MEDINFO 2015: eHealth-enabled Health
I.N. Sarkar et al. (Eds.)

doi:10.3233/978-1-61499-564-7-30

Template and Model Driven Development of Standardized Electronic Health Records

Stefan Kropf[a], Claire Chalopin[a], Kerstin Denecke[a]

[a] *Innovation Center Computer Assisted Surgery, University of Leipzig, Germany*

Abstract

Digital patient modeling targets the integration of distributed patient data into one overarching model. For this integration process, both a theoretical standard-based model and information structures combined with concrete instructions in form of a lightweight development process of single standardized Electronic Health Records (EHRs) are needed. In this paper, we introduce such a process along side a standard-based architecture. It allows the modeling and implementation of EHRs in a lightweight Electronic Health Record System (EHRS) core. The approach is demonstrated and tested by a prototype implementation. The results show that the suggested approach is useful and facilitates the development of standardized EHRSs.

Keywords:

Electronic Health Records, Reference Standards, Information Storage and Retrieval, Medical Informatics Applications, Medical Records Systems, Computerized, Health Information System

Introduction

Patient data are distributed in heterogeneous information systems [1] Accordingly, physicians need to interact with several systems to get a complete view of the health status of a patient. In addition, researchers need to access and reuse patient data for calculations or studies. In both scenarios, it is crucial to have all relevant patient data available and accessible in a standardized, patient-specific information model (IM) [2]. The concept of digital patient modeling adresses this issue and aims at integrating patient data which is distributed in several information systems [3]. Patient modeling on an information model layer corresponds to the development of standardized EHRSs, both for research and for health care, preferably following the open source approach [4]. At a first glance, the development of standardized EHRs and systems that handle these EHRs (an EHRS) seems to be an old-fashioned topic, where practical instructions and solutions already exist. Indeed, EHR standards are available. However, there is a lack of accessible instructions regarding how to apply these standards in practice, and more specifically, how to use standards in real world environments. Most reviewed health informatics-related standards can be classified [5] (ISO 17119) from the perspective of "what," along with conceptual specificity. Standards which can be classified from the alternative perspective of "how," and with *physical* specificity – like the ECG protocoll definition defined in EN 1064 – are rare. This is also reflected by the large number of publications that mainly concentrate on conceptual or logical specificity [6-8]. For implementing standardized EHRSs, developers need methods and instructions or practical solutions like Free Open Source Software (FOSS) analyses [9], the storage of data in persistence layers [10] or ideally an API, [11] development workflows, [12] or even a complete usable EHRS [13]. Therefore, this work introduces a system architecture that combines openEHR and XForms for the development of standard-based EHRs and EHRSs. These standards have been chosen since XForms is a well-established standard of the World Wide Web Consortium (W3C), and openEHR is recommended by e-health authorities such as NEHTA [14]. We used these technologies to build an information system for monitoring patients diagnosed with pituitary adenoma as a proof-of-concept. This paper focuses on the technical details of the development process.

Materials and Methods

We suggest a model-driven architecture based upon existing standards. Such a design principle decouples modeling processes from development processes. Standards are used to fulfill basic requirements in system design such as accessibility, reusability, and interoberability to achieve long term durability. The approach combines W3C-standards originating in web development with openEHR standards (see figure 1). These two groups of standards are interconnected: Instances of openEHR archetypes are mappable to XML.

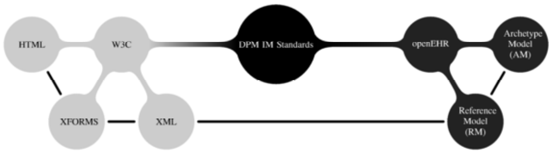

Figure 1: Selection of related W3C and openEHR standards

In the following, the selected standards, methods, and the development process are described. Later, we reflect upon issues that have been considered in the development process.

OpenEHR

Choosing the right standard for an EHRS is a challenge [15, 16]. Normative EHR information models have already been published by the European Committee for Standardisation (CEN)/Technical Committee (TC) 251. These European Norms are based notably on openEHR [17] and Health Level 7 (HL7) v3 Reference Information Model (RIM) [18]. OpenEHR archetypes are transformable to EN 13606 archetypes and vice versa [19, 20]. Both EN 13606 [17] and EN 14822 [18] are suitable for the information model, but openEHR delivers the most comprehensive approach. In short, openEHR delivers a large set of generalized, predefined, standardized building blocks (the archetypes). Reuse of these building blocks is ideal, because they provide a highly qualitative result of a collaborative development process of domain experts.

However, the collection and ordering of data should be a doctor's first function [21]. openEHR enables domain experts to create information structures without the need of a detailed technical understanding. Such domain modeling by experts is a great opportunity, because physicians can be integrated into the structuring process of medical records, [22] yielding a better understanding of the importance of well-structured, finely granular medical records. Further, it minimizes the need of tasks that invoke expensive downstream processing of natural language.

XForms

XForms is a standard published by the W3C. It follows the Model View Controller (MVC) pattern by seperating the model from its representation [23].

Figure 2 illustrates the XForm technique: The GUI and the input fields are bound directly to the XML via the XForms standard. Important for further reading is the model tag inside the XForms markup. It is located in the header of the file and includes the instance of the model, mapped to XML with possible localized namespace declarations.

The body of the XForm consists of model entities, referenced by a simple "ref" attribute. Figure 2 shows this convenient approach. A simple XPath reference binds input fields to the model definition. This binding concept demonstrates the core idea: The openEHR and XForms standards are glued together to benefit from their respective advantages. The digital patient model of one single EHR is represented by the corresponding model element in an XForm [2-12 in Figure 2].

```
1  <html [...]
2          <xforms:model id="4463197">
3                  <xforms:instance>
4                          <dpmim:myehr [...]
5                                  <dpmim:value>[...]
6                                  <dpmim:magnitude>507.8</dpmim:magnitude>
7                                  [...]
8                                  </dpmim:value>
9                          [...]
10                         </dpmim:myehr>
11                 </xforms:instance>
12         </xforms:model>
13 <xforms:body>[...]
14         <xforms:input ref="/dpmim:myehr/[...]/dpmim:value/dpmim:magnitude">
15                 <xforms:label>Cortisol</xf:label>
16         </xforms:input>
17         [...]
18 </xforms:body>
19 </html>
```

Figure 2: XForms model binding concept

Related OpenEHR Based EHRS

There are several openEHR-based frameworks available. Some of them are independently usable by the underlying persistant layer, such as Chen's openEHR Java reference implementation (http://github.com/wware/openehr-java/). Other approaches are bound to schemes in relational databases. LiU EEE (https://github.com/LiU-IMT/), medrecord (http://www.medrecord.nl/overview/), and EHRflex (http://ehrflex.sourceforge.net/) use XML databases for their persistant layer. Initial tests showed that the Representational State Transfer (REST) architecture of LiU EEE and medrecord is the most promising approach for realizing an EHRS. The REST architecture was already tested in openEHR implementations and showed how an EHRS should work physically [24, 25]. Consequently, we have chosen a REST architecture for our implementation. In addition, our approach combines XForms and openEHR, which results in a standardized storage of patient data in the system. To the best of our knowledge, such an approach has neither been introduced nor implemented previously in the context of patient modeling.

Development Process

The development process is divided into (i) model development and (ii) model-driven application development. This separation of the modeling process from the development process is highly beneficial, as it "empowers the health professional to define and alter the accurate knowledge and information they need in the granularity they need" [22]. Figure 3 shows the separation of work.

Domain Experts develop domain models

Within our concept, domain experts develop the domain model using the openEHR tools archetype editor and template designer. These tools enable clinicians to model their medical records themselves without deeper knowledge about databases.

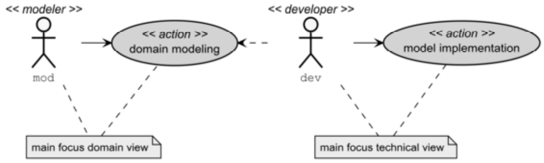

Figure 3: Separation of domain modeling and application development

Figure 4 illustrates the model development portion of the pipeline:

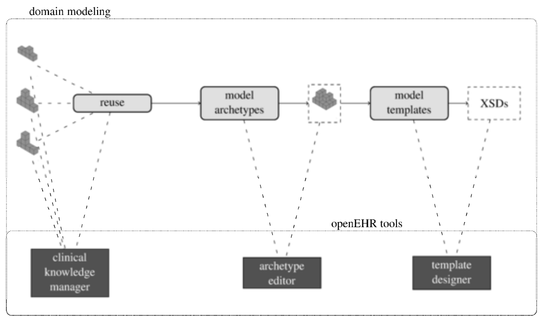

Figure 4: Domain modeling: generation of XSDs underpinned by openEHR tools

The first step in the domain modeling phase is the selection and, thus, reuse of existing building blocks – the archetypes. With the Clinical Knowledge Manager (CKM) [http://www.openehr.org/ckm/], a web repository of openEHR-compliant relevant archetypes for a specific application can be identified. Reuse of archetypes is beneficial for achieving a high degree of semantic interoperability among systems.

After identifying reusable archetypes, they are constrained to the specific use case [26]. New domain-specific archetypes can be modeled with the openEHR-tool archetype editor [27].

This modeling process yields archetypes that are formulated in a standardized language, the Archetype Definition Language (ADL). They are combined in so-called templates (i.e., larger structures), using the openEHR-tool template designer [27]. This concept of building templates out of smaller building blocks increases the probability of reusing single archetypes in different templates. Finally, after this modeling process, models are exported as XML Schema Definitions (XSDs). The W3C XML Schema provides a usable solution for the semantic validation of standard-based EHRs [28]. These

XSDs are then used as input files for the next phase: the model-driven application development.

Model-Driven Application Development and Architecture

Figure 5 shows the process of the application development and the system architecture in general. Out of the XSDs generated in the domain modeling phase, valid XML files are created. We refer to these files by the term “skeleton,” because they do not contain real patient data. To put it differently, they are the EHR templates. For each XSD, at least one corresponding XML skeleton is generated. Such an XML/XSD pair enables the validation of the EHRs according to the models at any time, which is highly beneficial, because standard compliance is ensured. The XML/XSD pairs can be exploited as a starting point for development using XML techniques like XSLT, or direct programming of the XML-DOM. However, in this paper, another more accessible solution is proposed: The generated valid XML is inserted into the XForms model and input elements are bound to the XForms model.

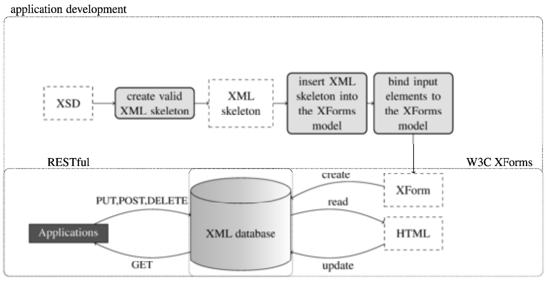

Figure 5: Model driven application development: XForm generation based on XSDs

The source code in Figure 2 shows the approach. To keep this paper as simple as possible, only an extract of one input field is listed. In practice, the instance/model part will be more complex. However, the integration of the model remains simple since it consists of only one copy/paste step. After this insert, the widgets are bound to the model with the “ref” attribute. Finally, the form just has to be designed. After these processing steps, an XForm with an integrated EHR model is stored in the XML database. The form data can then be read and viewed by clients, and updates can be stored and managed by the underlying open souce XML database system eXistdb [29].

In addition to the XML-XForm-GUI binding, the XML database enables the retrieval and storage of complete EHRs via the REST interface, which will be very handy for any kind of future development. In essence, the left side of Figure 5 illustrates that applications can easily access these XML data by a RESTful interface instantly delivered by the XML database eXistdb [30].

Results and Discussion

In this paper, a complete development process for standard-based EHRs was introduced. The domain modeling promises a high quality of the underlying models, and the application development ensures a rapid implementation. To achieve a high level of interoperability, the process starts with the reuse of archetypes, integrates free modeling tools, and ends in an accessible XML-based EHRS. The resulting XML can be interpreted as one possible representation of a personalized digital patient model. The archetypes and templates resulting from the domain modeling task are not only simple structures of information or web masks. Instead, they are valuable sources for other clinicians at other sites, formulated in a standardized language (ADL). These models can be published in worldwide repositories, which allows their reuse in other contexts where they can achieve a similar impact as publications in journals or conferences.

As a result of the combined usage of the EHR standard openEHR and the W3C standard XForms, a high level of accessibility, reusability, interoperability, and therefore long term durability is achieved. Essentially, the combination of the HL7 v3 RIM and XForms would work similar and would benefit also from the following advantages:

Nowadays, most of the information systems are internally based on traditional relational databases [31]. Much effort has to be done to convert this data into different formats (e.g., Extract Transform Load (ETL)). These ETL processes are very time consuming and also error-prone. To avoid these drawbacks, XML-based messages and whole XML-based EHRSs may become a standard solution in medical informatics (at least for small EHRSs).

The benefit of this paradigm shift is obvious when comparing this approach to other developing processes of semantically interoperable EHRs [7]. The main reason is that object-relational class mappings and ETL processes for the generation of messages or single EHRs can be ommitted.

EHRs and messages inside and between hospital information systems (HISs) are defined differently, called the “Message / Record Dichotomy” [6]. With the paradigm shift to XML for EHR storage, it is possible to use EHR extracts stored in data bases directly as messages, and vice versa. The resulting EHRs can easily be transformed to messages for exporting data to another HIS. This is an essential step towards standardized, generalized, and therefore simplified HIS-infrastructures.

Performance

A comparison of querying openEHR based XML databases indicates that XML databases today are not yet the right solution for a persistance layer of big data [32] As a proof-of-concept, we made a load test on the resulting EHRS. 10,000 (64 KB) XML files were transmitted via POST requests, and the time for GET requests was measured. Additionally, the latency time for the XForm-based GUI generation was detected. The results show that the latency ascends constantly with an increasing number of files (see figure 6).

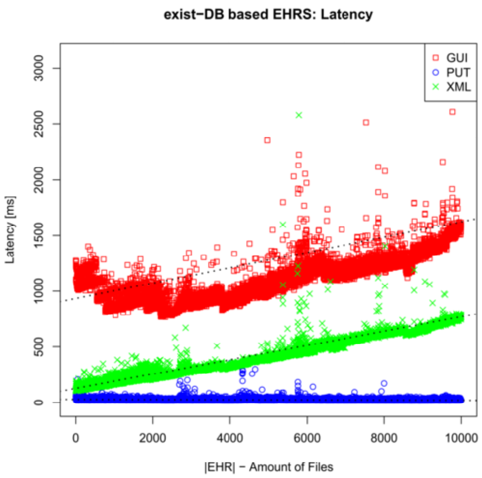

Figure 6: eXistdb based EHRS: Latency

Clearly, the performance of the application slows down with increasing database volume. However, since a total number of 10,000 EHRs will not be reached in our use case, the access times are acceptable. Anyway, because of the ongoing rapid development of storage and processing power, possible performance issues relating larger XML based EHRSs will become more and more negligible.

The Null Flavour Issue

One issue occurring in the XForms solution regards "null flavor" values. To explain this issue, consider the XForm entry in the normal case, e.g. when the value of the element "magnitude" is "507.8":

<dpmim:value>[…]

<dpmim:magnitude>507.8</dpmim:magnitude>

[…]

</dpmim:value>

The "magnitude" element does not allow empty values, which is of course reasonable. The valid solution to map empty values in XML would be the replacement of the whole <dpmim:value> element with a <dpmim:null_flavour> element, which contains the openEHR code string "271," which means "no information". Thus, in our example the entire <dpmim:value> element disapears, and is replaced with the <dpmim:null_flavour> element, resulting in:

<dpmim:null_flavour>[…]

<oe:code_string>271</oe:code_string>

[…]

</dpmim:null_flavour>

In combination with the XForm technique, this is an issue, because input fields can not be bound to possibly disappearing elements. One possible workaround is allowing null values in primitive data types: When a value is empty, the relating element will get a null attribute:

<dpmim:magnitude xsi:nil="true"/>

To ensure valid XML, the generated XSD has to be adjusted by allowing null values by the command `nillable="true"` in the correspondent auto-generated XSD element. This workaround allows the use of null values within the XForms model.

Minimal XML

Naturally, XML files come along with relatively much overhead. Additionally, archetypes imply intristic overhead, because they are generalized to enable their usage in different szenarios.

For our prototype system, an XML EHR template containing about 7,000 lines of code for only 24 entry fields was generated out of the XSD files. The overhead was reduced manually by removing unnecessary elements; the final EHR had about a few hundred lines. The important lesson we learned is that contraining the archetype as much as possible is one important prerequisite for the generation of minimal XML-based EHRs.

Conclusions

In this paper, a model-driven and standard-based development process for EHRs has been presented. The practical approach of rapid model driven development of standardized EHRs is recommendable for the development of small EHRSs. Aside from the benefits of standardized EHRs, this boosts the development process. After considering pros and cons of various approaches, a complete paradigm shift to XML for storing EHR data is suggested. XML containing patient data based on openEHR provides a suitable solution for representing digital patient models. The concept was validated by the development of a prototype for a special use case, which is an information system for monitoring patients diagnosed with pituitary adenoma, which will be described in another paper. In future work, additional applications will be implemented that will reuse the concept to use the strength of this approach with respect to application development.

References

[1] Bisbal J, Berry D. An analysis framework for electronic health record systems. Methods Inf Med. 2011;50(2):180--189.

[2] Kropf S, Denecke K. Digital Patient Model - Information Model. University of Leipzig; 2013. p. 123. 12th Leipzig Research Festival for Life Sciences.

[3] Denecke K. Model-based Decision Support: Requirements and Future for its Application in Surgery. Biomedical Engineering/Biomedizinische Technik. 2013;.

[4] Ohmann C, Kuchinke W. Future developments of medical informatics from the viewpoint of networked clinical research. Methods Inf Med. 2009;48(1):45--54.

[5] Health informatics - Health informatics profiling framework; 2004. ISO/TR 17119:2004(E).

[6] Beale T. Health Information Standards Manifesto. 2001; Available from: http://workflow.healthbase.info/monographs/ [cited 03.11.2014].

[7] Blobel B, Engel K, Pharow P. HL7 Version 3 compared to advanced architecture standards. Methods Inf Med. 2006;45:343--53.

[8] Goossen W, Goossen-Baremans A, van der Zel M. Detailed Clinical Models: A Review. Healthc Inform Res. 2010 Dec;16(4):201--214.

[9] Zuniga F, Enrique A, Win K, Susilo W. Functionalities of free and open electronic health record systems. International journal of technology assessment in health care. 2010;26(4):382--389.

[10] openEHR wiki persistence FAQs;. Available from: http://www.openehr.org/wiki/display/resources/Persistence+FAQs [cited 16.08.2013].

[11] Klasson F, Väyrynen P. Development of an API for creating and editing openEHR archetypes. Linköping; 2009. Masterthesis, urn:nbn:se:liu:diva-17558 (URN).

[12] Lopez DM, Blobel BG. A development framework for semantically interoperable health information systems. Intern. J. of Med. Inform. 2009;78(2):83--103.

[13] Atalag K, Yang HY. From openEHR Domain Models to Advanced User Interfaces: A Case Study in Endoscopy. In: Health Informatics New Zealand Conference; 2010.

[14] NEHTA Review. National E-Health Transition Authority Ltd : Review of Shared Electronic Health Record Standards. 2006. – Research report

[15] Begoyan A. An overview of interoperability standards for electronic health records. In: Proceedings of the 10th International Conference on Integrated Design and Process Technology, IDPT 2007, Antalya, Turkey.2007. p. 3--8

[16] Eichelberg M, Aden T, Riesmeier J, Dogac A, Laleci G. Electronic health record standards--a brief overview. In: Information Processing in the Service of Mankind and Health: ITI 4th International Conference on Information and Communications Technology: Information Technology Institute, Giza, Egypt.; 2006. p. 7--19.

[17] Health informatics - Electronic health record communication [Norm]; 2012. EN ISO 13606 standard familiy.

[18] Health informatics - General purpose information components [Norm]; 2006. DIN EN 14822 standard family.

[19] Martinez-Costa C, Menarguez-Tortosa M, Fernandez-Breis JT. An approach for the semantic interoperability of ISO EN 13606 and OpenEHR archetypes. J Biomed Inform. 2010 Oct;43(5):736--746.

[20] Costa CM, Menarguez-Tortosa M, Fernandez-Breis JT. Clinical data interoperability based on archetype transformation. J Biomed Inform. 2011 Oct;44(5):869--880.

[21] Weed LL. Medical records, patient care, and medical education. Irish Journal of Medical Science (1926-1967). 1964;39(6):271--282.

[22] Garde S, Knaup P, Hovenga EJ, Heard S. Towards Semantic Interoperability for Electronic Health Records--Domain Knowledge Governance for open EHR Archetypes. Methods Inf Med. 2007;46(3):332--343.

[23] Boyer JM. XForms 1.1 W3C Recommendation [Norm]; 2009. Available from: http://www.w3.org/TR/xforms11/.

[24] Sundvall E, Nyström M, Sandström M, Eneling M, Örman H, Karlsson D. REST Based Services and Storage Interfaces for openEHR Implementations;. Available from: http://www.imt.liu.se/~erisu/2010/EEE-Poster-multipage.pdf [cited 03.11.2014]

[25] Sundvall E, Nyström M, Karlsson D, Eneling M, Chen R, Örman H. Applying representational state transfer (REST) architecture to archetype-based electronic health record systems. BMC medical informatics and decision making. 2013;13(1):57.

[26] Beale T. Archetype Object Model; 2007. Revision: 2.0.1. Available from: http://github.com/openEHR/specifications/raw/Release-1.0.1/publishing/architecture/am/aom.pdf [cited 03.11.2014].

[27] Heard S., Beale T. openEHR – Modeling Tools. Available from: http://www.openehr.org/downloads/modelingtools [cited 03.11.2014].

[28] Rinner C, Janzek-Hawlat S, Sibinovic S, Duftschmid G. Semantic validation of standard-based electronic health record documents with W3C XML schema. Methods of information in medicine. 2010;49(3):271.

[29] Meier W. eXist: An open source native XML database. In: Web, Web-Services, and Database Systems. Springer; 2003. p. 169--183.

[30] Cahlander L, Cross P, Geldiyev Z, Stallman A, Turpin M. eXistdb open source XML database software architecture description. W3C: Introduction to Software Architecture. 2010.

[31] DB-Engines Ranking; Available from: http://db-engines.com/de/ranking [cited 22.04.2014].

[32] Freire SM, Sundvall E, Karlsson D, Lambrix P. Performance of XML Databases for Epidemiological Queries in Archetype-Based EHRs. In: Scand Conf Health Inform. vol. 13; 2012. p. 51--57.

Address for correspondence

Stefan Kropf / Innovation Center Computer Assisted Surgery

Semmelweisstr. 14

D - 04103 Leipzig, Germany