized to represent more specific types of entries. As we pointed out in the introduction, in this paper we deal with diagnoses, which are a special case of evaluations.

A *diagnosis* is defined as the act of identifying a disease from its signs and symptoms, as well as the decision reached by that act[1]. For this reason, in addition to representing a diagnosis as a subclass of `Evaluation`, its definition is enhanced with two properties: `hasFinding`, to indicate the conclusion reached by the physician about what is happening to the patient, and `hasObs`, to indicate the information about the observation(s) which lead to that assessment[2].

$$\texttt{Diagnosis} \equiv \texttt{Evaluation} \sqcap =1\ \texttt{hasFinding.Finding} \sqcap \exists \texttt{hasObs.Observation}$$

Specific diagnoses are defined as subclasses of the class `Diagnosis`. For example, the evidence obtained as a result of an ECG can be described by specializing the range restrictions of the properties `hasFinding` and `hasObs`. For instance, the observation that leads to an ECG diagnosis is an ECG Recording, which is made up of several components[3]: some of the components refer to information about the heart's electrical axis (i.e. the general direction of the heart's depolarization wavefront), while the others refer to information about the entire ECG.

$$\begin{aligned}
\texttt{ECGDiagnosis} &\equiv \texttt{Diagnosis} \sqcap =1\ \texttt{hasFinding.ECGFinding} \sqcap \exists \texttt{hasObs.ECGRecording} \\
\texttt{ECGRecording} &\equiv \texttt{Observation} \sqcap \exists \texttt{comp.P-Axis} \sqcap \exists \texttt{comp.QRS-Axis} \sqcap \exists \texttt{comp.T-Axis} \sqcap \\
&\quad \exists \texttt{comp.PR-Interval} \sqcap \exists \texttt{comp.QT-Interval} \sqcap \exists \texttt{comp.QTc-Interval} \sqcap \exists \texttt{comp.QRS-Duration} \sqcap \\
&\quad \exists \texttt{comp.Heart-Rate}
\end{aligned}$$

One advantage of working in the medical area is the existence of medical terminologies, such as SNOMED[21] and LOINC[22]. These terminologies cover most areas of clinical information and provide a consistent way to identify medical terms univocally, which can be very helpful at the time of gathering and exchanging clinical results. Our system takes advantage of these terminologies to enhance the definition of the classes in the canonical ontology. Thus, whenever is possible, each term in the ontology is related to its corresponding code in those terminologies:

$$\begin{aligned}
\texttt{ECGDiagnosis} &\equiv \exists \texttt{loinc.\{'8601-7'\}} \\
\texttt{ECGRecording} &\equiv \exists \texttt{loinc.\{'34534-8'\}} \\
\texttt{P-Axis} &\equiv \exists \texttt{loinc.\{'8626-4'\}}
\end{aligned}$$

The use of terminological codes into the definitions of the classes in the ontology increases the chances of achieving a successful communication.

Finally, since building a canonical ontology is not an easy task, we think that efforts that are being done to define archetypes in openEHR could be reused to achieve that task.

[1] http://www.merriam-webster.com/dictionary/diagnosis

[2] For the presentation we prefer a logic notation instead of the more verbose RDF/XML syntax.

[3] For the sake of brevity, in this example only some components of an ECG are taken into account. Please refer to [2] for the whole set of components.

B. DB2OntoModule

Taking into account the widespread use of relational databases to store EHR records, we show in this subsection the main features of the module DB2OntoModule[4]. This module takes as input a database schema and after applying a set of rules based on schema features, it obtains the ontological representations of those relational databases (i.e. the application ontology of that system). In the specialized literature many approaches for translating relational structures into more expressive formalisms can be found: object models, description logics and Semantic Web technologies. Some of them follow the so-called declarative approach, which first convert the relational structure into a declarative language and then the result is modified by the user to declare additional knowledge about the database (e.g [23]). Our proposal also uses the declarative approach but its novel contribution relies in the large number of schema properties that it considers, allowing to make explicit more knowledge, and in the fact that it associates to the obtained classes their corresponding codes that appear in well-known medical terminologies.

In order for the DB2OntoModule to accomplish the last feature, it deals with an element called "Terminology Manager" or, in short, "TM", which has an associated function of the form *getX(conceptName)*, where *X* is the name of a terminology (LOINC, SNOMED, or any other) and *conceptName* is the name of the relation or attribute whose terminological code is to be retrieved. For example, in the case of a relation *BloodPressure(id, systolic, diastolic)* the TM would contain:

Identifying path	LOINC	SNOMED
BloodPressure	18684-1	75367002
BloodPressure.systolic	8480-6	72313002
BloodPressure.diastolic	8462-4	271650006

Concerning schema features, the DB2OntoModule works as follows:

Relations: Relations of the relational schema are translated into OWL2 classes. Moreover, if for a given relation R, TM.*getLOINC(R)=LC* (being LC a particular LOINC code), a new axiom $\texttt{R} \equiv \exists \texttt{loinc.}\{\texttt{'}LC\texttt{'}\}$ is added to the ontology (analogously for other terminological codes).

Attributes: Two options arise: (1) If for a given attribute a in R ($R.a$) TM.*getLOINC(R.a)* returns some code LC, then a new class A is created (if there is no other class which already has that code). Moreover, the axioms $\texttt{A} \equiv \exists \texttt{loinc.}\{\texttt{'}LC\texttt{'}\}$ and $\texttt{A} \sqsubseteq \exists \texttt{value.}\mathit{getType(a)}$ are added. Finally, if attribute a is compulsory in R, the axiom $\texttt{R} \sqsubseteq \exists \texttt{hasA.A}$ is added. (2) If there is no code for $R.a$ in TM, a property a is created in the ontology, where *Domain(a)*=R and *Range(a)=getType(a)*. Moreover, if attribute a is compulsory in R, the axiom $\texttt{R} \sqsubseteq \exists \texttt{a}$ is added.

Integrity constraints: An integrity constraint such as *R.a*>30 adds a new axiom of type $\texttt{R} \sqsubseteq \exists\ \texttt{hasA.(A and } \exists \texttt{value[>30])}$ if $R.a$ is in TM, and a new axiom of type $\texttt{R} \sqsubseteq \exists\ \texttt{a[>30]}$ otherwise.

[4]Other modules, such as the ADL2Onto module, would be used to perform the translations between other sources and the ontology

Once the previous steps are accomplished, the next one involves enriching the obtained descriptions by using several types of information, such as inclusion, exclusion and functional dependencies:

Inclusion dependencies: Three different situations are considered (see a previous work [24] from our group for more details): (1) Dependencies between key (*R.K*) and non-key (*S.x*) attributes, which indicate the existence of a foreign keys of type *S.x*⊆*R.K*. These dependencies are reflected by defining an association between the ontology classes obtained from those relations ($\texttt{S} \sqsubseteq \exists \texttt{x.R}$); (2) Dependencies between the keys of two relations (*R.K*⊆*R'.K'*); and (3) Dependencies between a subset of a key and a key (*R.subK*⊆*R'.K'*), which also have the corresponding reflection.

Exclusion dependencies: An exclusion dependency between the keys of two relations (*R.K*∩*R'.K'*=∅) creates a new axiom of the form $\texttt{R} \sqsubseteq \neg \texttt{R}'$ in the ontology. In addition, if there is no class in the ontology that subsumes both R and R′, such new class S is created and the axioms $\texttt{R} \sqsubseteq \texttt{S}$ and $\texttt{R}' \sqsubseteq \texttt{S}$ are added.

Functional dependencies: If a functional dependency of the form $R.X \rightarrow R.y$ is detected, with X and y being a non-key attribute set and a non-key attribute respectively, a new class X is created. Moreover, two new properties hasX and hasY are defined and the axioms $\texttt{R} \sqsubseteq \exists \texttt{hasX.X}$ and $\texttt{X} \sqsubseteq \exists \texttt{hasY.}\mathit{getType(y)}$ are added to the ontology.

Furthermore, the ontology can be enriched by using domain information for attribute values, for example, in the case of properties expressed by enumerating attribute values. For an attribute $R.a$ whose possible values are either $A1$ or $A2$, if both have a corresponding code in the TM, classes A1 and A2 are created in the ontology. Moreover, one general class to group those two classes is created (e.g. A0) and axioms $\texttt{A1} \sqsubseteq \texttt{A0}$, $\texttt{A2} \sqsubseteq \texttt{A0}$, $\texttt{A1} \sqsubseteq \neg \texttt{A2}$ and $\texttt{R} \sqsubseteq \forall \texttt{a.A0}$ are added. However, in the case where $A1$ and $A2$ have no terminological code in the TM, class A0 is created as an enumeration of two individuals $a1$ and $a2$, and axiom $\texttt{R} \sqsubseteq \forall \texttt{a.A0}$ is added too.

All the previous types of considerations are applied in the following sequence: first inclusion dependencies; then when the input relational schema is not in second or third normal form, functional dependencies are used to create new classes; next exclusion dependencies are exploited and last integrity constraints and domain information for attribute values are considered. Finally, once the DB2OntoModule has performed the steps above, a candidate ontology has been created. However, we feel that it is advisable to allow the health system administrator to modify the ontology in a flexible way. For example, some common changes could be substituting $\sqsubseteq$ relationships with $\equiv$ relationships, modifying the names of the terms that have been created, or adding some missing terminological code. These changes can be done manually using any well-known ontology editor.

The DB2OntoModule at work: For example, a particular registration for an ECG diagnosis may consist of four relational tables according to the following schema (all attributes are considered compulsory)

ECGDiagnosis(code, finding, recording)

```
ECGObservation(code, axis, global)
ECGAxis(code, P-Axis, QRS-Axis, T-Axis)
ECGGlobal(code, PR-Interval, QT-Interval,
   QTC-Interval, QRS-Duration, Heart-Rate)
```

and the following inclusion dependencies between non-key and key attributes:

$$\texttt{ECGDiagnosis.recording} \subseteq \texttt{ECGObservation.}\underline{\texttt{code}}$$
$$\texttt{ECGObservation.axis} \subseteq \texttt{ECGAxis.}\underline{\texttt{code}}$$
$$\texttt{ECGObservation.global} \subseteq \texttt{ECGGlobal.}\underline{\texttt{code}}$$

Moreover, let us consider the bogus case where the attribute `finding` of the ECGDiagnosis table must be either "Normal ECG" or "Abnormal ECG". As a result of applying the initial steps for transforming the schema to ontology elements four new classes are created in the ontology: `EGCDiagnosis`, `ECGObservation`, `ECGAxis` and `ECGGlobal`, each with its respective LOINC code. Moreover, since the compulsory attribute `P-Axis`, whose type is "int", has also a LOINC code at the TM, axioms $\texttt{P-Axis} \equiv \exists \texttt{loinc.}\{\texttt{'8626-4'}\}$, $\texttt{ECGAxis} \sqsubseteq \exists \texttt{hasP-Axis.P-Axis}$ and $\texttt{P-Axis} \sqsubseteq \exists \texttt{value.xsd:int}$ are created (same process for the other attributes). Then, the rules for inclusion dependencies are applied, and, for example, from the inclusion dependency $\texttt{ECGObservation.axis} \subseteq \texttt{ECGAxis.}\underline{\texttt{code}}$, axiom $\texttt{ECGObservation} \sqsubseteq \exists \texttt{axis.ECGAxis}$ is created. Moreover, information about the allowed values for the `finding` attribute is considered and a new class `ECGFinding` is created as superclass of two other classes `NormalECG` and `AbnormalECG`. Finally, manual changes are applied. For example, we have chosen to substitute some of the subclass relationships with equivalence relationships, so the created ontology has, among others, the following axioms[5]:

$$\begin{aligned}
\texttt{a:ECGDiagnosis} &\equiv \exists \texttt{a:finding.a:ECGFinding} \sqcap \\
&\quad\ \exists \texttt{a:recording.a:ECGObservation} \\
\texttt{a:ECGDiagnosis} &\equiv \exists \texttt{a:loinc.}\{\texttt{8601-7}\} \\
\texttt{a:ECGObservation} &\equiv \exists \texttt{a:hasAxis.a:ECGAxis} \sqcap \\
&\quad\ \exists \texttt{a:hasGlobal.a:ECGGlobal} \\
\texttt{a:ECGObservation} &\equiv \exists \texttt{loinc.}\{\texttt{34534-8}\} \\
\texttt{a:ECGAxis} &\equiv \exists \texttt{a:hasP-Axis.a:P-Axis} \sqcap \\
&\quad\ \exists \texttt{a:hasQRS-Axis.a:QRS-Axis} \sqcap \\
&\quad\ \exists \texttt{a:hasT-Axis.a:T-Axis} \\
\texttt{a:NormalECG} &\equiv \texttt{a:ECGFinding} \sqcap \texttt{a:value.}\{\texttt{"Normal ECG"}\} \\
\texttt{a:NormalECG} &\equiv \texttt{a:snomed.}\{\texttt{102593009}\}
\end{aligned}$$

The second task of the DB2OntoModule is to create the Σ links that indicate how to transfer the information from the database to the ontology that has been created from it (and vice versa). This task was previously tackled by our research group, so we refer to the reader to [25] for further technical details.

[5]Throughout the paper, namespaces `a:` and `b:` will be used to refer to terms in the application ontologies of two particular systems A and B. Moreover, namespace `c:` or no namespace are used to indicate the terms in the canonical ontology.

C. MappingModule

Once an application ontology of one particular system has been generated by the corresponding translator module, it must be integrated with the canonical ontology, and the mappings between the terms of that application ontology and the canonical ontology must be created. A *MappingModule* has been implemented for this purpose. Wide research has been done in the specialized literature about ontology mapping (e.g. [26]), so working in new techniques for that same issue is out of the scope of our work. So, our MappingModule takes a pragmatic approach and receives as input a set of basic mapping axioms specifically defined by the system administrator (for example, to state that the property `a:loinc` is equivalent to the property `c:loinc`). Then, it incorporates these basic mappings into the ontologies and, with the help of a reasoner, it creates an integration mapping that relates the terms of the application ontology with those of the canonical ontology.

However, our module presents a distinguishing feature, since it considers mappings between ontology paths, which are rarely considered in other works. In order to be aware of the importance of discovering path mappings, let us compare the definitions of classes `c:ECGRecording` and `a:ECGObservation` in sections II-A and II-B respectively. Both share the same LOINC code (34534-8), so their semantics are the same. Looking at the description of `c:ECGRecording` , it can be seen that any individual belonging to that class will be directly related to an individual of the class `c:P-Axis` via the property `c:comp` (assume the same intuition for the other components). However, in the case of the descriptions in the application ontology of system A, it turns out that classes `a:ECGObservation` and `a:P-Axis` are not directly related, but indirectly: first `a:ECGObservation` is related to the class `a:ECGAxis` via the property `a:hasAxis` and then the class `a:ECGAxis` is related to the class `a:P-Axis` via the property `a:hasP-Axis` . Then it could be stated that there is a simple path between classes `c:ECGRecording` and `c:P-Axis` , while there is a composite path between classes `a:ECGObservation` and `a:P-Axis` .

Intuitively, those two paths could be regarded as equivalent, since their only difference is from the structural point of view caused by the heterogeneous origin of the ontologies, not from a semantic point of view. Let us show how our module deals with that aspect:

Definition 1. *An ontology path is a regular expression of the form* $\texttt{A.(p.[B])}^{+}$ *where* `A,B` *represent class names and* `p` *represents property names, all from the same ontology.*

Let us denote equivalences between paths with the symbol $\equiv_p$. For instance, the aforementioned example is represented as:

$$\texttt{a:ECGObservation.a:hasAxis[a:ECGAxis].a:hasP-Axis.[a:P-Axis]} \equiv_p \texttt{c:ECGRecording.c:comp[c:P-Axis]}$$

Although in this example an equivalence path mapping has been presented, a corresponding idea is valid for subclass path mappings ($\sqsubseteq_p$) and superclass path mappings ($\sqsupseteq_p$). In order to determine path mappings, first path mapping candidates are searched:

Definition 2. *Let* $Path_C = C_0.p_1[C_1]....p_n[C_n]$ *and* $Path_D = D_0.q_1[D_1]....q_m[D_m]$ *be two ontology paths. A path mapping candidate exists between* $Path_C$ *and* $Path_D$ *if any of the following statements holds:*

- $C_0 \sqsubseteq D_0$ *and* $C_n \sqsubseteq D_m$*(represented as* $Path_C \overset{\sqsubseteq}{\approx} Path_D$*)*
- $C_0 \sqsupseteq D_0$ *and* $C_n \sqsupseteq D_m$*(represented as* $Path_C \overset{\sqsupseteq}{\approx} Path_D$*)*

Moreover, if $Path_C \overset{\sqsubseteq}{\approx} Path_D$ *and* $Path_C \overset{\sqsupseteq}{\approx} Path_D$ *then* $Path_C \overset{\equiv}{\approx} Path_D$

A path mapping candidate becomes a proper path mapping when the semantics of both paths is found to be the same. Path mappings are useful at the time of transforming individuals from one ontology so that they meet the requirements of the target ontology. The implementation of path mappings is done by using SWRL[27] rules. SWRL increases the expressivity of OWL and thus allows to model more domain knowledge than the one achieved by using OWL in its own. Moreover, since SWRL can be tightly integrated with OWL, there is no impedance mismatch between the modelling language and the rules language: SWRL rules can use directly the classes, properties and individuals defined in the OWL model. For example, the path mappings shown before would be implemented using the following rules (one in each way):

$R1$ `a:ECGObservation(?e)` $\wedge$ `a:hasAxis(?e,?x)` $\wedge$ `a:hasP-Axis(?x,?p)` $\rightarrow$ `c:comp(?e,?p)`

$R2$ `c:ECGRecording(?e)` $\wedge$ `c:comp(?e,?p)` $\wedge$ `c:P-Axis(?p)` $\wedge$ `swrlx:createOWLThing(?e,?x)` $\rightarrow$ `a:hasAxis(?e,?x)` $\wedge$ `a:ECGAxis(?x)` $\wedge$ `a:hasP-Axis(?x,?p)`

As looking for all the candidate path mappings between two large ontologies might be a hard task considering both time and resources, a threshold can be established to indicate the maximum length of the paths to be searched. Some other heuristics could be applied too to discover candidate path mappings efficiently.

So, to sum up, the *integration mapping* that is generated between an application ontology and the canonical ontology can be defined as it follows:

Definition 3. *An integration mapping is a structure* $\mathcal{I} = \langle O, G, \mathcal{M} \rangle$ *where* O *is a set of OWL2 axioms that comprises the application ontology corresponding to a healthcare institution,* G *is the set of OWL2 axioms for the canonical ontology, and* $\mathcal{M}$ *is a set of mapping axioms of any of the following forms:*

- $C_o \sqsubseteq Exp_g$*,* $C_o \sqsupseteq Exp_g$ *or* $C_o \equiv Exp_g$*, where* C_o *is a class name from* O*, and* Exp_g *is a OWL2 class expression that uses only terms from* G*.*
- $p_o \sqsubseteq p_g$ *or* $p_o \sqsupseteq p_g$ *, where* p_o *is a property name from* O*, and* p_g *is property name from* G*.*
- $sameAs(i_o, i_g)$ *, where* i_o *is the name of an individual from* O*, and* i_g *is the name of an individual from* G*.*
- $Path_o \sqsubseteq_p Path_g$*,* $Path_o \sqsupseteq_p Path_g$ *or* $Path_o \equiv_p Path_g$*, where* $Path_o$ *is an ontology path in* O *and* $Path_g$ *is an ontology path in* G*.*

The result of the engineering process of producing the set $\mathcal{M}$ of mapping axioms is the key for the interoperability of two different health information systems.

III. Framework at work

The main contribution of our proposal is the capability of one system B of interpreting information sent by another system A on the fly, without prior peer-to-peer agreement on the semantics and syntax of the interchanged data. In this example, let us suppose that the database schema of system A is the one presented in section II-B. Moreover, in the case of system B, let us consider that it follows the HL7 standard and that different representations are used to represent ECG information depending on the result of the ECG (e.g.: `ECGNormalDiag` for normal ECG results, `ECGAbnormalDiag` when abnormalities have been detected). The work of the ADL2OntoModule and MappingModule led to the following axioms, with respect to the application ontology of system B:

`b:ECGNormalDiag`	$\equiv$	`b:ECGDiagnosis` $\sqcap$
		`=1 b:finding.b:ECGNormalFind` $\sqcap$
		$\exists$`b:component.b:P-Ax` $\sqcap$
		$\exists$`b:component.b:QRS-Ax` $\sqcap$ ... $\sqcap$
		$\exists$`b:component.b:Heart-R`
`b:ECGDiagnosis`	$\equiv$	`b:loinc.{8601-7}`
`b:ECGNormalFind`	$\equiv$	`b:snomed.{102593009}`
`b:loinc`	$\equiv$	`c:loinc`
`b:component`	$\equiv$	`c:comp`
`b:snomed`	$\equiv$	`c:snomed`
`b:finding`	$\equiv$	`c:finding`
p1	$\sqsubseteq_p$	*p2*

where *p1*=`b:ECGNormalDiag.b:component[b:P-Ax]` and *p2*=`c:ECGDiagnosis.c:hasObs[c:ECGRecording].c:comp[c:P-Axis]`.

Moreover, let us suppose that system A wants to send to system B the following information about the ECGDiagnosis whose *code* is *ecg01*:

$\sigma_{code='ecg01'}$(*ECGDiagnosis*)= (ecg01, Normal ECG, r01)
$\sigma_{code='r01'}$(*ECGRecording*)= (r01, ax01, gl01)
$\sigma_{code='ax01'}$(*ECGAxis*)= (ax01, 27, 88, 49)
$\sigma_{code='gl01'}$(*ECGGlobal*) = (gl01, 138, 390, 39, 112, 62)

Finally, assume that some of the mapping axioms between the application ontology of system A and the canonical ontology are the following:

`a:loinc`$\equiv$`c:loinc`	`a:snomed`$\equiv$`c:snomed`
`a:finding`$\equiv$`c:hasFinding`	`a:recording`$\sqsubseteq$`c:hasObs`
`a:hasAxis`$\sqsubseteq$`c:comp`	`a:value`$\equiv$`c:value`

The process that needs to be carried out is composed of several steps:

Step 1: Classification of the information in the application ontology. In this step the information to be sent is converted into statements about individuals generated for

the application ontology of system A. For example, the main individual a:ecg01 will be an instance of the class a:ECGDiagnosis. This is a straightforward process thanks to the Σ links created by the DB2OntoModule between the storage system of system A and its application ontology. Among others, the following OWL statements (represented as triples) will be created:

(a:ecg01 *rdf:type* a:ECGDiagnosis) (a:r01 *a:hasAxis* a:ax01)
(a:ecg01 *a:finding* a:f01) (a:ax01 *rdf:type* a:ECGAxis)
(a:f01 *a:value* "Normal ECG") (a:ax01 *a:hasP-Axis* a:pax01)
(a:ecg01 *a:recording* a:r01) (a:pax01 *rdf:type* a:P-Axis)
(a:r01 *rdf:type* a:ECGObservation) (a:pax01 *a:value* 27)

Step 2: Enrichment of the local information at the application ontology. In this step implicit information (regarding the individuals) that can be inferred from the application ontology of system A is made explicit with the help of a reasoner. For example, in this step each individual inherits a terminology code from its corresponding class:

(a:ecg01 *a:loinc* 8601-7) (a:ax01 *a:loinc* 8607-4)
(a:f01 *a:snomed* 102593009) (a:pax01 *a:loinc* 8626-4)
(a:r01 *a:loinc* 34534-8)

Step 3: Classification of the information in the canonical ontology. At this point, thanks to the equivalence, subsumption and path mappings that have been defined by the MappingModule and the help of a reasoner, the individuals are now classified as instances of the concepts of the canonical ontology. For example, given that a:ECGObservation≡∃loinc.{34534-8} and c:ECGRecording≡∃loinc.{34534-8} it is wise to think that the MappingModule will infer the equivalence mapping a:ECGObservation≡c:ECGRecording. Then, as the assertional box of the application ontology of system A contains the triple (a:r01 *rdf:type* a:ECGObservation), the new triple (a:r01 *rdf:type* c:ECGRecording) is inferred. Moreover, since triples (a:r01 *rdf:type* a:ECGObservation), (a:r01 *a:hasAxis* a:ax01) and (a:ax01 *a:hasP-Axis* a:pax01) exist, path rule $R1$ is fired and the triple (a:r01 *c:comp* a:pax01) is generated. The remaining new triples, some of which are shown next, can be figured out accordingly.

(a:ecg01 *rdf:type* c:ECGDiagnosis) (a:ecg01 *c:hasObs* a:r01)
(a:ecg01 *c:loinc* 8601-7) (a:r01 *c:comp* a:pax01)
(a:ecg01 *c:hasFinding* a:f01) (a:pax01 *rdf:type* c:P-Axis)
(a:f01 *c:snomed* 102593009) (a:pax01 *c:value* 27)
(a:r01 *rdf:type* c:ECGRecording)

Step 4 : Recognition at the receiver's ontology. The triples generated up to this moment are sent to system B and, thanks to the ontological mappings defined for this ontology by the MappingModule, the individuals will be recognized as instances of the classes of its application ontology. For example, due to (a:f01 *c:snomed* 102593009), b:snomed≡c:snomed and the definition of class b:ECGNormalFind, f01 is classified as an individual of class b:ECGNormalFind, and then, due to the definition of class b:ECGDiagnosis, now the main individual a:ecg01 is classified as an individual of class b:ECGNormalDiag :

```
<entry typeCode="DRIV">
    <organizer classCode="OBS" moodCode="EVN">
        <code code="102593009"
              codeSystem="2.16.840.1.113883.6.96"
              codeSystemName="SNOMED CT"
              displayName="Normal ECG Finding"/>
        <component>
              <observation classCode="OBS" moodCode="EVN">
                    <code code="8626-4"
                          codeSystem="2.16.840.1.113883.6.1"
                          codeSystemName="LOINC"
                          displayName="P wave axis"/>
                    <value xsi:type="PQ"
                           value="27" unit="deg"/>
              </observation>
        </component>
        <component>
            ...
        </component>
        ...
    </organizer>
</entry>
```

Fig. 2. Excerpt of the generated HL7 entry

(a:ecg01 *b:loinc* 8601-7)
(a:ecg01 *rdf:type* b:ECGDiagnosis) (a:ecg01 *b:component* a:pax01)
(a:ecg01 *b:finding* a:f01) (a:pax01 *rdf:type* b:P-Ax)
(a:f01 *rdf:type* b:ECGNormalFind) (a:pax01 *b:value* 27)
(a:ecg01 *rdf:type* b:ECGNormalDiag)

Step 5: Storage at the receiver's system: At this point, it is straightforward to store the information into the underlying repository of system B due to the Σ links that indicate how to transform the collection of triples into a suitable HL7 document (see Fig.2). Notice that since the main individual ecg01 has been recognized as of class b:ECGNormalDiag, it is possible to choose from the HL7 entry templates of system B the one which represents only information about normal ECG results –despite in the sender's system there was only one table for storing all kind of ECG diagnoses.

IV. Conclusions and discussion

We have presented a semantic-based framework which allows the interoperability of medical diagnoses between health information systems, including those which were not developed following EHR standards. The feasibility of the idea has been proved through an example. To sum up, the main features of the framework presented in this paper are the following: (1) It is extensible to both standard and proprietary models, since any healthcare institution could create its own application ontology and relate it to the terms of the canonical ontology via an integration mapping. Two modules are provided in order to help with this adaptation: one module that facilitates the task of obtaining the definitions of the application ontology from a particular underlying system and another module that facilitates the task of linking definitions of the application ontology to definitions of the canonical ontology; (2) It uses a formal ontology as canonical conceptual model, which allows to focus on semantic aspects that are independent of the languages or technologies used to describe EHRs. As a result, it is not based on peer-to-peer transformations but on the semantic acknowledgement of one instance of a class in the source ontology as instance of another class in the target ontology; (3) The features of any specific system remain unknown to the other systems in the framework. Acknowledging and using the canonical ontology as a shared model is enough; (4) Reasoning

plays a major role in several parts of the framework, which decreases the need of human intervention.

However, there are still some challenges, such as those regarding scalability, that need to be addressed in order for this approach to be accepted widely. In the case of the DB2OntoModule, the existence of the terminology manager TM is assumed. The fact that a particular term of a database has a corresponding terminological code in the TM allows a more precise definition of that term in the application ontology. We are aware that database systems may not provide with such a set of correspondences, so syntactic and semantic similarity measures (such as Levenshtein distance[6] or WordNet[7]-based similarity) between the terms in the database and those in the terminologies would have to be applied in order to obtain a set of candidate codes. Moreover, relational databases whose schema can be consulted have been chosen as underlying repositories. In the real world data can be far messier and come from unstructured or semi-structured sources. In general, the less structured the source is, the more difficult the construction of the ontology will be. In the case of unstructured sources, machine learning and text mining algorithms could be used in order to create an ontology from input documents. For semi-structured data in XML, XQuery[8] and XPath[9] could be used for extraction of relevant information, and moreover, fuzzy extensions of those languages could be used to enhance that extraction. Another technique that could be applied in semi-structured sources is ILP[28]. With respect to the core task of building an agreed canonical ontology, efforts devoted to classifications on standards (e.g. openEHR) or terminology taxonomies (e.g. SNOMED-CT) can be exploited and oriented towards the design of such an ontology. Finally, challenges concerning mappings between the application and canonical ontologies are diverse (e.g. variable granularity of the information, different types of data, etc.). In this paper some steps towards resolving mapping issues have been given via the detection of path mappings and their implementation using SWRL rules, but, as stated in section II-C, extensive work has already been made on this area, so the definition of a new approach is out of the scope of this paper. Additionally, we suggest that specific systems publish voluntarily the integration mappings between their application ontology and the canonical ontology, so that other systems could benefit from this knowledge at the time of creating their integration mapping.

ACKNOWLEDGMENT

The work of Idoia Berges is supported by a grant of the Basque Government (Programa de Formación de Investigadores del Departamento de Educación, Universidades e Investigación). This work is also supported by the Spanish Ministry of Education and Science TIN2010-21387-C02-01.

REFERENCES

[1] V. N. Stroetman(ed.), D. Kalra, P. Lewalle, A. Rector, J. M. Rodrigues, K. A. Stroetmann, G. Surjan, B. Ustun, M. Virtanen, and P. E. Zanstra, "Semantic Interoperability for Better Health and Safer Healthcare," European Commission, Tech. Rep., Jan. 2009.
[2] "openEHR," 2011, available at *http://www.openehr.org*.
[3] "ISO 13606-1: Electronic Health Record Communication Part 1: Reference Model," 2008.
[4] "HL7-CDA," 2011, available at *http://www.hl7.org*.
[5] P. Schloeffel, T. Beale, G. Hayworth, S. Heard, and H. Leslie, "The relationship between cen 13606, hl7 and openehr," in *Health Informatics Conference, HIC 2006*, Sydney, Australia, 2006.
[6] V. Kashyap and A. P. Sheth, "Semantic and schematic similarities between database objects: A context based approach," *The Very Large Databases Journal*, vol. 5, no. 4, pp. 276–304, 1996.
[7] "OWL2 Web Ontology Language," World Wide Web Consortium, 2009, *http://www.w3.org/TR/owl2-overview/*.
[8] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," *Knowledge Engineering Review*, vol. 11, pp. 93–136, 1996.
[9] T. Beale and S. Heard, "An Ontology-based Model of Clinical Information," in *Proceedings of the 12th World Congress on Health (Medical) Informatics - Building Sustainable Health, MEDINFO 2007*, Brisbane, Australia, 2007, pp. 760–764.
[10] C. Wroe, "Is semantic web technology ready for healthcare?" in *Paper presented at the 3rd European Semantic Web Conference, ESWC'06*, Budva, Montenegro, jun 2006, http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-194/paper2.pdf.
[11] J. H. Weber-Jahnke and J. Williams, "The smart internet as a catalyst for health care reform," in *The Smart Internet - Current Research and Future Applications*, 2010, pp. 27–48.
[12] R. Krummenacher, E. P. B. Simperl, D. Cerizza, E. D. Valle, L. J. B. Nixon, and D. Foxvog, "Enabling the european patient summary through triplespaces," *Computer Methods and Programs in Biomedicine*, vol. 95, no. 2-S1, pp. 33–43, 2009.
[13] O. Kilic and A. Dogac, "Achieving Clinical Statement Interoperability using R-MIM and Archetype-based Semantic Transformations," *IEEE Transactions on Information Technology in Biomedicine, to appear.*, 2009.
[14] V. Bicer, O. Kilic, A. Dogac, and G. B. Laleci, "Archetype-Based Semantic Interoperability of Web Service Messages in the Health Care Domain," *Int'l Journal on Semantic Web & Information Systems*, vol. 1, no. 4, pp. 1–22, 2005.
[15] C. Martínez-Costa, M. M. Tortosa, and J. T. Fernández-Breis, "An approach for the semantic interoperability of ISO EN 13606 and openEHR archetypes," *Journal of Biomedical Informatics*, vol. 43, no. 5, pp. 736–746, 2010.
[16] L. Lezcano, M.-Á. Sicilia, and C. Rodríguez-Solano, "Integrating reasoning and clinical archetypes using owl ontologies and swrl rules," *Journal of Biomedical Informatics*, vol. 44, no. 2, pp. 343–353, 2011.
[17] The openEHR Foundation, "Archetype Definition Language," 2007, available at *http://www.openehr.org/releases/1.0.2/architecture/am/adl.pdf*.
[18] B. Prados-Suarez, C. Molina, M. Prados, and C. Peña, "On the use of an ontology to improve the interoperability and accesibility of the electronical health records (ehr)," in *International Workshop on Semantic Interoperability, IWSI 2011*, Rome, Italy, jan 2011, pp. 73–81.
[19] R. Hedayat, "Semantic web technologies in the quest for compatible distributed health records," Department of Information Technology, Uppsala University, White Paper, mar 2010.
[20] L. González, G. Llambías, and P. Pazos, "Towards an e-health integration platform to support social security services," in *6th International Policy and Research Conference on Social Security*, Luxembourg, Luxembourg, sep 2010.
[21] "SNOMED," 2011, available at *http://www.ihtsdo.org/snomed-ct/*.
[22] "LOINC," 2011, available at *http://loinc.org/*.
[23] P.-A. Champin, G.-J. Houben, and P. Thiran, "Cross: An OWL wrapper for reasoning on relational databases," in *ER*, ser. Lecture Notes in Computer Science, C. Parent, K.-D. Schewe, V. C. Storey, and B. Thalheim, Eds., vol. 4801. Springer, 2007, pp. 502–517.
[24] J. M. Blanco, A. Illarramendi, and A. Goñi, "Building a federated relational database system: An approach using a knowledge-based system," *Int. J. Cooperative Inf. Syst.*, vol. 3, no. 4, pp. 415–456, 1994.
[25] J. M. Blanco, A. Goñi, and A. Illarramendi, "Mapping among knowledge bases and data repositories: Precise definition of its syntax and semantics," *Inf. Syst.*, vol. 24, no. 4, pp. 275–301, 1999.
[26] J. Euzenat and P. Shvaiko, *Ontology matching*. Springer-Verlag, 2007.
[27] "SWRL," 2011, available at *http://www.w3.org/Submission/SWRL/*.
[28] S. Muggleton, "Inductive logic programming," *New Generation Computing*, vol. 8, no. 4, pp. 295–318, 1991.

[6] http://www.levenshtein.net/
[7] wordnet.princeton.edu/
[8] http://www.w3.org/TR/xquery/
[9] http://www.w3.org/TR/xpath20/