HIC 2006 Bridging the Digital Divide: Clinician, consumer and computer
Eds: Johanna Westbrook & Joanne Callen
Publisher: Health Informatics Society of Australia Ltd (HISA)
ISBN 0 9751013 7 4

Evaluation of a Persistent Store for openEHR

Jon Patrick[1], Richard Ly[1], Donna Truran[2]

[1]*School of Information Technologies University of Sydney*

[2]*National Centre for the Classification in Health University of Sydney*

ABSTRACT

A logical database model for openEHR was developed to attest to the feasibility of whether openEHR's architecture is practical for an operational information system. A variant of the Entity-Attribute-Value (EAV) modelling technique and entity relationship diagrams (ERDs) are used to design the generic data structures. It was found that the generic nature of openEHR's technical architecture provided a good basis for factoring interoperability and life long health records. However, this comes at a cost as there is a significant performance tax on data insertion, validation, retrieval and querying. Furthermore, it was found that the role of standardisation in the EHR community is a vital factor in determining the success of these systems' ideal capabilities.

Keywords:

openEHR, Entity-Attribute-Value, archetype, persistent store

Introduction

While current technology may satisfy short to medium term health informatics requirements, many have the view that the satisfaction of long term requirements lies within the next generation EHRs and HIS. Commonly dubbed, "standards based systems", these heavily promote standardisation of important key areas to promote knowledge level and technical interoperability. The ideal information systems of tomorrow will have a high level of flexibility allowing any healthcare agent to access their patient's data in a point of care environment. Beale [1] identifies ten main requirements of an ideal EHR system of which the most important in this study are: Interoperability, Use of knowledge bases, and, factoring constant changes.

Bott [2] identifies that the main issue preventing harmonious interaction between health care systems lies within the differences in the array of proprietary software in the industry and lack of standardisation in the health community. Moreover, Beale [3] reinforces this notion as he argues that current "standards in health informatics tend to be judged in terms of themselves, against particular local requirements, or against each other". This is particularly evident in Australian EDs where the dominant market software package has numerous versions in situ.

Bernstein [4], who assessed the transition of current EHRs to next generation EHRs in Denmark suggests that the "prerequisite for real integration and semantic interoperability is agreement of data content and the information models." This directly refers to standardisation issues as discussed above but additionally, the need for a modelling paradigm shift.

HIC 2006 Bridging the Digital Divide: Clinician, consumer and computer
Eds: Johanna Westbrook & Joanne Callen
Publisher: Health Informatics Society of Australia Ltd (HISA)
ISBN 0 9751013 7 4

Bird [5] describes the approaches to building EHR systems via three well known methodologies; the unstructured approach, the "BIG" model approach, and the generic approach. The unstructured

approach to EHR is simply a "warehouse filled with unstructured text". This can be problematic as the system cannot be queried readily and reported for management purposes. The "BIG" model approach has a "separate table for each clinical concept" leading to excessively large schemas. While this proves effective in querying, it leads to errors as few people can completely understand the entire model. Furthermore, the model is brittle over time as it does not factor for the volatile medical domain. Finally, the generic model is designed to allow a "wide variety of data to be accommodated in a general purpose set of data structures". The model is small enough to understand conceptually, however suffers from query difficulty as the data stored is similar to that of the unstructured process. In order to overcome the problem of lower data quality from generic modelling, a constraint mechanism must be introduced to ensure that the stored information is valid in terms of the clinical domain. This is referred to as the "Archetype Model" which was initially developed by the GEHR project and later adopted by openEHR, and it is reasonably well aligned, and in some ways makes advances on the CEN 13606 standard.

The future of these information systems will heavily focus on interoperability and the delivery of a unified healthcare system where EHRs can be accessed by clinicians consistently and reliably regardless of the number of vendors. How this will be achieved lies within the standardisation of knowledge and information models. This distinctively leads to a required technical paradigm shift in the information modelling community to adopt a two level modelling approach. This two level framework is reinforced by Bernstein [4] who believes that this paradigm shift "creates the basis of moving from 1st generation EHR systems to 2nd generation system ... (which) will better support the new healthcare requirements with patient focused and shared care based on extensive re-use of information across organisations."

The aim of this project is to create an open source data model to investigate the feasibility of using the openEHR architecture (V0.96) as a viable basis for developing an interoperable EHR exemplified in an ED information system. Emergency Department (ED) information systems must satisfy *national minimum data collection* (NMDS) requirements and ideally facilitate the use of clinical terminologies to achieve a high level of interoperability. In order to demonstrate the viability of using openEHR architecture, several aspects of openEHR are separately modelled: Archetype Data Dictionary (ADD), Demographics Model (DM), Generic Data Model (GDM), and parts of the EHR Model relevant in recording clinical findings. Furthermore, other areas related to openEHR and the data model was also investigated. These include performance issues and core advantages and disadvantages of the architecture itself. Specific openEHR constituents that are not related to this research topic were not investigated, for example EHR_Extract and EHR versioning from the RM.

Methodology

openEHR Architecture

openEHR assumes a two level framework initially adopted from the Australian GEHR project called the Reference Model (RM) and the Archetype Model (AM) [6]. Conceptually, a two level framework is used to separate knowledge and information models. This facilitates interoperability on the knowledge and information level as there is no need to alter the generic data models if domain concepts changes [7].

Publisher: Health Informatics Society of Australia Ltd (HISA)
ISBN 0 9751013 7 4

Reference and Archetype Models

The RM represents the semantics of storing and processing in the system. It essentially contains a set of generic data structures that is flexible enough to model most, if not all the logical structures that occur in clinical records. As a result, the RM is designed to have a small schema that is invariant in the long run and free from any domain specific knowledge [8]. openEHR has a class of logical data structures used to represent data instances. An EHR requires structured data in the form of single values (weight, height), lists of values (blood test results), tables (visual acuity results) and trees (biochemistry results) of values. In addition, openEHR's data structure model also caters for historical values (time series of blood pressures) [9]. Ultimately, these data structures are decomposed into compounds or elements. Compounds can be further decomposed into compounds or elements.

The Archetype Model (AM) contains the knowledge enabling environment by defining domain level structure and constraints on the generic data structure residing in the RM. The medium in which these constraints are delivered is called "archetypes". Essentially, an archetype is a formalised expression describing a medical concept expressed in openEHR's Archetype Definition Language (ADL). It provides the mechanism at runtime to place constraints on EHR data instances ensuring the data entered into the system conforms to certain structure and rules. Archetypes are designed to be defined by clinicians, hence the separation of domain knowledge from concrete information capture. This allows greater flexibility as it facilitates the volatile medical world where, new medical concepts often arise or change. Furthermore, archetypes can be grouped and specialised much like inheritance from a programming perspective [10]. This allows archetypes to be modified for specific and/or local use. Like inheritance, when a specialised archetype is instantiated, openEHR's rules govern that its instantiation will contain all relevant parts related to its parent's attributes. By having such flexibility, it enables archetypes to be standardised allowing it to be interoperable across the knowledge domain.

Entity Attribute Value (EAV) Modelling

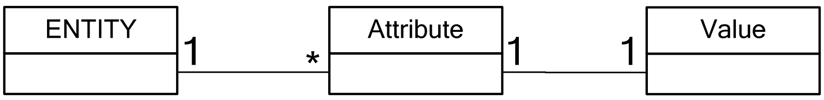

Figure 1: Basic structure for EAV modelling.

In conventional modelling techniques, each object's attribute is represented as columns. However this model proves inflexible in the long run. As the database evolves, the number of parameters grows and as a result requires the addition of new columns. This consequently leads to frequent modification of the underlying schema making it unsuitable for the medical domain [11]. Entity Attribute Value (EAV) Modelling was used as the basis for designing a suitable data model to design the core generic data structure. Conceptually EAV modelling can be visualised as a database with three tables; Entity table linked to Attribute table which in turn is linked to a Value table (Figure 1). In EAV modelling, each column in conventional schemas is represented as a row in the tables. Effectively, each object attribute is instantiated by populating the tables via rows rather than column base as seen in conventional modelling techniques. This allows the database to store a diverse range of data in only a few tables.

ISBN 0 9751013 7 4

Findings

The following section describes the results of investigating the persistent storage any system using openEHR principles would need to consider. While the design of the data models is subjective to the designers, it is evident that certain methodologies must be used and followed in order to be able to store runtime instances of EHR data. The design was developed in a modular strategy for each of the: Archetype data Dictionary, Demographics Module, Generic data Model, and Composition Package. The details of each design decision are reported in [12]. The integrated design is shown in Figure 4.

Performance

During the design of the data model, it became evident that the model may suffer from performance issues once it was in a fully operational environment. The cause of these performance issues largely arise from the use of EAV modelling and openEHR's generic nature of its data structures. Insertion, retrieval, validation and querying are the main concern in the performance of a fully operational EHR. Further more, to obtain any information from the database, there must be a significant amount of pre and post processing to decompose and reconstruct information from the generic data structures.

For example, to insert an instance of EHR information (e.g. a Blood Pressure reading) during runtime, the software layer must query and construct its corresponding archetype from an archetype repository, and then perform a comparision to make sure the data instance adheres to all constraints and rules imposed by the archetype. Once the EHR data instance is validated, the software layer must decompose the EHR instance into its entity, attributes and value instances to be inserted into the appropriate generic tables.

The performance issue is evident in the following example. A simple query to "find the NMDS information for a patient with health record number 1" requires a fairly large SQL statement (Fig. 3). The core of the performance issue revolves around the significant number of SQL joins between generic tables and the amount of filtering required to extract the appropriate results. This result demonstrates the additional complexity when querying these generic structures. For more complex queries such as "obtain health record for patient X" or aggregation queries, we would require several SQL statements in which to find the desired information.

```
SELECT
        ehr.EHR_ID, ehr.PATIENT_ID, ehr.EHR_DATE_TIME_STAMP,
        c.COMPOSITION_COMPOSER, ec.HEALTH_CARE_FACILITY,
        ec.EVENT_CONTEXT_START_TIME,
        ec.EVENT_CONTEXT_END_TIME,
        ds.DATA_STRUCTURE_TYPE, ds.DATA_STRUCTURE_NAME,
        its.ITEM_STRUCTURE_NAME, i.ITEM_NAME,
        ev.ELEMENT_VALUE, ev.ELEMENT_UNIT
FROM
        EHR AS ehr, COMPOSITION AS c, EVENT_CONTEXT AS ec,
        DATA_STRUCTURE AS ds, ITEM_STRUCTURE AS its, ITEM AS i,
        ELEMENT_VALUE AS ev
WHERE
        ehr.EHR_ID              =       c.EHR_ID
And     c.COMPOSITION_ID        =       ec.EVENT_CONTEXT_ID
And     ec.EVENT_CONTEXT_ID     =       ds.STRUCTURE_ID
And     ds.DATA_STRUCTURE_TYPE=         'EVENT_CONTEXT'
And     ds.DATA_STRUCTURE_ID =          its.ITEM_STRUCTURE_ID
And     its.ITEM_STRUCTURE_ID   =       i.ITEM_STRUCTURE_ID
And     i.ITEM_ID               =       ev.ELEMENT_VALUE_ID
And     ehr.EHR_ID              =       1;
```

Publisher: Health Informatics Society of Australia Ltd (HISA)
ISBN 0 9751013 7 4

ITEM_STRUCTURE_NAME	ITEM_NAME	ELEMENT_VALUE
NMDS	Establishment	11M[O]12345
NMDS	Health Service Event - Presentation Date	03/02/2006
NMDS	Health Service Event - Presentation time	15:00
NMDS	Non - Admitted ED Service Episode - Episode lengt	120
NMDS	Non - Admitted ED Service Episode - Transport Mode	1 (Ambulance)
NMDS	Non - Admitted ED Service Episode - Patient departure status	1 (Admitted to this hospital)
NMDS	Non - Admitted ED Service Episode - Triage Category	3 (Urgent - Within 30 minutes)
NMDS	Non - Admitted ED Service Episode - Type of Visit to ED	2 (Return Visit - Planned)
NMDS	Non - Admitted ED Service Episode - Waiting Time	20
NMDS	Compensable Status	9 (Not Stated/Not Known)
NMDS	Indigenous Status	4 (Neither Aboriginal or Torrest
NMDS	Area of usual residence, geographical area code (ASGC 2004)	12345 (Liverpool)
NMDS	Country of birth (SACC 1998)	4321 (Australia)
NMDS	Date of Birth	05/08/1921
NMDS	Person identifier	MRN10000001
NMDS	sex	1 (Male)
NMDS	Episode of care - funding eligibility indicator	1 (YES)

Figure 3: SQL statement and the returned results in prototyped database.

Accordingly, a study on the performance of EAV modelling found similar concerns[11]. They modelled database schemas using both the conventional approach and EAV approach to store 10 years of clinical microbiology data. Entity centered and attribute centered query execution times were compared under varying conditions of database size and system memory. They found that the performance was very similar for entity centered queries, however for attribute centered queries, EAV models were approximately “three to five times less efficient that it’s conventional counterpart” [11]. Furthermore, the “differences in query efficiency became slightly greater as database sized increased” indicating that this performance tax will have a substantial impact on the operation of such a system.

Conclusions

The next generation systems need to facilitate interoperability on a large scale and archetypes provide a very attractive medium for achieving it. The use of a common repository of archetypes allows information exchange between systems. Archetypes also provide the means for separating domain knowledge from data structures. The use of generic data structures makes the information systems highly adaptable for change. This is especially useful in the medical domain as concepts are continually being introduced and evolved. The result is that the table schemata do not have to be modified if domain knowledge changes.

EAV modelling, while it may cause performance issues, has many advantages. Besides being able to represent data in a generic form, it has a significantly smaller schema when compared to the schemas in conventional modelling. For database management, this is advantageous as the tables are much easier to manage. However this is more than offset by the increase in complexity of SQL statements required to query the tables. The degree of this impact requires further research with larger EHR datasets and for concurrent access of records. Furthermore, one must consider which type of querying will have the greatest performance impact on the system and how often these will be used. More advanced aggregation queries will take longer. openEHR’s models are very generic due to the separation of its knowledge model. Without a concrete data model, it will make queries much more difficult to construct and manage.

For archetypes to be useful in facilitating interoperability and sharing of consistent medical knowledge, a committee or organisation must be set up to create archetypes and manage these archetypes. Archetype creation and governance can be time consuming and difficult to manage.

HIC 2006 Bridging the Digital Divide: Clinician, consumer and computer
Eds: Johanna Westbrook & Joanne Callen
Publisher: Health Informatics Society of Australia Ltd (HISA)
ISBN 0 9751013 7 4

According to Bernstein [4], the Aarhus County (Denmark) whom are developing interoperable EHRs, had found it a "challenge to … obtain an extensive and coherent set of archetypes." An opportunity exists to reduce the amount of time and effort required to generate these archetypes. Instead of relying on clinician's medical knowledge, an ontology like SNOMED-CT could provide the basis in which a foundational set of archetypes can be computationally generated in ADL format.

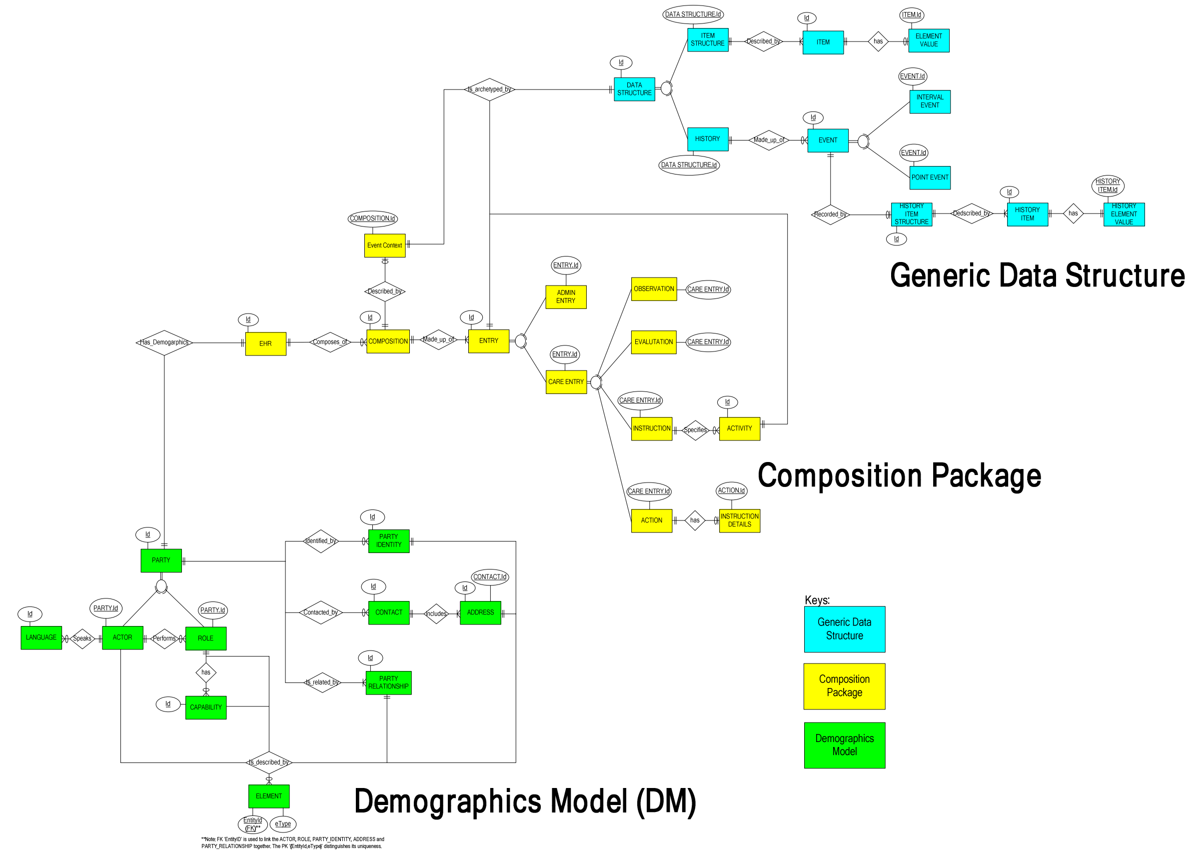

Figure 4: The openEHR Reference Model.

References

[1] Beale T. (2005). 'The Health Record – why is it so hard?', *IMIA Yrbk of Med Info 2005*, pp223-226.
[2] Bott, J. O. (2004). 'The Electronic Health Record: Standardisation and Implementation'. *Institute for Medical Informatics, Technical University of Braunschwieg,* Germany.
[3] Beale T. (2001). 'Health Information Systems Manifesto Revision 1.0'.
[4] Bernstein K., Bruun-Rasmussen M., Vingtoft S., Andersen SK., Nohr C. (2004). 'Modelling and implementing electronic health records in Denmark', *International Journal of Medical Informatics (2005)*, Vol 74, pp 213-220.
[5] Bird L., Goodchild A., Tun Z. (2003). 'Experiences with a Two-Level Modelling Approach to Electronic Health Records', *Jnl of Res and Prac In Information Technology'*, Vol. 35, No.2 pp 121-138.
[6] Beale T., Heard S., Kalra D., Lloyd D., Schloeffel P. (2006). 'Introducing openEHR', Release 1.0.
[7] Beale T. (2001). 'Archetypes Constraint Based Domain Models for Future Proofed Systems', Revision 2.2.1, pg. 16.
[8] Beale T., Heard S., Kalra D., Lloyd D. (2006). 'Reference Model: The openEHR EHR Information Model', Release 1.0 public comment
[9] Beale T., Heard S., Kalra D., Lloyd D. (2005). 'Reference Model: The openEHR Data Structures Information Model', Revision 1.6 pp9-38.
[10] Beale T., Heard S. (2005). 'Archetype Definition and Principles', Revision 0.6

HIC 2006 Bridging the Digital Divide: Clinician, consumer and computer
Eds: Johanna Westbrook & Joanne Callen
Publisher: Health Informatics Society of Australia Ltd (HISA)
ISBN 0 9751013 7 4

[11] Chan R., Nadkarni P., Marenco L., Levin F., Erdos J., Miller P. (2000). 'Exploring Performance Issues for a Clinical Database Organised Using an Entity-Attribute-Value Representation', *Journal of the American Medical Informatics Association*, Col. 7 No. 5, pp 475-487.
[12] Patrick, J. & Ly, R. (2006). Assessment of openEHR model for Emergency Department Information System. Technical Report, School of Information Technologies, Uni. of Sydney, Australia.