Ontology-based Model for Interoperability between openEHR and HL7 Health Applications

Cristiano André da Costa, Matheus Henrique Wichman, Rodrigo da Rosa Righi
Software Innovation Laboratory - SOFTWARELAB
Applied Computing Graduate Program
Universidade do Vale do Rio dos Sinos - UNISINOS
São Leopoldo, Brazil 993022–750
Email: cac@unisinos.br, matheusw@edu.unisinos.br, rrrighi@unisinos.br

Adenauer Corrêa Yamin
Computer Science Graduate Program
Universidade Federal de Pelotas - UFPEL
Pelotas, Brazil 96075–630
Email: adenauer@inf.ufpel.edu.br

Abstract—Information Technology (IT) applied to health care can bring many benefits; however, its adoption remains low. One of the barriers that prevent IT adoption is the lack of interoperability among health care systems. Since nowadays there are many standards for Electronic Health Record (EHR) systems, methods to exchange information among them are needed. In this context, this article proposes a method to retrieve clinical data created by independent health systems using the same input query. Different from related work, the base of the proposed method is on the mapping of health record to OWL ontologies, enabling us to use rules and features from the OWL language to create interoperability equivalences between each system. Our approach was evaluated by querying an ontology fed with mixed openEHR and HL7 records. The results are encouraging, bringing the benefits of using a single entry point to reach in a transparent way different models of health applications.

Index Terms—Interoperability; Electronic Health Record; Ontology; Health Informatics.

I. INTRODUCTION

Information Technology (IT) applied to health care can bring many benefits, including supporting physicians decisions, costs saving, greater engagement of patients during their treatment and human error rate reduction during procedures. However, adoption of IT tools in health remains low, especially in underdeveloped countries [1], [2]. Several factors contribute to preventing health systems implementation. We can group these factors in human, financial and technical barriers [2], [3]. Human barriers include lack of awareness of the importance of using Electronic Health Records (EHR), no experience with computer applications, negative impressions about Health Information Systems (HIS), as well as other factors. Financial barriers are due to high initial cost to buy EHR solutions and lack of investment to infrastructure changes, support, training, and maintenance. The technical barriers are related to the system implementation, like complicated interfaces, no adaptation to use mobile devices and mainly the absence of interoperability between other systems in use (e.g., disease-specific surveillance systems).

Some efforts have been made to address the interoperability issue [4], [5]. Initially, a possible solution comprises the use of defined sets of electronic messages, transmitted using EDIFACT\(^1\) or HL7\(^2\). [6] These messages were used to support service administration, billing, communication and to help in public health measures. However, few messages have been developed to treat the health care process itself, and the existing ones proved not to be flexible enough to represent more specific cases. Given the limitations with EDIFACT messages approach, some efforts were conducted to create an EHR in which data entry is standardized and the information offered is complete, comprehensive, unambiguous and linked to other sources [5].

A dual-model architecture has been adopted to satisfy modern EHR requirements [6]. Following this approach, generic properties from the health record are moved to the Reference Model (RM) while specific information needed by each profession, specialty or service is described through Archetypes [6]. This architecture model is a requirement to solve the widely acknowledged challenge of semantic interoperability, which is the ability of different clinical systems to share health record data while preserving their meaning faithfully. The dual-mode architecture helps to achieve this kind of interoperability by making explicit to other systems the underlying structure, through the Reference Model. Currently, the dual architecture is being used in the design of openEHR Information Architecture, CEN 13606 specification, and HL7 Templates.

People frequently change their health care provider from time to time, and this transition can lead to a fragmented EHR since different providers may use diverse systems causing the distribution of clinical data among several databases [4]. By allowing one HIS to interoperate with another HIS permits to keep the patient EHR consistent during health care provider changes. Also, having access to the entire patient history can help physicians to investigate diseases causes and to identify patterns by analyzing old observations. Although clinical data sometimes is shared during health provider changes, it can come in a different format, preventing this data from being merged with the EHR in the new health care provider system.

In this context, this article proposes a method to retrieve

\(^1\)http://www.unece.org/cefact/edifact/welcome.html
\(^2\)http://www.hl7.org
clinical data created by independent health systems using the same query. To accomplish that, we first mapped health records to ontologies [7]. In particular, we used the Ontology Web Language (OWL) \(^3\) to make explicit concept equivalences in each system. From this point, using a reasoner, we can fetch clinical data from different systems through a single SPARQL query. To evaluate the proposed method, we retrieve blood pressure observations from an openEHR health record and an HL7 health record, taking as reference the openEHR specification. Thus, the proposed method can be extended to retrieve any EHR structure.

The remainder of this article is organized as follows. In Section II we present our method to map health records to ontologies. Section III explain how equivalences are created in OWL. Section IV presents results from the experiment we made. Section V discuss similar approaches. Section VI draws some conclusions and future work.

II. MAPPING HEALTH RECORDS TO ONTOLOGIES

To store health records in a knowledge base, we have to translate the EHR to an ontology. With ontologies, instances from the health record are represented by individuals of a type corresponding to OWL classes. The conversion process has to take into account the architecture of the EHR system used: when clinical concepts and commons structures share the same layer (single-model architecture) or are in separate levels (dual-model architecture). Despite differences between architectures, both models can be expressed using the Unified Modeling Language (UML), making clear the concepts and relations between them\(^8\). Having the UML representation, the methodology developed by Noy and McGuinness \([9]\) can be applied to generate the ontological representation for each UML diagram. In that case, we created OWL classes following the UML representation. The ontology terms were as close as possible as the ones used in UML. Moreover, we defined cardinality and property ranges by creating property restrictions on the OWL classes.

A. Single-model Health Records

EHR systems based on a single-model architecture (i.e., HL7) require only a set of ontologies to represent the health record semantically. The ontology we need derives from the HL7 Reference Information Model (RIM) which is the foundation of the HL7 standard. The HL7 RIM is composed of five core classes: Act, ActRelationship, Participation, Role and Entity. Other classes extend these to specialize in concepts. The RIM ontology will use these terms as top-level classes. After the conversion of an EHR system to an ontology, individuals are created to store the data.

Some efforts have been made to translate the RIM specification to an ontology. Initially, Bhavana Orgun has developed an RDFS ontology following the HL7 Version 3. This ontology traced important concepts and properties from the HL7 specification. Afterward, Helen Chan\(^4\) extended Orgun’s work by covering a broader range of artifacts from Version 3. Also, Zhanga et al. \([10]\) have translated the HL7 RIM to an ontology. Zhanga approach differs from others because in that case RIM concepts were grouped in three root classes: (1) Model - representing the six backbone classes of RIM -, (2) Data - grouping standard data types -, and (3) Expression - designed to fill the gap between static knowledge and that one generated by inferences.

The ontology used here derives from both Chan’s and Zhanga’s approaches. We extracted terms from HL7 Version 3 specification, and corresponding OWL classes created following the original hierarchy, as Figure 1 shows. The OWL classes created were distributed in two other ontologies: RIM ontology to group core classes and Data Types ontology accommodating data types definitions. The group Expression was not used since it is not necessary to represent rules as individuals.

B. Dual-model Health Records

When the organization of an EHR system follows the dual-model architecture (i.e., openEHR), it requires both RM and archetypes translation. The translation of a dual-model EHR system generates one or more ontologies representing the RM and another group of ontologies for each archetype employed. In that case, individuals will be distributed across ontologies. The translation of RM follows a similar approach to that one used in HL7. The openEHR RM is divided into five documents (i.e., data types, data structures, support, among others). Thus

\(^3\)https://www.w3.org/TR/owl-guide

\(^4\)https://www.w3.org/wiki/HCLS/ClinicalObservationsInteroperability/HL7CDA2OWL.html
the translated version is composed of five ontologies, each one comprising one single document. Since Román et al. [11] have already translated the RM, we extended it by revising the existing ontologies and making them up-to-date to the latest openEHR RM version. Figure 2 shows the hierarchy from the OWL version of openEHR RM.

Although OWL language has not the concept of generic types, like many programming languages, we used subclasses as an alternative to properly represent the restriction imposed by a generic type. The openEHR RM has several classes, like VERSIONED_OBJECT<T>, where their behavior depends on the T type they are locked. These classes are represented in OWL by creating a class where the ranges of those properties, depending on the T type, as being owl:Thing. Later, possible values to that generic type are represented through subclasses in which constrains the ranges to their final types. In our example, the data property from VERSIONED_OBJECT_T class allows any descendants of owl:Thing as value. However, VERSIONED_COMPOSITION class overrides that restriction by allowing only COMPOSITION individuals.

The translation of openEHR archetypes to OWL is based on the method presented in Leonardo Matías Ph.D. thesis [12]. Matías approach targets the definition of archetypes. However, the current implementation is only capable of translating a limited set of RM structures, so we extended it by covering more structures. We also developed another translator to convert instances of archetypes to OWL individuals.

The archetype translator begins by taking the definition of the archetype written using the Archetype Definition Language (ADL). For example, to translate the definition of a blood pressure archetype (showed briefly in Figure 4) the translator would extend the RM class OBSERVATION and map ADL restrictions to OWL property restrictions, forcing property ranges to only allow instances from classes extended within the archetype ontology instead of their original ranges. This process is repeated for each level, recreating the archetype hierarchy. The node identifier (e.g. at0001) is represented in the translated ontology through an OWL annotation property NodeID. These identifiers help to create a precise mapping between the ADL and the ontology created. Figure 3 shows the resulting ontology from the archetype translation. As mentioned earlier, the method extended the OBSERVATION class creating the Blood_Pressure class, whose name was extracted from the comment in each ADL node. That class contains a restriction on its data and protocol properties allowing only instances from the extended version of HISTORY and ITEM_TREE classes.

The other translator we developed takes as input an instance of an archetype, or any other RM structure, then convert it to OWL individuals. The instance translator sends the instance through a series of serializers, where each one is responsible for the translation of a type of the RM structure. Once found the serializer for the type being convert, then it creates a triple for every property from the instance. If the property value is a primitive data type (i.e., number, date, string, etc.) it is mapped to an equivalent data type in OWL. Otherwise, when the property value is an instance of another type, the
The Semantic Web Rule Language - SWRL [13] was designed to increase the expressivity of an ontology through a set of inference rules. It works by defining an antecedent and a consequent part which if conditions specified in the former one are correct than the conditions from the latter one must also be true. We can use SWRL rules to increase the expressivity of the HL7 ontology by classifying individuals based on its properties, allowing to define a mapping with the openEHR ontology. An approach to achieve the mapping between blood pressure observations from openEHR and HL7 is to make Observation individuals to also belong to a custom OWL class, like Blood_Pressure_Observation, created only for the mapping.

A point of departure to develop the SWRL rule needed is to investigate how an Observation is used to store blood pressure records. The Observation class is a specialization of the Act class so it inherits the attributes moodCode and code. The moodCode attribute defines the stage of the act, specifying whether the act is an activity that has happened, can happen, is happening, is intended to happen, or is requested/demanded to happen, while the code attribute determines the clinical concept represented by the act by linking its value to an external coding system, like LOINC terminology. [14] Figure 6 illustrates the attributes of an instance of Observation recording blood pressure.

As shown, the moodCode attribute receives a CS object containing the code for an event that already has happened, the EVN code, while to the code attribute is assigned a CS object with the LOINC code for Blood Pressure. The SWRL rule of Figure 5 checks the values of these attributes so when they occur then the individual receives the OWL class Blood_Pressure_Observation.

After the classification of individuals through the SWRL rule, we have a one-to-one mapping between openEHR and HL7. Thereat, we can use the owl:equivalentClass property to connect the openEHR’s Blood_Pressure class with the HL7’s Blood_Pressure_Observation class, creating an equivalence between these two EHR standards.

IV. Evaluation

The method presented here attempts to address the interoperability issue between different EHR standards by using features from the OWL language. To evaluate it, we begin with the problem that physicians sometimes need to investigate the entire EHR of a patient. However, due to several reasons, this EHR can be fragmented across several institutions and in different formats, making necessary the development of specific queries for each EHR system. Given that scenario, the evaluation of our approach pretends to verify if the SWRL rules and the OWL equivalence constructs are enough to allow a single query to be used to fetch data from different systems. We carried the evaluation by mapping the data to individuals in each EHR system ontology and the development of the query to fetch these individuals. If the query fetches data from a specific patient, we also need to establish a uniform way to
identify patients since EHRs have different ways to express to whom the data is related.

The data for the evaluation is composed of blood pressure and heart rate readings of three patients extracted from the MIMIC-III [15] database. After extraction, the data was normalized and divided into two tables: one for blood pressure and another for heart rate. Table I shows the blood pressure readings and Table II contains heart rate measurements extracted. Diastolic and systolic values are both in millimeters of mercury (mmHg) unit. We choose to represent the first five rows of each table in openEHR while the rest we represented in HL7.

<table>
<thead>
<tr>
<th>Patient ID</th>
<th>Date</th>
<th>Diastolic</th>
<th>Systolic</th>
</tr>
</thead>
<tbody>
<tr>
<td>188</td>
<td>2161-07-02</td>
<td>60</td>
<td>117</td>
</tr>
<tr>
<td>188</td>
<td>2161-07-03</td>
<td>68</td>
<td>130</td>
</tr>
<tr>
<td>188</td>
<td>2161-07-03</td>
<td>61</td>
<td>127</td>
</tr>
<tr>
<td>711</td>
<td>2185-03-22</td>
<td>94</td>
<td>152</td>
</tr>
<tr>
<td>711</td>
<td>2185-03-22</td>
<td>73</td>
<td>130</td>
</tr>
<tr>
<td>711</td>
<td>2185-03-23</td>
<td>73</td>
<td>121</td>
</tr>
<tr>
<td>1709</td>
<td>2118-01-04</td>
<td>102</td>
<td>168</td>
</tr>
<tr>
<td>1709</td>
<td>2118-01-04</td>
<td>88</td>
<td>120</td>
</tr>
<tr>
<td>1709</td>
<td>2118-01-04</td>
<td>81</td>
<td>136</td>
</tr>
</tbody>
</table>

**TABLE I**

BLOOD PRESSURE READINGS

<table>
<thead>
<tr>
<th>Patient ID</th>
<th>Date</th>
<th>Heart Rate (BPM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>188</td>
<td>2157-01-12</td>
<td>75</td>
</tr>
<tr>
<td>188</td>
<td>2157-01-12</td>
<td>76</td>
</tr>
<tr>
<td>188</td>
<td>2157-01-12</td>
<td>81</td>
</tr>
<tr>
<td>711</td>
<td>2184-05-26</td>
<td>91</td>
</tr>
<tr>
<td>711</td>
<td>2184-05-26</td>
<td>76</td>
</tr>
<tr>
<td>1709</td>
<td>2115-04-28</td>
<td>79</td>
</tr>
<tr>
<td>1709</td>
<td>2115-04-28</td>
<td>87</td>
</tr>
<tr>
<td>1709</td>
<td>2115-04-29</td>
<td>75</td>
</tr>
</tbody>
</table>

**TABLE II**

HEART RATE READINGS

After preparing the data, we create manually the individuals for each record using the Protégé tool [16]. We placed the individuals in an ontology repo which acts as a repository for all the health records, as illustrated in Figure 7. That ontology imports the OWL version of HL7 (as shown in Figure 1) and the ontologies corresponding to the openEHR RM. To represent blood pressure and heart rate readings we translated to OWL the archetypes openEHR-EHR-OBSERVATION.blood_pressure.v1 and openEHR-EHR-OBSERVATION.pulse.v1, both are available on the openEHR Clinical Knowledge Manager - CKM [17]. The resulting ontologies from the archetype translation were also imported into the repo ontology.

Before the owl:equivalentClass property be used, the SWRL rule to classify HL7 individuals must be executed. The rule for blood pressure observations is the same to the one represented in Figure 5 and for heart rate readings a very similar rule is used but checking for the appropriated
LOINC code and assigning the custom OWL class Heart_Rate_Observation instead. With individuals classified, the owl:equivalentClass property can be used to connect blood_pressure:Blood_Pressure class with the BloodPressure_Observation class or the pulse:Heart_Heart_beat class with Heart_Rate_Observation class.

Although is possible to fetch records from different sources using only the owl:equivalentClass property, physicians frequently need to query a single patient but HL7 and openEHR use different methods to identify patients. Given that problem, we could use again SWRL rules to create a uniform means to identify the patient whether openEHR or HL7 is used. The openEHR stores the identifier of the patient in the subject property of the ENTRY class, parent of OBSERVATION. This identifier is an instance of the wrapper class PARTY_IDENTIFIED or PARTY_SELF. The identifier itself is recorded in a way which depends on the wrapper class used, since we used PARTY_IDENTIFIED class the identifier is an instance of DV_IDENTIFIER class added to the identifiers list. Figure 8 gives an example of rule to extract that identifier from the subject property and copy it to the data property patientID.

ehr:ENTRY(?e) ∧ ehr:subject(?e, ?pid) ∧ cprm:identifiers(?pid, ?idf) ∧ datatypes:id(?idf, ?id) ⇒ patientID(?e, ?id)

Fig. 8. Rule to add the patientID property to openEHR individuals

HL7 follows a similar approach to the one used in openEHR. However, the patient identifier is stored together with other personal identifiers, like the identifier from the doctor, the nurse or any other person involved. This structure makes it hard to develop a rule covering all representations of the patient identifier since there is a large number of fields in which the rule has to check. In Figure 9 there is an example of rule to extract the identifier and put it in the patientID property. The observations used in our evaluation only contains identifiers of the patient; therefore in real situations, this rule should be expanded.


Fig. 9. Rule to add the patientID property to HL7 individuals

Through the patientID property, now we have a uniform way to identify the patient, not mattering the EHR standard used. This property and the equivalences created earlier allows us to query data distributed in different formats but using a single SPARQL query, like Figure 10 demonstrates. In this example, the query fetches all blood pressure observations for all patients and, as expected, both records from HL7 as well as from openEHR are retrieved.

To retrieve data for a specific patient, we could use the value of patientID property to filter them. Figure 11 gives an example of a query to fetch observations from just one patient (e.g., the patient 711). As explained, this patient contains readings represented through the HL7 but also others in openEHR. Because equivalences we created, both types are returned.

V. RELATED WORK

Some efforts already have been made to address the interoperability between heterogeneous health systems. The work done so far is based on the mapping of the health records to ontologies and with the help of custom tools information are correlated.

Bicer et al. [18] attempts to create a mapping between HL7 v2 and HL7 v3 messages. Since version 2 of HL7 uses plain-text messages while version 3 is based on the RIM specification, the mapping to ontologies helps to create a consistent representation of both types. An OWL mapping tool called OWLmt was developed to help the mapping.

Martínez-Costa el al. [19] focus on the interoperability between archetypes from openEHR and ISO EN 13606. As both
standards follow the dual model architecture, the solution presented is based on the transformation of openEHR archetypes into ISO EN 13606 and vice versa. The conversion is made by combining Semantic Web with Model-driven Engineering technologies.

More recently, Roehrs et al. [4] propose a model for interoperability focused on Personal Health Records (PHR). Their proposal is focused on the use of Natural Language Processing (NLP) techniques for converting different formats of data, including openEHR and HL7. As a final result, they also store the data in an ontology.

VI. CONCLUSION

The IT applied to health care has the potential to improve substantially the group of processes involved in this area. These improvements would vary from better resource management, better support in decision making and, the most important, improvement in the quality of the treatment for patients. To accomplish these goals, the information exchange between health systems is essential to offer to physicians, at an acceptable speed, all the data they need.

This article presented the proposition of a method to reach the interoperability between heterogeneous health systems by the use of ontologies and rules. Our approach allows retrieving data from different health care systems at once, only using a single query. This is allowed because we use the OWL features to map equivalences between standards. Our approach is divided into two steps: (1) the translation to OWL of the HIS structure and (2) the process of creating bindings between similar structures of each system. To create the bindings, we use the construction of equivalences provided by OWL, in addition to SWRL rules when these constructs cannot be used at first. These bindings are used by the reasoner to infer additional knowledge not explicit in the ontology. Although this paper focuses on the data extraction from health systems, we could apply the proposed method presented to many scenarios involving systems not designed to interface with each other.

After the evaluation, the method showed viability to use ontologies to retrieve information from different sources at the same time. However, due to the semantic meaning of equivalence constructs, the reasoner deduced some incorrect assertions which could lead to wrong information being retrieved when the query differs from the initial purpose. In our view, this limitation can be overcome by the creation of rules and further specification of the repo ontology.

As future work, we plan to extend the interoperability to additional standards. We also plan to evaluate our proposition with huger sets of data, including anonymized data of a partner hospital. Finally, we plan to extend the ontology to cover additional data on electronic health records.

ACKNOWLEDGMENTS

This work was supported in part by the Coordination for the Improvement of Higher Education Personnel - CAPES (Finance Code 001) and the National Council for Scientific and Technological Development - CNPq (Grant Numbers 303640 / 2017-0 and 405354 / 2016-9).

REFERENCES