Implementing High-Level Query Language Interfaces for Archetype-Based Electronic Health Records Database

Shelly Sachdeva, Subhash Bhalla
Graduate Department of Computer and Information Systems,
University of Aizu, Aizu-Wakamatsu, Fukushima, 965-8580 (Japan)
{d8111107, bhalla}@u-aizu.ac.jp

Abstract

In contrast to a single doctor-patient relationship, there are several departments in a hospital. Thus health data may be scattered and can be termed as islands of information. Electronic health records (EHRs) can make healthcare organizations operate more efficiently. These will help in reducing medical errors, and improving health care, in general. The greatest challenge in the exchange of healthcare data about patients, concerns efficient and meaningful exchange. A query language is essential for health information systems. The proposed study aims to develop a graphical interface for querying EHR data. The motive of the research is to meet the querying needs of healthcare consumers.

1. Introduction

Electronic health record (EHR) is a patient-centric, longitudinal view of data collected from several source systems [1]. These help in providing information during emergencies. In general, EHRs can help as decision support tool, and in improving quality, security, time and communication among doctors. The demand for Electronic Health Records will increase gradually. In January 2011, the US government stimulus incentives will be enforced [3]. The rapid developments in the field of EHR present numerous challenges for data management. Also, determining the needed requirements at each user's level is difficult.

In essence, the proposed Electronic Health Records (EHRs) have a complex structure that may include data of about 100-200 parameters, such as temperature, blood-pressure and body mass index. Individual parameters have their own contents (Figure 1.). In order to serve as an information interchange platform, EHRs use archetypes to accommodate various forms of contents [10]. The EHR data has multitude of representations. The contents can be structured, semi-structured or unstructured, or a mixture of all three. These can be plain text, coded text, paragraphs, measured quantities with values and units, date, time, date-time, and partial date/time, encapsulated data (multimedia, parsable content), basic types (such as boolean, state variable), container types (list, set) and uniform resource identifiers (URI).

15th International Conference on Management of Data
COMAD 2009, Mysore, India, December 9–12, 2009

The International Organization for Standardization (ISO) has recently approved a new standard ISO 13606, for the communication and semantic interoperability of EHR extracts [5]. This standard is based on a dual model architecture. First a simple and generic reference model is defined for the representation of data at storage level (physical level). An archetype model is used for the representation of complex domain concepts of the EHR at conceptual level (logical level). The Reference model (RM) has a fixed (small) number of generic classes [4]. It represents the generic structures of components of health record information. The software and data are built from RM. Thus, healthcare information system can be built using relatively simple information models and conceptual database schemas. This reflects the stable characteristics of the design of EHRs and their components.

2. EHR and Archetypes

Archetypes provide semantic modelling (independent of software). These are expressed in form of constraints on data whose instances conform to a RM. Archetypes are language neutral and can be authored and translated into any language. Archetypes connect information structures to formal terminologies. RM is stable, where as the archetypes are dynamic. Every archetype is written with respect to a particular 'reference model'. The archetypes are used to constrain the valid structures of instances of the generic class belonging to RM. For example, a generic class "PARTY" can represent different domain concepts such as patient, doctor or nurse. The openEHR foundation, CEN/TC251 (European Committee of standardization) and Microsoft aims to generate mechanisms and standards for sharing health information worldwide, by using these archetype based EHRs [7,8,9].

2.1 Archetype Description Language (ADL)

Archetypes for any domain are described using a formal language known as Archetype description language

(ADL) [11]. ADL is path addressable like XML. The openEHR Archetype Object Model (AOM) describes the definitive semantic model of archetypes, in the form of an object model [12]. The AOM defines relationships which must hold true between the parts of an archetype for it to be valid as a whole. In simpler terms, all archetypes conform to AOM. Since EHR has a hierarchical structure, ADL syntax is one possible serialisation of an archetype. ADL uses three other syntaxes, cADL (constraint form of ADL), dADL (data definition form of ADL), and a version of first-order predicate logic (FOPL), to describe constraints on data which are instances of RM [11].

The ADL archetype structure consists of archetype definition (expressed using cADL syntax), language, description, ontology, and revision_history (expressed using dADL syntax), invariant section (expressed using FOPL). The invariant section introduces assertions which relate to the entire archetype. These are used to make statements which are not possible within the block structure of the definition section. Similarly, the dADL syntax provides a formal means of expressing instance data based on an underlying information Model [11]. The cADL is a syntax which enables constraints on data defined by object-oriented information models to be expressed in archetypes or other knowledge definition formalisms [11].

There are many parameters, such as weight, body temperature and heart rate in an EHR. The ADL for a parameter 'Blood Pressure' (BP) (also see Figure1) and other parameters are available at common repository [19]. ADL has a number of advantages over XML. It is both machine and human processable, and approximately, takes half space of XML. The leaf data type is more comprehensive set (including interval of numerics and date/time types). ADL adheres to object-oriented semantics that do not confuse between notions of attributes and elements. In ADL, there are two types of identifiers (from reference model) - the type names and attributes. Formally, it is possible to convert ADL into XML format and other formats [11].

3. Complexity of EHR data

For different parameters in EHR, the archetypes are distinct, structured models of domain content, such as, data, state and protocol. For example, 'data' may be captured for a blood pressure observation. It contains complete knowledge about a clinical context, (i.e., attributes of data). 'State' contains context for interpretation of data). 'Protocol' contains information regarding gathering of data, as shown in Figure 1. A total of 187 archetypes have been developed by openEHR till date [13]. These are expected to increase in number (may be thousands or more) as the domain knowledge regarding health expands. Thus, we are able to represent the complex EHR data and expanding knowledge concepts in a more efficient manner, through accepted standards. A domain knowledge governance tool called Clinical Knowledge Manager has been developed by openEHR foundation for development of archetypes [13].

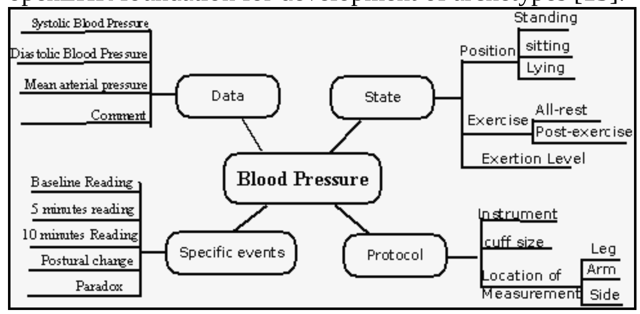

Figure 1. Parameter (Blood pressure) in Archetype form

4. Querying Archetype based EHR

The EHR system must have an appealing and responsive query interface that provides a rich overview of data and an effective query mechanism. The overall solution should be designed with an end to end perspective in mind. A query interface is required that will support users at varying levels of query skills. These include semi-skilled users at clinics or hospitals.

4.1 Archetype Query Language (AQL)

To query upon EHRs, a query language, Archetype Query Language (AQL) has been developed [15]. It is neutral to EHRs, programming languages and system environments. It depends on the openEHR archetype model, semantics and its syntax. AQL is able to express queries from an archetype-based EHR system. The use of AQL is confined to a skilled programmers' level. It was first named as EQL (EHR Query Language) which has been enhanced with the following two innovations [14]:

i) utilizing the openEHR path mechanism to represent the query criteria and the response or results; and

ii) using a 'containment' mechanism to indicate the data hierarchy and to constrain the source data to which the query is applied.

4.2 High-Level Database Query Interfaces

AQL is difficult for semi-skilled users. It requires the knowledge of archetypes and knowledge of languages such as SQL and XML. At the present moment, there is no easy-to-use query language interface available for EHRs database. We propose to study a high-level interface for querying EHR database based on the proposed XQBE [16]. An alternative approach proposed by Ocean informatics [20] suggests using a query builder tool, to construct AQL query. It requires form related inputs and more skills on the part of the user. The goal is similar and it is easier to achieve with the help of XQBE.

XQBE [16] is a user-friendly, visual query language for expressing a large subset of XQuery in a visual form. It requires all data to be in XML form. Its simplicity, expressive power and direct mapping to XQuery are some of the highlighting features for its use. Like XQuery, XQBE relies on the underlying expressions in XML. It presents a user with XML sub-tree expressions for the items of user interests. XQBE's main graphical elements are trees. There are two parts, the source part which describes the XML data to be matched against the set of input documents, and the construct part, which specifies which parts will be retained in the result, together with (optional) newly generated XML items.

In order to adopt a XQBE like interface at user level, we propose to convert ADL into XML. ADL can be mapped to an equivalent XML instance. ADL is hierarchical in nature and provides a unique identification to each node. Thus, paths are directly convertible to XPath expressions. These can be created. According to Beale and Heard [11], the particular mapping chosen may be designed to be a faithful reflection of the semantics of object-oriented data. There may be need for some additional tags for making the mapping of nested container attributes since XML does not have a systematic object-oriented semantics. Thus, single attribute nodes can be mapped to tagged nodes of the same name. Container attribute nodes map to a series of tagged nodes of the same name, each with the XML attribute 'id' set to the node identifier. Type names map to XML 'type' attributes.

In the present proposal, the patient data description is converted to XML form. It is suitably reformed for adoption of XQBE interface. Thus users can directly use XQBE query interface to access patient data. This process eliminates the need to learn and use the AQL language on the part of the users. The XQBE skills can be learnt with ease [16].

5. Mapping ADL to XQBE for EHR data

The queries play a crucial role in decision support and epidemiological situations. The XQBE approach for archetype-based EHRs is being proposed for semi-skilled users (such as doctors, physicians, nurses). The mapping process to create XQBE is shown in following steps (Figure 2).

i) The conversion of ADL file into XML file.
ii) Generation of DTD for the XML file (Figure 3).
iii) Generation of XQBE interface structure (Figure 4).

Subsequently, for the semi-skilled user, this three step process will facilitate in querying archetype based EHRs.

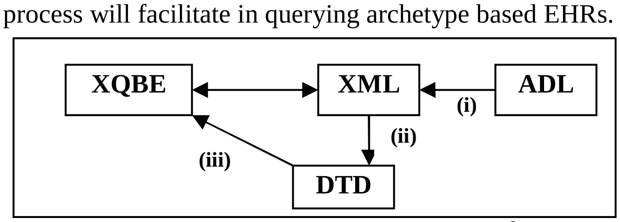

Figure 2. Mapping process to present XQBE for EHRs.

The step (ii) in above process will aid in the guided construction of query provided by XQBE [16]. However, for some users (skilled in use of XML) the step (ii) may not be needed and XQBE can be used directly for the XML file.

In order to highlight a contrast between different query languages, consider the following examples.

Query Scenario1. Find all Blood Pressure values, having the systolic_BP and diastolic_BP, (where systolic_BP >= 140 and diastolic_BP >=90).

Chunlan,et al.[14] gives the AQL syntax for the above query. By using XQBE approach for querying, we perform step (i) to step (iii) as explained, on BP parameter. For each case of query, and for querying different parameters of EHR, we need to convert each parameter (in form of adl) to a corresponding xml for the demonstration. We propose to develop an automated tool in the subsequent phase. The clinical user will be provided with a substituted XQBE interface (Figure 4) in place of AQL.

```
<!ELEMENT adl_version ( #PCDATA ) >
<!ELEMENT archetype ( original_language,
description, archetype_id, adl_version, concept,
definition, ontology ) >
<!ELEMENT archetype_id ( value ) >
<! ELEMENT assumed_value (terminology_id,
code_string, magnitude?, units?, precision? ) >
<! ELEMENT attributes (rm_attribute_name,
existence, children+, cardinality? ) >
<! ATTLIST attributes xsi: type
(C_MULTIPLE_ATTRIBUTE |
C_SINGLE_ATTRIBUTE) #REQUIRED >
<! ELEMENT cardinality (is ordered, is unique,interval)
```

Figure 3. A sample of BP.dtd

XQBE is a visual interface. A user is presented with graphical image of EHR components, for example, blood Pressure (BP) in this case. In Figure 4, based on the selected source data, the user defines a target sub-tree (in XML form) to express the query (and its outcome). The query is expressed by using the graphical elements of XQBE [16]. The source part of the query is built using the DTD. A guided construction is provided to the user to add predicates for the query. The construct (or result) part of the query is built by the user using the graphical elements of XQBE by dragging and dropping them. Figure 4 shows the element nodes and subelement nodes in the source part.

The subelements (systolic and diastolic) of the BP element, one systolic and one diastolic satisfy condition1 (systolic>=140) AND condition2 (diastolic>=90) are described with the help of XQBE convention. As per the convention, an arc containing '+' indicates that 'children' element node may exist at any level of nesting (as in Xpath we use '//'). The construct part consists of element node for BP (set tag T-node), and also element nodes for systolic and diastolic, which relates the projected BP element nodes to its systolic and diastolic subelements. The fragment node (shown by filled triangle) indicates that the entire fragment of systolic and diastolic must be retained in the result. A binding edge between source part and construct part indicates to construct as many BP element nodes (in construct part) as BP element nodes (in source part).

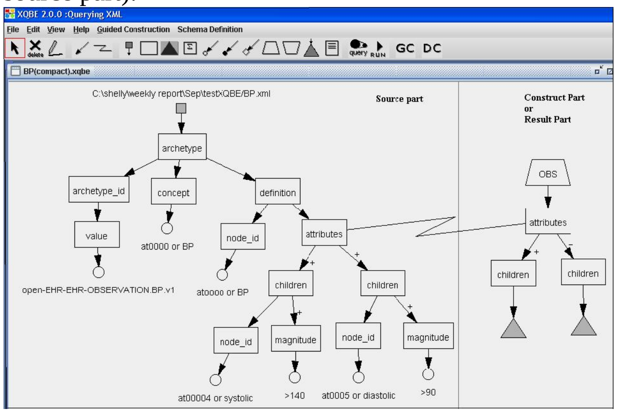

Figure 4. BP.XQBE- an XQBE template for query

The XQBE will generate a corresponding XQuery expression, which may be executed.

Query Scenario2. Find details for all blood pressure values with position recorded.

At the user level, in Figure 5, the query is expressed by using the graphical elements of XQBE. The query indicates the existential quantification, .i.e., in order for BP element to be included, a position element must exist.

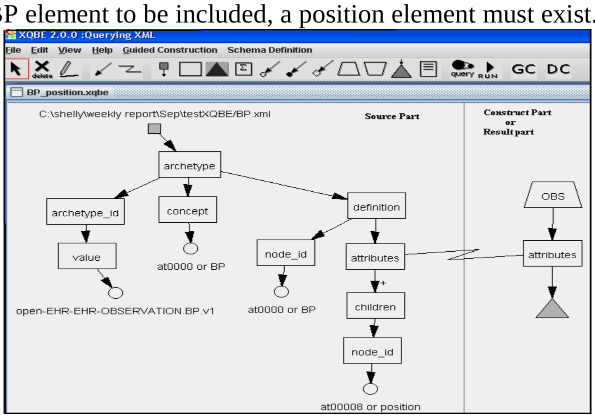

Figure 5. An XQBE template for query.

5.1 Implementation Details and Related Works

The openEHR specifications for AQL and EHR are based on an object-oriented framework. Thus, these allow multiple representations [14]. Hence, EHRs can be represented as relational structures (governed by an object/relational mapping layer), and in various XML storage representations [14].Our aim is to prepare a prototype system with query support for the high-level interface for the semi-skilled users. An intermediate layer of mappings has being developed in XML to perform the conversion of form using an editor [17]. Further, a mechanism has been developed to convert BP.xml to BP.dtd. Subsequently, XQBE 2.0.0 interface can be used to provide the query interface for the query scenarios [18]. The above steps demonstrate the feasibility of adoption of a XQBE interface for EHR data.

6. Summary and Conclusions

EHRs are becoming a reality, as a consequence a number of healthcare providers and their partners are responding to the government and patient demands for better service. A high level interactive query interface for querying EHR data is required. The research aimed at developing a query interface for semi-skilled and skilled healthcare professionals. It will increase productivity of doctors, clinicians. The high-level interface has been developed by mapping ADL to match the XQBE requirements. Skilled and semi-skilled workers using EHR data can use XQBE, an easy-to-use query language interface.

References

1. Jens Edlef Møller, Henrik Vosegaard ”Experiences with Electronic Health Records” IT Pro, published by IEEE Computer Society, March/April 2008.
2. http://www.bizjournals.com/twincities/stories/2009/08/24/daily55.html?s=industry&i=healthcare FierceEMR,3 Sep,09.
3. http://www.healthdatamanagement.com/news/ARRA-38893-1.html (news Aug 28,2009)
4. Beale T, Heard S., Kalra D., Llyod D., The openEHR Reference Model: EHR Information Model. The openEHR release 1.0.2. , openEHR Foundation, 2008.
5. ISO 13606-1, Health informatics - Electronic health record communication - Part 1: RM, 1st edition, 2008.
6. ISO 13606-2, Health informatics - Electronic health record communication - Part 2: Archetype Interchange Specification, 1st edition, 2008.
7. www.openehr.org (Accessed 02/06/2009)
8. www.cen.eu (European committee for Standardization, Technical committee on Health informatics, Standard for EHR communication).
9. http://www.microsoft.com/industry/healthcare/technology/HealthFramework.mspx (Microsoft Health framework).
10. BealeT,Heard S.,” openEHR Architecture: Architecture Overview in The openEHR foundation, release 1.0.2.”,published by openEHR Foundation, 2008.
11. Beale T, Heard S.,” The openEHR Archetype Model-Archetype Definition Language ADL 1.4”, openEHR release 1.0.2, Issue date 12 Dec 2008.
12. Beale T.,” The openEHR Archetype Model-Archetype Object Model”, openEHR release 1.0.2, 20 Nov 2008.
13. http://www.openehr.org/knowledge/ (Clinical knowledge Manager) (Accessed 07/05/2009).
14. Chunlan M., Heath F., Thomas B., Sam H.,”EHR Query Language (EQL)-A Query Language for Archetype-Based Health Records”, MEDINFO 2007.
15. http://www.openehr.org/wiki/display/spec/Archetype+Query+Language+(AQL)+(Ocean) (Archetype Query Language) (Accessed 28/05/2009)
16. D. Braga, A. Campi, Stefano Ceri,” XQBE (XQueryBy Example): A Visual Interface to the Standard XML Query Language”, ACM Transactions on Database Systems, Vol. 30, No. 2, June 2005, PP. 398–443.
17. https://wiki.oceaninformatics.com/confluence/display/TTL/Archetype+Editor (Archetype editor).
18. http://dbgroup.elet.polimi.it/xquery/XQBEDownload.html (XQBE 2.0.0).
19. http://www.openehr.org/svn/knowledge/archetypes/dev/html/index_en.html (ADL for archetypes) (Accessed 04/07/2009).
20. http://www.oceaninformatics.com/Solutions/ocean-products/Clinical-Modelling/Ocean-Query-Builder.html (Query builder, Ocean Informatics) (Accessed 10/06/2009).