MEDINFO 2007
K. Kuhn et al. (Eds)
IOS Press, 2007

Framework for Clinical Data Standardization Based on Archetypes

Jose A. Maldonado[a], David Moner[a], Diego Tomás[a], Carlos Ángulo[a], Montserrat Robles[a], Jesualdo T. Fernández[b]

[a] *Biomedical Informatics Group, ITACA Institute, Technical University of Valencia, Spain*
[b] *Departamento de Informática y Sistemas, University of Murcia, Spain*

Abstract

Standardization of data is a prerequisite to achieve semantic interoperability in any domain. This is even more important in the healthcare sector where the need for exchanging health related data among professional and institutions is not an exception but the rule. Currently, there are several international organizations working on the definition of electronic health record architectures, some of them based on a dual-model approach.

We present both an archetype modeling framework and LinkEHR-ED, an archetype editor and mapping tool for transforming existing electronic healthcare data which do not conform to a particular electronic healthcare record architecture into compliant electronic health records extracts. In particular, archetypes in LinkEHR-ED are formal representations of clinical concepts built on a particular reference model but enriched with mapping information to data sources which define how to extract and transform existing data in order to generate standardized XML documents.

Keywords:

medical records, archetypes, standardization, data translation, information systems

Introduction

Health care is a sector where the need for sharing information is the norm rather than the exception. However, the health data of one patient is usually scattered among the different health facilities where she/he has been attended. This leads to distributed and heterogeneous data resources, all of them containing health data, making the exchange of data across systems and organizations very difficult. This situation has created a large gap between the potential and actual value of the information content of electronic health records (EHR). Closing this gap by making efficient use of the health data held by these systems, could improve significantly patient care, clinical efficiency and empower research activities.

Due to the special sensitivity of medical data and the wide range of ethical and legal constraints, data exchange must be done in a meaningful way, avoiding all possibility of misunderstanding or misinterpretation. The faithful communication in EHR crucially depends on the standardization of the EHR architecture (EHRA) used to communicate data. Currently there are several international organizations [1] working on the definition of an EHRA. Health Level 7 [2] supports two message protocols: HL7 Version 2 and HL7 Version 3. The technical committee 251[3] of the European Committee for Standardization is working on a new full European Standard and future ISO norm for the communication of the EHR called EN13606 [4][5]. The OpenEHR foundation [6] maintains an EHR architecture designed to support the constructions of distributed, patient-centered, life-long, shared care health records.

Due to the complexity and the continuous evolution of the health domain a new approach for the development of EHR systems has been proposed. The methodology is known as the dual model methodology which is based on a clear separation between information and knowledge. The former is described through a Reference Model (RM) that contains the basic entities for representing any entry in an EHR. The latter is based on archetypes, which are formal definitions of clinical concepts in the form of structured and constrained combinations of the entities of a RM. Examples of Dual Model EHRA are CEN/TC251 EN13606 and openEHR.

Currently, in most organizations there is a vast amount of health related data that does not conform to an EHRA that need to be converted into standardized EHR extracts in order to be exchanged with other organizations. In this paper we address the problem of how to use archetypes to make public these legacy data in the form of standardized EHR. We argue that archetypes are suitable for this purpose. First, archetypes allow the formal description of the semantics of legacy and un-standardized health data. On the other hand, by mapping the structure of archetype definitions to the elements of the data sources, it is possible to generate normalized EHR extracts compliant with the underlying RM.

Dual model approach

The dual model approach distinguishes a reference model and archetypes.

A reference model is an object oriented model that is used to represent the generic and stable properties of health record information. It comprises a small set of classes that define the generic building blocks to construct EHRs. It

specifies how health data should be aggregated to create more complex data structures and the context information that must accompany every piece of data in order to meet ethical and legal requirements.

An archetype is a formal definition of a distinct, domain-level concept in the form of structured and constrained combinations of the classes of the RM. What is important is that for each concept in the domain we want to use, a definition can be developed in terms of constraints on structure, types, values, and behaviors of RM classes. Basically, archetypes are means for providing semantics to data instances that conform to some reference model by assuring that data obey a particular structure (combination of classes of the reference model) and satisfy a set of semantic constraints. This is achieved by linking data structures and content to knowledge resources as terminologies and ontologies. Their principal purpose is to provide a powerful, reusable and interoperable way of managing the creation, description, validation and query of EHRs.

ADL (Archetype Definition Language) [7] is a formal language for expressing textually archetypes developed by OpenEHR that has also been adopted by CEN. ADL is a textual language for specifying constraints on data instances of an RM in a formal way. An archetype expressed in ADL is composed of four main parts: header, definition, ontology and revision history. The header section contains the archetype metadata. In the definition section is where the modeled clinical concept is represented in terms of a particular RM class. This description is built by constraining several properties of classes and attributes, such as existence, occurrences or cardinality or by constraining the domain of atomic attributes. It is important to notice that in this section only those entites that need to be constrained should appear. The ontology section is where the entities defined in the definition section are described and bound to terminologies. Finally the revision history section contains the audit of changes to the archetype.

Archetype modeling

The current ADL specification is not precise enough regarding archetype specialization; this hinders a precise understating of archetypes and their implementation. As a consequence our first work was to define a precise archetype modeling framework as a prerequisite for implementing tools providing enhanced support for archetypes. Since our main concern is the generation of standardized EHR extracts we have focused on the data definition facet of archetypes. We view the definition section of archetypes as a database schema which describes subset of instances of a class from a particular RM. In this section we present briefly the data model used to describe data instances (EHR extracts) and the schemas that formalize the definition section of archetypes, the composition and specialization of archetypes and the relationship between a business concept and its archetypes and between an archetype and its instances. An extended formal definition and additional examples can be found in [8], due to lack of space these are omitted here.

Data model

Archetypes impose a hierarchical structure to the EHR, therefore we have chosen a data model based on trees with labeled nodes to formalize their data instances. It is similar to the models presented in [9,10] but our data model supports both ordered and unordered nodes. Although archetypes do not impose an order on class attributes or attribute values, it is possible to define ordered multi-valued attributes such as attributes whose value is a list.

The representation of data instances (in our context EHR extracts compliant with a RM) is straightforward. Each object is described by a data tree. The root node is labeled with the class name and has one child for each attribute. The children are labeled with the attribute names and each of them has one child labeled with the corresponding type (class) name. This mechanism is repeated iteratively. Atomic values are represented by a leaf node labeled with a value.

Schema model

For the representation of the definition section of archetypes we have developed a type system that allows the specification of decidable sets of data trees, i.e., in our context set of instances of the RM. We assume the existence of a finite set of primitive types C, i.e., the set of primitive types defined in the reference model, an infinite set of type variables T disjoint with C and an infinite set P of label predicates.

Definition 1. A multiplicity list is a regular expression of the form:

$$t_1^{(l_1:u_1)} \ldots t_n^{(l_n:u_n)} \tag{1}$$

Where n≥1, t_iÎCÇT, l_i£u_i and $t_i^{(l_i:u_i)} = t_i^{l_i} \mid t_i^{l_{i+1}} \mid \ldots \mid t_i^{u_i}$

Definition 2. A constrained multiplicity list (CML) is language definition expression of the form:

$$\left(t_1^{(l_1:u_1)} \ldots t_n^{(l_n:u_n)}\right)^{|l:u|} \tag{2}$$

Where $t_1^{(l_1:u_1)} \ldots t_n^{(l_n:u_n)}$ is a multiplicity list, l≤u,

$$\sum_{i=1}^{n} l_i \geq u \quad \text{and} \quad \sum_{i=1}^{n} u_i \geq l \; .$$

Intuitively the language generated by a constrained multiplicity list is composed by all the word defined by the regular expression whose length (number of symbols) is between l and u inclusively. As an example consider the CML $\left(A^{(1:2)} B^{(1:3)}\right)^{|3:4|}$ which defines the language {ABB, ABBB, AAB, AABB}.

We formalize the definition section of archetypes as a set of type definitions (a schema).

Definition 3. A type definition has either the form t:=l_t<r_t> or t:=l_t{r_t}. Where t is a type name, l_t is a label predicate and r_t is a CML over the set the type names and primitive types.

A type definition has two parts. The first is a label predicate that describes the valid labels of nodes. The second is a CML that is used to describe the sequence of children that a node may have. An expression of the form $t:= l_t<r_t>$ defines a sets of data trees t whose root node is labeled with a label that satisfies l_t, its children are ordered and are describe by r_t. On the other hand, an expression of the form $t := l_t\{r_t\}$ defines a set of data trees t whose root node is labeled with a label that satisfies l, its children are unordered and at least one permutation is described by r.

Definition 4. A schema is a set of type definitions, one of which must be declared to be the root type, i.e. whether roots of data trees can be assigned this type.

We need to define the semantics of a schema, i.e. the set of data tree that it models. Intuitively, a data tree D conforms to a schema S if it is possible to assign to every node d of D a type T_i from S, and d satisfies the label predicate and the CML of T_i.

Both RM and definition sections of archetypes can be modeled as a schema. Each type in the schema represents an RM/archetype entity, i.e. a class or an attribute, and the CML represent its structure. The label predicate describes the name of the entity or the valid domain of atomic attributes. As an example consider the class *PERSON* from Figure 1. It can be formalized by the type definition:

$$t_{\text{PERSON}} := \text{is\_}\ person(X)\left\{\left(t^{(1:1)}_{addresses}\, t^{(1:1)}_{name}\right)^{|2:2|}\right\}$$

where is_person(X) is a unary predicate that is true only when X is equal to the string "person" and $t_{addresses}$ and t_{name} are the types that, respectively, model the attributes addresses and name.

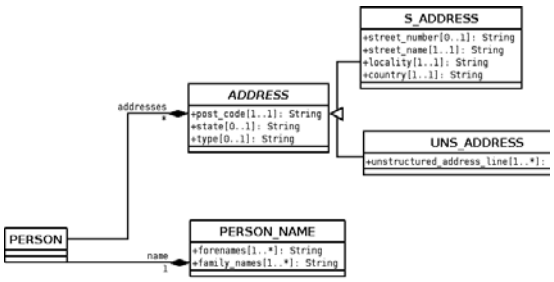

Figure 1 - Example of OO model

Let see how different types of archetype constraints are modeled with our type system. Existence constraints of attributes can be easily expressed by the CML attached to the type that models the attribute. Cardinality and occurrence constrains are mutually related. Actually, it only makes sense to constrain the occurrence of node objects that are inside a block introduced by a multi-valued attribute. Cardinality constraints can be modeled by the length constraint of CML while occurrence constraints by regular expressions. Alternative constraints for mono-valued attributes can be expressed by a CML whose length constraint is equal to |0..1| if the attribute is optional or to |1..1| otherwise, and the regular expression is of the form $t_1^{(0:1)}\ldots\ t_n^{(0:1)}$ where $t_1,..t_n$ are the alternative types. Finally, domain constraints on primitive types are straightforwardly modeled as label predicates. As an example, let us consider the following ADL expression:

```
PERSON[at001]  matches {
    addresses cardinality matches {1..*} matches {
        ADDRESS[at002] occurrences matches {0..1}
            matches {...}
        ADDRESS[at003] occurrences matches {0..*}
            matches {...}
}
```

Which can be modeled by the following set of type definitions:

$$PERSON[at001] := is\_PERSON(X)\left\{\left(t^{(1:1)}_{addresses}\right)^{|1:1|}\right\}$$

$$t_{addresses} := is\_addresses(X)\left\{\left(t^{(0:1)}_{ADDRESS[at002]}\, t^{(0:*)}_{ADDRESS[at003]}\right)^{|1:*|}\right\}$$

$$t_{ADDRESS[at002]} := is\_ADDRESS(X)\{...\}$$

$$t_{ADDRESS[at003]} := is\_ADDRESS(X)\{...\}$$

Archetype specialization

Archetypes can be defined by further constraining other archetype, i.e., by specialization, in order to obtain a more adequate or fine grained representation of a clinical concept. An archetype is specialized by providing narrower constraints on data. The overall idea is that all data instances that conform to the more specialized archetypes also conform to the more general but there can be data instances of the more general archetype which are not data instances of the specialized archetype. Multiple inheritance is not allowed. We formalize the inheritance relationship between archetypes by means of the subsumption relation [10], i.e., an archetype A is more general than archetype B if A subsumes B. The subsumption relation is also used to formalize the relationship between business concepts and archetypes, we say that an archetype A specialized a business concept B if B subsumes A. What makes the proposed subsumption relation interesting is that it not only captures the containment relationship between the set of data instances defined by two archetypes, but also captures some of the structural relationship between node objects from both archetypes.

Intuitively, subsumption is based on defining a mapping function () between types, on inclusion between both label predicates and languages defined by the CML of these types. All these can be translated to archetype specialization. Subsumption mappings specify specialization relationships between node objects and attribute object from parent and child archetypes. This is compatible with the syntactical rules used in ADL to specify the specialization of node object. In ADL specialization of coded concepts, e.g. PERSON[at0001], is indicated by using the same root, followed by an extension, e.g. PERSON[at0001.1]. Note that this defines partly the subsumption mapping, e.g. (PERSON[at0001.1]) = PERSON[at0001]. On the other hand,

inclusion of label predicates assures both that only class and attribute names from the reference model are used in archetypes, and the domain of atomic attributes in specialized archetype are a subset of the domain in the parent archetype. Finally, CML controls that super-types and subtypes have the same structure.

As stated before, RM and the definition section of archetypes can be modeled by the proposed type system. One interesting consequence is that "archetypable" classes of RMs can be considered as archetypes (from a data definition point of view). Therefore, the same logic can be applied both to the specialization of an exiting archetype or to the definition of a new one by constraining a RM class. This brings about the possibility of building flexible archetype editors capable of working with several RM or different version of the same reference model. This possibility has been explored in LinkEHR-Ed.

Mapping and data translation

Since the health data to be made public resides in the underlying data sources, it is necessary to define some kind of mapping information that links entities described in the archetype to data elements in data repositories (e.g. tables and attributes in the case of relational data sources). In the health care domain very few generic EHR data transformation efforts exist. Commercial tools with limited capabilities exist but they are mainly focused on the generation of HL7 v2.x or EDI messages and none of them supports archetypes. Furthermore, definition sections of archetypes can not be represented by XML schemas. Thus, current tools for data translation for XML schemas can not be used for this purpose.

In our scenario an archetype is considered to be a view that provides abstraction in interfacing between the data sources and the RM used to communicate the EHR extracts. Since EHR extracts have an inherent hierarchical structure, we have chosen XML as canonical data model, i.e. data sources are viewed a XML documents.

There exists two kinds of mappings: atomic attribute mappings and class mappings. Atomic attribute mappings define how to obtain a value for an atomic attribute of an archetype by using a set of values from the data sources. For this purpose rules relating a set of source paths to an archetype path identifying an atomic attribute are used. This kind of mappings preserves node paths from the root, i.e. the node contexts. It is possible for an archetype attribute to have more than one mapping and it also possible to utilize functions involving more than one source path (for instance the addition of the value of two source attributes). A rule may also contain a condition specifying the subset of values of the data source that can be used to compute values. Obviously, there must be at least one of this kind of mapping for each mandatory atomic attribute.

On the other hand, for each constrained class there exists a class mapping which contains both the query to be used to retrieve all the data necessary for generating data instances and the set of attributes that identify univocally the class instances. The combination of both components allows the conversion from source data to XML documents compliant with the RM. The query extracts the relevant information and for each different combination of values of the identification attributes a new instance of the class is generated.

Archetype designers are responsible of defining the atomic attribute mappings and the system tries to generate [11] from them a set of candidate class mappings by taking into account the structure of the RM entity being used, the archetype constraints and the integrity constraints of data sources. This approach alleviates the work of defining how to populate archetypes since it is easier for the designer to indicate which data elements of the data sources are relevant to a certain archetype attribute, rather than to specify the possible complex query required to extract and transform all the relevant information. As a result, an XQuery expression is generated, which transforms the XML view of the data source into an XML document that satisfies the constraints imposed by the archetype and at the same time is compliant with the RM.

Results

LinkEHR-Ed (http//pangea.upv.es/linkehr) is a visual tool implemented in Java under the Eclipse platform which allows the edition of archetypes that can be based on different RMs, mapping specification between archetype and data sources and the generation of data conversion scripts which generate XML documents compliant with the RM. Figure 2 describes the overall edition process. LinkEHR Ed is composed of four main components.

Reference model manager. In LinkEHR-Ed new reference models expressed as a W3C XML Schema can be imported at any time. Note that the XML Schema is supposed to describe the XML documents compliant with the RM. The import process generates archetype representations of the RM archetypable classes that will be immediately available as basis for the creation of archetypes by means of specialization. Currently, many of the characteristic of W3C XML schemas are supported, such as all the data types, name spaces, imports and includes (reference models can be defined by several files) and several structures such as complex and simple types, elements, attributes, inheritance by extension and restriction, sequence, choice, all, attributes, patterns and groups and their respective facets. Two reference models have been imported and used successfully: EN13606 and OpenEHR. The XML schema of EN13606 schema has been developed by us from the UML model due to the lack of an official one. For OpenEHR the official schema (actually a set of XML schemas) has been used [6]. This feature has been very useful in keeping in pace with their evolution without modifying a single line of code.

Semantic validation module. In LinkEHR-Ed only the logic that guides archetypes specialization, which is based on the subsumption relationship described before, is hard coded. This module, given an archetype, tests whether the archetype is valid with respect the RM entity that it constrains (i.e., it is subsumed by the RM entity) or in the case that there exists a parent archetype it test that its constraints are narrowed that those of the parent archetype (i.e., it is subsumed by the parent archetype).

Mapping module. It is in charge of managing data sources schemas, attribute mappings and the generation of candidate class mappings. Given a candidate class mapping, it generates an XQuery expression that outputs standardized XML EHR extracts satisfying all the constraints stated in the archetype.

Visual interface. LinkEHR-Ed provides two different interfaces, one for the health domain experts and one for the information technologies professionals. On the one side, the Health domain expert will be in charge of archetype definition and so they must have some knowledge about the RM they are working with. But the main idea for this perspective is to hide the underlying complexity of the system and the Dual Model architecture logic involved in the designing of an archetype as we can not presuppose any computer management skills for this expert. A set of available constraints and applicable restrictions is provided during design time. On the other side is the information technologies expert, who knows the structure of data sources of the organization and his role is to map the archetype definition tree nodes to them. A mapping definition interface fills nearly all this edition perspective. It is composed by a graphical representation of the archetype definition tree and a graphical representation of the diverse data sources available. Users can then add or modify mapping transformations between elements of both representations. In this case, LinKEHR-Ed can be seen as visual mapping and data translation tool.

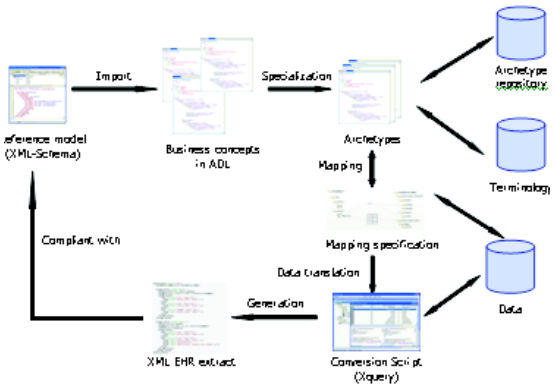

Figure 2 - Archetype edition and mapping process

Conclusion

In this paper, we have presented an archetype modelling framework and LinkEHR-Ed, a tool that allows the utilization of archetypes for upgrading already deployed systems in order to make them compatible with an EHR architecture standard. The overall objective is to maintain in-production systems and applications without any changes while providing a means for making public clinical information in the form of standardized EHR extracts, hiding technical details, location and heterogeneity of data repositories. Therefore, we use archetypes as a semantic layer over the underlying databases associating them with domain specific semantics. LinkEHR-Ed combines in an easy manner the formal representation of knowledge of a heath domain expert, represented by an archetype, with the mapping information to clinical data sources for semantic interoperability and standardization purposes.

Acknowledgments

This work was supported in part by the Spanish Ministry of Education and Science under grant TSI2004-06475-C02.

References

[1] Eichelberg M, Aden T, Riesmeier J, Dogac A and Leleci GB. A survey and analysis of electronic healthcare record standards. ACM computing Surveys 2005: 37(4): 277-315.
[2] Health Level 7: www.hl7.org
[3] European Committee for Standardization. Technical Committee on Health Informatics: www.centc251.org.
[4] European Committee for Standardization. Health informatics-Electronic health record communication- Part 1: Reference model. Draft European Standard for CEN Enquiry prEN13606-1, 2006.
[5] European Committee for Standardization. Health informatics-Electronic health record communication- Part 2: archetypes. Draft European Standard, prEN13606-2, 2006.
[6] OpenEHR foundation: http://www.openehr.org.
[7] Beale T, Heard S. Archetype Description Language 1.4 (ADL). The OpenEHR Foundation.
[8] Maldonado JA. Historia Clínica Electrónica Federada Basada en la Norma Europea CEN/TC251 EN13606 PhD. Dissertation. Technical University of Valencia, 2005.
[9] Beeri C, and Milo T. Schemas for integration and translation of structured and semi-structured data. Proceedings of the 7th International Conference on Database Theory (ICDT'99) , pp. 296-313, 1999
[10] Kuper GM, Simeon J. Subsumption for XML types. Proceedings of the 8th International Conference on Database Theory (ICDT'01), pp. 331-345, 2001.
[11] Popa L, Velegraki Y, Miller RJ, Hernández MA, Fagin R. Translating web data. Proceedings of the 28th VLDB Conference, pp. 598-609, 2002.

Address for correspondence

Jose Alberto Maldonado, Ph.D.
Biomedical Informatics Group. ITACA Institute.
Technical University of Valencia
Valencia 46022, Spain
jamaldo@upv.es