ON DESIGNING AN EHCR REPOSITORY

Petr Aubrecht, Kamil Matoušek and Lenka Lhotská
The Gerstner Lab, Czech Technical University in Prague, Technická 2, 166 27 Praha 6, Czech Republic

Keywords: EHCR, databases, multiagent systems.

Abstract: In an ongoing project, a pilot study and implementation of repository design for electronic home care records (EHCR) is described. Electronic home care record is based on the idea of electronic health record, however it also satisfies additional information and functionality requirements specific for home care. The design is based on the home care data and service model (K4Care model). First we analyzed the problem and decided about the platform, storage technology, cooperation with other parts of the system being developed, and basic structure of the EHCR. Then we focused on the design of data storage and transformation of the K4Care model into a database structure. Finally cooperation between the database and multiagent system is proposed.

1 INTRODUCTION

An electronic health record (EHR) is a distributed personal health record in digital format. The EHR should provide secure, real-time, patient-centric information to aid clinical decision-making by providing access to a patient's health information at the point of care. An EHR is typically accessed on a computer or over a network. An EHR almost always includes information relating to the current and historical health, medical conditions and medical tests of its subject, thus representing a longitudinal collection of information for and about patients. In addition, EHR should contain data about medical referrals, medical treatments, medications and their application, demographic information and other non-clinical administrative information.

Although there are few standards for modern day electronic records systems as a whole, there are many standards relating to specific aspects of EHRs. These include:

ASTM Continuity of Care Record (CCR) is a patient health summary standard based upon Extensible Markup Language (XML); the CCR can be created, read and interpreted by various EHR or Electronic Medical Record (EMR) systems, allowing easy interoperability between otherwise disparate entities.

American National Standards Institute (ANSI) X12 (EDI) is a set of transaction protocols used for transmitting virtually any aspect of patient data. It has become popular in the United States for transmitting billing information, because several of the transactions became required by the Health Insurance Portability and Accountability Act (HIPAA) for transmitting data to Medicare.

CEN, the European Committee for Standardization, was founded in 1961 by the national standards bodies in the European Economic Community and EFTA countries. Now CEN is contributing to the objectives of the European Union and European Economic Area with voluntary technical standards which promote free trade, the safety of workers and consumers, interoperability of networks, environmental protection, exploitation of research and development programs, and public procurement. CEN - CONTSYS (EN 13940) is a system of concepts to support continuity of care. CEN - EHRcom (EN 13606) is the European standard for the communication of information from EHR systems. CEN - HISA (EN 12967) is a services standard for inter-system communication in a clinical information environment.

DICOM is a heavily used standard for representing and communicating radiology images and reporting

Health Level Seven (HL7) is one of several ANSI-accredited standards developing organizations operating in the healthcare arena. HL7 messages are used for interchange between hospital and physician record systems and between EMR systems and practice management systems; HL7 Clinical

Aubrecht P., Matoušek K. and Lhotská L. (2008).
ON DESIGNING AN EHCR REPOSITORY.
In *Proceedings of the First International Conference on Health Informatics*, pages 280-285

Document Architecture (CDA) documents are used to communicate documents such as physician notes and other material.

Integrating the Healthcare Enterprise (IHE), while not a standard itself, is a consortial effort to integrate existing standards into a comprehensive best-practice solution

ISO TC 215 has defined the EHR, and also produced a technical specification ISO 18308 describing the requirements for EHR Architectures.

OpenEHR represents a next generation public specifications and implementations for EHR systems and communication, based on a complete separation of software and clinical models.

Various factors involving the timing, the right players, market history, utility, governance play a key role in the overall enrichment of the standard and certification development. The standardization and certification even though seem to bring uniformity in the EMR development; they do not guarantee their acceptability and sustainability in the long run.

The core of any EHR system is a data repository that is usually realized by a database system. Types of databases include the following ones (Bontempo & Saracco, 1995): hierarchical, network, relational, and object oriented. The hierarchical and network databases represent older forms that are not used in newer applications. Relational database departs significantly from those two types and is the most common form of database today. Relational databases are constructed using tables instead of tree and network structures. The tables do not specify how to retrieve the required data or navigate through predefined path. Object oriented database is the most recent approach to database management. The object oriented database structure is derived from object oriented programming and has no single inherent database structure. The structure for any given class or type of object can be anything a programmer finds useful. Furthermore, an object may contain different degrees of complexity, making use of multiple types and multiple structures.

In the next sections we will describe individual issues related to EHCR design in more details, namely persistence and structure of EHCR.

2 SCHEMA OF PERSISTENCE

In programming, persistence refers specifically to the ability to retain data structures between program executions, such as, for example, an image editing program saving complex selections or a word processor saving undo history.

This is achieved in practice by storing the data in non-volatile storage such as a file system or a relational database or an object database. Design patterns solving this problem are container based persistence, component based persistence and the Data Access Object model. When first introduced, the idea was that persistence should be an intrinsic property of the data, in contrast with the traditional approach where data is read and written to disk using imperative verbs in a programming language. This emphasis has largely disappeared, resulting in the use of persist as a transitive verb: On completion, the program persists the data.

2.1 Storage Technology

For persistence, there were two ways in our case:

- Relational database;
- XML storage.

Relational databases have long history and today are able to store data reliably and efficiently. There exist a big number of available databases both commercial and free.

XML databases are relatively new technology with promising future, allowing natural storage of data with complex structure (e.g. annotated texts). On the other hand, there is no emphasis on reliability. Also effectiveness of query execution (using XPath) is usually low.

Because the content of the database is critically important, we emphasize robustness of the storage. Another important point is amount of data to be stored, thus the ability of the technology to manage huge dataset is crucial. Therefore the chosen technology is relational database.

This requirement also affected the further decision about which database to use. We analyzed advantages and disadvantages of several potential databases. The database we typically use for storage of a big amount of data is PostgreSQL. It provides a rich query language with strong emphasize of the Structured Query Language (SQL) standard. It is proven to be reliable and has a great support for transactions. Its BSD licence is the most free and open. From the administrative point of view, PostgreSQL allows use of several languages for stored procedures including Java, which can be used for simplification of some tasks. For this database there exists a convenient graphical interface. The performance of PostgreSQL is more than sufficient.

Another database server is represented by Mysql. It is known to be very fast for read-only databases. Unfortunately, it has significant drawbacks – limited SQL and transactions supported only in one backend with a loss of performance. Practical experience shows many issues e.g. saving invalid dates, accepting syntax errors, or even unpredictable errors. Its license is not so free, the resulting product must either be of GNU General Public License (GPL) or Mysql must be bought.

There are other databases, which could be used. For example, Firebird, which is free, but with many specifics. Microsoft SQL Server faced several security issues in past and requires a specific operating system.

Big players in the field – Oracle and DB/2 are extremely expensive for our purpose. Although Oracle provides a free version, it is limited in database size.

Based on the analysis we decided to use PostgreSQL. In case of remote database access, we are going to establish a Secure Sockets Layer (SSL) connection to the database to protect the transferred data.

2.2 Overall Schema of Persistence

When we started the project, one of the key questions was the format of messages in the system. The formats discussed were the medical standards HL/7 and ENV 13606 standards. We all decided to implement these standards consistently with their purpose – only for data interchange between this system and external institutions. In detail, this communication will be operated by specialised translation software agents (Hyacinth, 1996). We suggest that import will be manual in this project, because of the small amount of medical data to import and because the implementation is too complex.

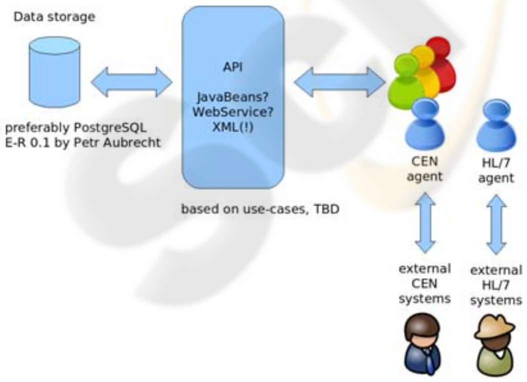

Figure 1: Schema of communication.

The format of messages within the multiagent system – MAS (Wooldridge, 2002) will depend on the concrete content of documents described in the K4Care model (Campana, 2006). The overall schema is in the Figure 1.

3 DATABASE STRUCTURE

3.1 Document Storage

For document storage there were two approaches available:

- Data oriented with all data precisely defined and stored in multiple tables;
- Document oriented with documents seen as an atomic structure with no intervention with its internal structure.

The decision is quite simple – the design of the required documents is an incremental process, documents are analyzed and their detailed structure is prepared. Moreover, we expect in the future more document types, which will lead to modification of the structure. In the first case it requires changes in the database structure and also in the API between agents and data storage.

The decision is then to store particular documents as XML documents without knowledge (from the storage/database point of view) of its internal structure. Together with documents their schema will be stored – at the moment of document storage, the schema must be known.

There is one key requirement on the document schemas – to contain a text part for agents, which will not understand the particular type, to be able to display at least some information to the human client.

3.2 Structure of EHCR

The schema evolved originally from the fundamental report of the project (Campana, 2006), describing the problem domain. We identified data, which are required to be stored and then we suggested a structure holding all the data. Later, a part of the schema, e.g. rights of particular users to some actions/documents, formed the ontology layer. A part of the schema is shared with the ontology layer. Figure 2 shows the schema of the storage.

The objects on the storage side are divided into three groups:

The first group forms EHCR, e.g. description of a patient and all his/her medical documentation. In

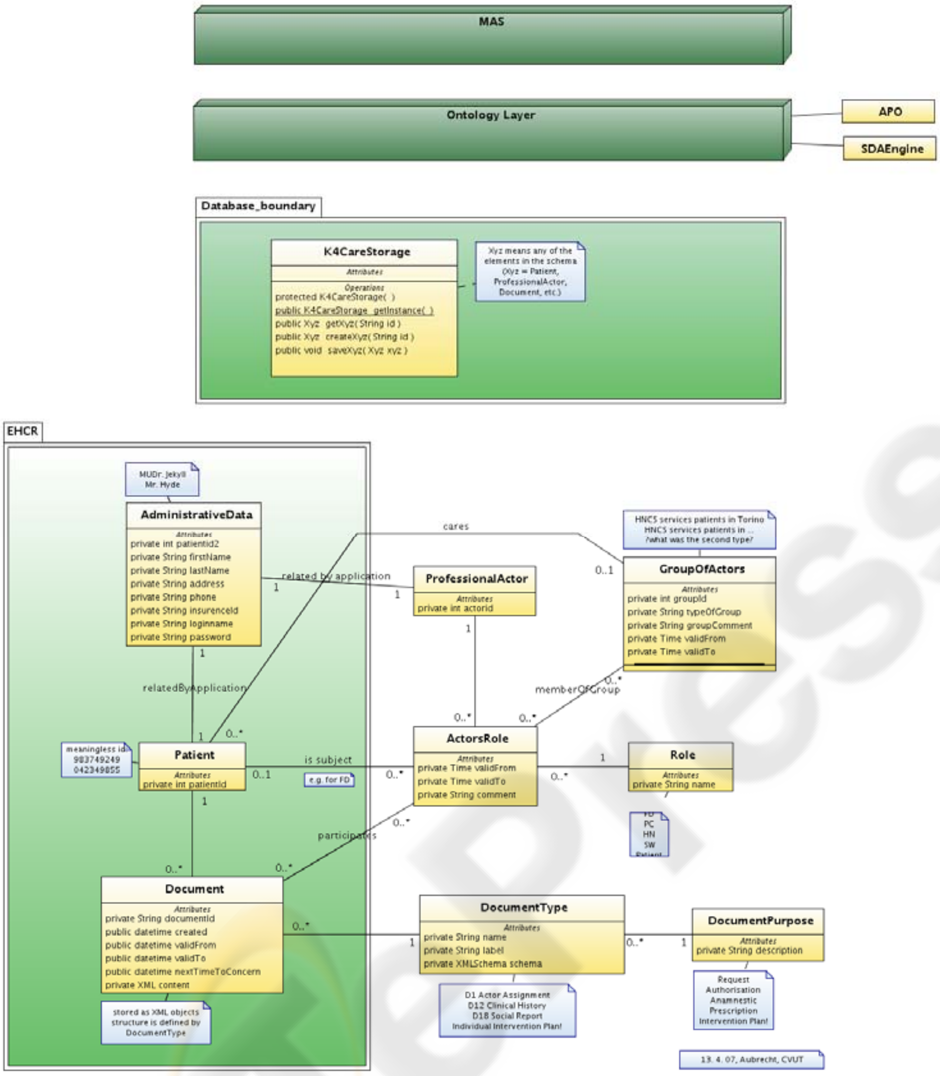

Figure 2: K4Care Model.

the schema this part is grouped into light green area. It consists of administrative data and documents. One of the requirements within the project was to separate these two kinds of information, so the relation between them is not presented in the database, but implemented in Java program by a kind of a cryptic function. These two parts can be also physically separated (in two databases, possibly on two computers) and joined by the program. Documents are stored as XML-structured texts.

The second group of objects, *ProfessionalActor*, *ActorsRole*, and *GroupOfActors*, describes the structures used in the system. Information about professionals is stored in a similar way to the patients – separately from the administrative data. An important part of these objects is dedicated to groups of the professionals and which patients are served by these groups.

The third group of objects, *DocumentType*, *DocumentPurpose*, and *Role*, is a part of schema shared with ontology layer. Information about document types and purposes and about roles of professionals must be stored in both storage (e.g. because it is required to find professionals or documents with a professional in some particular role) and ontology (these information is a part of descriptions of rules performed by the ontology layer).

Documents represent an important part of the whole schema. They are stored in XML format due to the structured content, which can be changed in the future. The corresponding XML schema is stored in the *DocumentType*, so during each save operation, the structure of the XML document can be verified. As a result, the data store does not “understand” the content of the XML documents. Their processing is done by the higher levels of the system. Only a few common attributes are stored explicitly as specific attributes. These attributes are used in document search.

Each document is assigned an attribute *NextTimeToConcern*, which allows launching some actions in the future, like appointments after some period, regular vaccination, etc.

Besides the document data, the EHCR have to store the following, so called *intervention plans*:

- *Formal Intervention Plan* (FIP). It is not going to change for a long time and belongs to a group of diseases, syndromes or symptoms (from ontology layer). FIPs will not be stored in the repository, as they are not connected to any patient. The stored FIPs may be associated to diseases, syndromes and even symptoms, described by codes proposed by medical partners using the International Classification of Diseases version 10. These codes should be used in the formal/individual intervention plan descriptions.
- *Individual Intervention Plan* (IIP) belongs to a single particular treatment of a patient. This kind of intervention plan is valid only for the time of the treatment. It is created on demand and based on a FIP. It will be attached to the EHCR of the patient.

The IIPs will be stored as documents, similarly to any other medical information. The plans including FIPs and IIPs will be expressed using SDA* model (Riaño, 2007). An interpreter and a graphical interface for creating and modifying the plans are under development.

Steps written using the SDA* model will be performed by software agents in the MAS part of the system.

3.3 Multiagent System and Database Cooperation

Cooperation between multiagent system and storage is schematically shown in figure 2. First, between these two parts is an ontology layer, responsible for checking rights and providing semantic services to the MAS.

In the upper part, there is the interface of the storage to the outside components (client side). It can be either ontology layer, as designed in the K4Care project, or directly the multiagent system.

The opposite side uses *K4CareStorage* object as an entry point to the storage and as a controller for creation and modification of all of the available objects. For each object it provides all of the four operations: creation, retrieve, update, and remove.

Objects returned are connected together using pointers to the related object, so the connection using primary and foreign keys (used in the database) is hidden to the client side and thus convenient.

Both the developers of the agents and the developers of the database may need to modify such interfaces. These modified interfaces will be handled carefully.

4 DOCUMENT SCHEMAS

Currently, XML schemas are being created to make clearer the structure of the EHCR documents planned to be used in K4Care. All the documents defined in (Campana, 2006) can be incorporated in EHCR as long as they belong to the patient.

The structure of the most relevant documents is being discussed with the medical partners. Real documents are examined and described in the form of XML Schema. Together with documents are stored intervention plans as a specific type of document (in most cases with *NextTimeToConcern* attribute set, when a next step of the plan should be performed).

In the tables 1 and 2, there are shown for illustration the most relevant service specific and common documents to be implemented in the first stage:

Table 1: Examples of the service specific documents for the first stage.

Document name	Abbreviation and type
MDE scales	D11 – anamnestic multi-dimensional evaluation
Set of forms filled in by the evaluation unit (EU) during the first problem assessment and/or in occasion of the periodical or end-treatment re-evaluation.	
Clinical history	D12 – anamnestic clinical assessment
All the available pertinent clinical information of the patient (HCP) – previous test results, discharge sheets, consultations, previous treatment. It is written by the family doctor (FD) and the physician in charge of the home care (PC); it is read by EU and the other professionals in charge of the patient (according to their competencies in the process of care of the individual patient), by the patient him/herself.	
Physical examination report	D13 – anamnestic physical examination
The report contains signs and symptoms of diseases and/or conditions written by FD, PC, specialist physician (SP); can be updated in any occasion of evaluation. It is read by FD, PC, head nurse (HN); SP and nurse (Nu) in charge of the patient.	
Medical follow-up form	D19 – anamnestic follow-up
It is written by FD or PC during the follow-up activities.	
Nursing follow-up form	D20 – anamnestic follow-up
It is written by Nu or HN during the follow-up activities.	

Table 2: Examples of the common documents for the first stage.

Document name	Abbreviation and type
Actor assignment	D01 – request
Is the information that links an individual action to individual HCP for an action (or series of actions) to be performed.	
Actor confirmation	D02 – authorization
Is the information that declares that the actor knows the assignment and accepts it.	

5 CONCLUSIONS

The initial stage of the design and development of EHCR data repository in the frame of the K4CARE project has been described. Data storage was chosen with respect to the nature of the data – absolute requirement of secure and safe way of handling. For this purpose there were used industrial standards for storage: robust relational database engine with transactions and SQL query language. In case the application will access storage remotely, it can provide SSL connection to the server.

The schema of data stored in the database reflects the needs of medical specialists. Documents themselves are stored in XML format in order to allow further evolution of their structures. Other data is stored in relational form to allow fast search and concurrent access.

The overall schema is general enough to allow cooperation with external partners using different formats and efficient inner communication within parts of the system.

ACKNOWLEDGEMENTS

The research has been supported by the EU FP6 project No. 026968 K4CARE "Knowledge-Based HomeCare eServices for an Ageing Europe" and partially by the GA CR project No. 201/05/0325 "New Methods and Tools for Knowledge Discovery in Databases".

REFERENCES

Bontempo, C. J., & Saracco, C. H. (1995). *Database Management Principles and Products,* Upper Saddle River, N.J.: Prentice-Hall PTR

Campana, F., et al. (2006). *D01 – The K4CARE Model*, interim report.

Hyacinth, S. N. (1996). Software Agents: An Overview, In *Knowledge Engineering Review,* 11(3):1–40, September 1996, Cambridge University Press.

Riaño, D. (2007). The SDA* Model: A Set Theory Approach. Machine Learning and Management of Health-Care Procedural Knowledge. In *CBMS 2007, 20th IEEE International Symposium on Computer-Based Medical Systems.* IEEE Computer Science Press.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems,* John Wiley & Sons Ltd.