Survey of openEHR storage implementations

Samuel Frade[1], Sergio Miranda Freire[2,3], Erik Sundvall[3], José Hilário Patriarca-Almeida[1], and Ricardo Cruz-Correia[1]

[1]*Center for Research in Health Technologies and Information Systems - CINTESIS, Faculty of Medicine of University of Porto (FMUP) - Portugal*
[2]*Department of Information and Education Technology in Health Care, State University of Rio de Janeiro, Brazil*
[3]*Department of Biomedical Engineering, Linköping University, Sweden*
{samuelfrade, jalmeida, rcorreia}@med.up.pt, sergio@lampada.uerj.br, erik.sundvall@liu.se

Abstract

Efficiently storing and retrieving archetype-based patient information can be a challenging task. This paper surveys current archetype-based system implementations in the world and in particular the different approaches that have been used to create 13606- or openEHR-based storage repositories. Data is reported from systems with a few records to millions of records, including both deployed systems in production and experimental systems. Worldwide 21 projects were found, 4 did not reply and 1 did not provide data. Many systems (n=11) base their storage on RDBMS, then often (n=6) with some XML data fields. Dedicated XML (n=3) and object-oriented (n=2) databases were other examples of storage used. Query formalisms used include SQL, AQL, XQuery and XPath. Service interfaces via SOAP (n=12) or REST (n=6) are common. Most systems support dynamic configuration using new/changed archetypes and templates dynamically without system restart. Some (n=7) systems use demographic archetypes. In addition to the built in DBMS indexing mechanisms, one project reports use of an additional inverted index to achieve improved performance.

1 Introduction

The medical area is continuously changing, so it is very difficult to define one standardised information representation that is valid for all data that might need to be stored.

OpenEHR allows the standardization of the Electronic Health Record (EHR) architecture following a multi-level modelling approach, which separates information from knowledge [1]. The first level (the reference model - RM) specifies a generic model according to which data will be stored and communicated. The second level (the archetype model) defines constraints to the reference model that represents concepts in a specific domain. This fact changes the way health information systems are developed. Domain experts define the structure and element types of the domain concepts (making it possible to create new concepts or update the current ones), while the system developers are just concerned about creating instances that represent the data according to the RM and the archetypes and creating user interfaces for the templates [1]. The ISO 13606 series of standards follows a similar approach to openEHR, but it is based on a different RM [2, 3].

The Archetype Query Language (AQL) [4], that uses paths (similar to XPath) to access archetype based EHR data, has been suggested as an interoperable query method that is independent of underlying storage implementation design.

The dual-model approach is claimed to facilitate the evolution of EHR systems. However the efficient storage and retrieval of archetype-based patient information in openEHR- and 13606-format is not straightforward. There is very little available information on experimental or production openEHR-based systems and how they address the storage issue. Some measurements [5,6] and implementation suggestions [7,8] have been published.

This paper is a survey on the current implementations of openEHR-based repositories.

978-1-4799-1053-3/13/$31.00

2 Methods

2.1 Search of projects

Having a working archetype-based storage repository was the criteria for selection. The search was made using the following sources:

- websites related to openEHR containing lists of projects,
- openehr mailing lists archives and
- personal contacts

2.2 Collected variables

A form with a number of variables was created and sent to each of the projects found in the search. A representative of the respective project answered the form. The collected variables on each project were:

- Data Model: Type of Data Models used to store and access data (eg: Relational, XML, Object Oriented,etc)
- Database engines: Database engine(s) currently used in the project (eg:Oracle,Microsoft SQL Server,HBase,etc)
- Querying: Type of querying used to retrieve data (eg: AQL, SQL, Xquery,etc)
- Number of installations: Number of installations in production.
- Number of users: Total number of users accessing the installations in production (added together).
- Number of patient records: Total number of patient records (EHRs) stored in all installations.
- Database Size (GB): Size in GB allocated by the databases of all installations
- Licensing: EHR system licence (eg: Apache 2.0, Creative Common, Commercial,etc)
- Service Layer: If it has a Service Layer (eg: web services, REST, etc)
- Dynamic configuration: If it supports new/changed archetypes and templates dynamically (without need to restart or redeploy)
- Demographic archetypes: If it uses demographic archetypes
- Auditing: If it uses openEHR version control mechanisms (eg: Contribution and versioned_composition)
- openEHR Release: what version of the specifications was used
- ADL version: what version of the ADL was used

3 Results

3.1 Project Search Results

21 projects were found related to 19 companies or institutions. Of these, 4 did not reply and 1 would not provide data. We ended up with data from 16 different projects. (see Table 1).

Table 1: Respondent projects names and information

	Project/Product	Company/Institution	Abbreviation	Country
Companies	Base24	Code24 B.V.	B24	NL
	eWEAVE	eWEAVE AB	eW	SE
	HSEAVS	Valencia Health Agency, Tech. Univ. Valencia, Novasoft, Everis	HSEAVS	ES
	OceanEHR platform	Ocean Informatics	OEHR	AU
	Open EHRGen	CaboLabs	OGen	UY
	Open EHRServer	CaboLabs	OServ	UY
	OpenHealth:MultiCare	Infinnity Solutions	OH	RU
	OpenKernel	ROSA Software	OK	NL
	RecordPoint	Extensia	RP	AU
	Sistema Clinico Integrado	IdealMed	Crit	PT
	Think!EHR Platform	MARAND	TEHR	SI
R&D Institut.	EHRFlex	Tech. Univ. Valencia, Veratech	EHRF	ES
	GastrOS	University of Auckland	GOS	NZ
	LinkEHR	Tech. Univ. Valencia, Veratech	LEHR	ES
	LiU EEE	Linköping University	LiU	SE
	OpenEyes	Moorfields Eye Hospital	OEyes	UK

The projects that answered our form were run by companies (n=11), universities (n=4) and one hospital.

The companies/institutions are from Sweden(n=2), Spain(n=2), Australia(n=2), Netherlands(n=2), Portugal, New Zealand, U.K., Slovenia, Russia, Uruguay.

3.2 Project Properties Results

The data on the 16 projects is shown in Table 3. Most of the projects are using a relational data model approach (n=11), although XML is being stored in some as an SQL field (n=6). Other storage data models used are XML (n=2), Object Oriented, binary blob, and smart indexing.

As expected of systems that follow the relational approach, SQL databases are prefered (n=11), without any of relational database management systems (RDMSs) standing out, Microsoft SQL Server (n=5), MySQL (n=5), PostgreSQL (n=5), Oracle (n=3). Some of these include NOSQL solutions. Other systems reported were: eXist (n=2), baseX, MongoDB, Firebird, SQLite and H2. One of the projects still has not decided which database to use and one other claimed that it could use any persistence layer.

Most of the querying is being done by SQL (n=11), with some including Xpath in the query (n=4). The use of AQL is mentioned in 7 of the 16 projects, and 1 mentioned the use of a Modified AQL. Xquery (n=5) is another query formalism well represented. Two other solutions were Grails ORM DSL based on Hibernate API and Groovy GPaths.

Seven of the projects have non-commercial licensing: Apache 2.0 (n=4), GPL3, Sencha4 and GPL3+MPL. The remaining (n=7) are Commercial, Proprietary (n=1) and not definitely decided (n=1).

Service interfaces via SOAP (n=12) or REST (n=6) are common. Other answers were Java Remoting and PHP-API

Most systems support dynamic configuration using new/changed archetypes and templates dynamically without system restart (n=14). Only some systems (n=7) are using demographic archetypes. More than half the projects (n=10) are using either openEHR- version control mechanisms or ISO13606 audit classes.

The majority of the systems are up to date (n=13) with the openEHR Release Version (1.0.2) or are using ISO13606, as well as ADL version 1.4.1 (n=15).

Data collected (Table 2) shows that there are some systems deployed and in production (n=8) and others in experimental or development phase (n=5). In the deployed projects, the number of people using the system vary from a few thousand to a few hundred thousand. This variation is increased in the patient records number where it goes from a few thousand to more than ten million.

4 Discussion

The survey indicates that there are some fairly large installations of openEHR- and 13606-based systems running and there are reports of even larger ones (for 12 million inhabitants) being set up [9] . The survey found active projects in South America, Eurasia and Oceania, but none in Africa or North America.

This initial survey did not ask for performance data, but from responses, comments, mailing list postings and previous studies [6] we can assume that there are implementations that handle many clinical data access use cases with appropriate response times. On the other hand we suspect that multi-patient queries (for epidemiological studies etc) are harder to handle with good performance in systems that have not indexed leaf values contained in xml-fields etc.

Archetype based EHR data is logically tree-structured [1], and different tree-structures appear every time a new archetype starts being used. Tree structures can of course be mapped to relational models, but you would in that case want to avoid methods that involve too many joins at retrieval time and avoid methods that need manual creation of new tables when new archetypes appear.

The systems based on relational tables thus most likely handle the model impedance mismatch with various automated workarounds and indexing strategies. Obviously other experiences or benefits of RDBMS, have outweighed this mismatch inconvenience for many implementors. The fact that various RDBMS-based solutions have for a long

Table 2: Enquiry quantitative variables results

	B24	EHRF	eW	GOS	HSEAVS	LEHR	LiU	OEHR	OGen	OServ	OEyes	OH	OK	RP	Crit	TEHR
Number of installations	3	unk	0	prot	dev	3	prot	3			>10	6	dev	4	1	5
Number of users	2.5k	unk	10	prot	50k		prot	55k			1k	1k		100k	25	250k
Number of patientrecords	20k		20	samp	166m		600k	1.5m			200k	100k		2m	4k	12.5m
Database Size(GB)			6		1k		16-71				100	200		2k		125

$dev : development, prot : prototype, unk : unknown, samp : sample$

time been discussed and documented on the openEHR wiki, may also have contributed to the dominance of relational storage solutions.

Further studies comparing system performances would be desirable, but for doing that, realistic, openly accessible EHR test data, test queries and standardised test procedures would be needed. (Some of the authors are working on providing EHR test data openly.) Also the use of network-databases and RDF-triple-stores for persisting archetype based EHR data does not seem to have been extensively explored or documented yet.

The vendor Marand kindly provided some extra information regarding one of their implementation solutions - they use an inverted index implemented using Apache Lucene [10] to handle a lot of their AQL query resolution process. They report a throughput of 40.000 AQL-queries per second for a dataset with 20 million patients and 1 billion documents. To our knowledge that approach to openEHR indexing has not been widely published before.

There is a handful of archetype-based systems used in production now, but we will likely see more actors active as the market gets more mature in the coming years.

Many different kinds of storage systems are used, including ones that may not be straightforward to implement for tree-structured openEHR- and 13606-data, especially for analytic multi-patient queries used in epidemiology etc.

The mechanisms behind some approaches have been published. One way to interpret the variety in approaches is that there is currently an active ongoing search for better solutions. We hope that this survey will contribute to raising awareness of some solution alternatives and increase communication between actors.

Openly shared query-sets and data-sets combined with specified test procedures would likely make it easier to compare the performance of the different solutions. Perhaps in the form of recurring open competitions or performed by certification organizations.

5 Acknowledgments

SM Freire received a scholarship from CAPES Foundation - Brazil (Proc. 4055/11-0).

This work was also supported by Gulbenkian Foundation, through the research project "OpenObs.care - Criação de Sistemas de Informação em Saúde baseados em OpenEHR" (MP/P-125963).

Finally, we wish to thank all survey respondents for devoting time to answer our questions.

6 References

[1] T. Beale, S. Heard, "OpenEHR architecture overview", Accessed 2012 jul 20,. Available at: http://www.openehr.org/releases/1.0.2/architecture/overview.pdf

[2] ISO, "IS 13606: Health informatics Electronic healthcare record communication Part 1: Reference Model. International Organization for Standardization", 2008 p. 83. Available at: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=40784.

[3] ISO, "IS 13606: Health informatics Electronic healthcare record communication Part 2: Archetype interchange specification. International Organization for Standardization", 2008 p. 124. Available at: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=50119.

[4] [AQL] Archetype Query Language Description - Specifications - openEHR Wiki, Accessed 2013 feb 20, Available at: http://www.openehr.org/wiki/display/spec/Archetype+Query+Language+Description

[5] T. Austin, "The Development and Comparative Evaluation of Middleware and Database Architectures for the Implementation of an Electronic Healthcare Record", PhD Thesis, University College London, 2004.

[6] S.M. Freire, E. Sundvall, D. Karlsson, P. Lambrix, "Performance of XML Databases for Epidemiological Queries in Archetype-Based EHRs", Proceedings Scandinavian Conference on Health Informatics 2012, p. 51-57.

[7] Persistence FAQS - Resources - openEHR Wiki, Accessed 2013 feb 20, Available at: http://www.openehr.org/wiki/display/resources/Persistence+FAQs.

[8] Node+Path Persistence - Developers - openEHR Wiki, Accessed 2013 feb 20, Available at: http://www.openehr.org/wiki/x/NwAM

[9] Marand Think!EHR Platform, Accessed 2013 feb 10, Available at: http://www.marand-thinkmed.com/news-title.

[10] The Apache Software Foundation, "Apache Lucene, Accessed 2013 feb 20, Available at: http://lucene.apache.org/.

Table 3: Enquiry qualitative variables results

		B24	EHRF	eW	GOS	HSEAVS	LEHR	LiU	OEHR	OGen	OServ	OEyes	OH	OK	RP	Crit	TEHR	Total
Data Model	Relational			●			●		●	●		●	●					6
	XML data in relational tables	●			●	●							●		●	●		6
	XML		●				●	●						●			●	5
	Object Oriented				●	●												2
	binary blob								●									1
	smart indexing								●									1
Database engines	Microsoft SQLServer				●		●		●				●		●			5
	MySQL	●		●						●	●	●						5
	PostgreSQL						●			●	●	●					●	5
	Oracle					●										●	●	3
	eXist		●					●										2
	baseX							●										1
	MongoDB																●	1
	Firebird												●					1
	SQLite				●													1
	H2										●							1
	any																●	1
	not definately decided													●				1
Querying	SQL			●	●		●		●	●		●	●					7
	AQL							●	●			●	●	●			●	6
	Xquery		●				●	●					●	●				5
	SQL+XPath	●				●									●	●		4
	ModifiedAQL					●												1
	Grails ORM DSL on Hibernate									●								1
	XML+GroovyGPath+Indexing										●							1
Licence	Commercial	●					●		●				●		●	●	●	7
	Apache 2.0				●			●		●	●							4
	GPL3		●									●						2
	Sencha4			●														1
	MPL		●															1
	Proprietary					●												1
	not definately decided													●				1
Services	SOAP	●		●		●	●		●		●	●	●	●	●	●	●	12
	REST							●	●		●	●	●				●	6
	Java Remoting																●	1
	PHP-API	●																1
Dynamic configuration	Yes	●	●	●	●		●	●	●	●	●		●	●	●	●	●	14
	No					●						●						2
Demographic archetypes	Yes	●		●		●	●							●	●	●		7
	No		●		●			●	●	●	●	●	●				●	9
Auditing	Yes	●		●		●		●	●		●		●	●		●	●	10
	No		●		●		●					●			●			5
openEHR Release	0.96														●			1
	1.0			●														1
	1.0.1			●	●		●											3
	1.0.2/ISO13606	●	●	●		●		●	●	●	●	●	●	●		●	●	13
ADL Version	1.4.1	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	15
	N/A														●			1