Medical Informatics in a United and Healthy Europe
K.-P. Adlassnig et al. (Eds.)
IOS Press, 2009
© 2009 European Federation for Medical Informatics. All rights reserved.
doi:10.3233/978-1-60750-044-5-175

Biomedical Data Management: A Proposal Framework

Douglas TEODORO [a,1], Rémy CHOQUET [b], Emilie PASCHE [a], Julien GOBEILL [a,c], Christel DANIEL [b], Patrick RUCH [a,c], Christian LOVIS [a]
[a] *SIM, University of Geneva and University Hospitals of Geneva, Geneva, Switzerland*
[b] *INSERM, Université Paris Descartes, Paris, France*
[c] *HEG, University of Applied Science, Geneva, Switzerland*

Abstract. The expansion of biomedical knowledge, reductions in computing costs and spread of IT facilities have led to an explosion of the biomedical electronic data. However, these data are rarely integrated and analysed because of lack of tools. The integration process is complex due to technical and semantic heterogeneity as well as lack of reliability in such distributed system. In addition, for the specific case of biomedical data, privacy is a crucial constraint. This paper presents a pilot system that will be used in the European FP7 DebugIT project to integrate biomedical data from several healthcare centres across Europe.

Keywords. data integration, federated databases, view integration

1. Introduction

The last ten years have been marked by the most important increase of biomedical data in the human history [1]. Thanks to the availability of computing at decreased costs and with a concomitant expansion in computing power and facilities, these data can be accessed and processed at very large scale. Biomedical databases covers a growing part of information ranging from clinical findings to genetic structures, including social, behavioural, societal and environmental data. However, there is modest secondary usage of this information in order to improve further quality and safety of patient care.

In order to develop a system that provides access interoperability and homogeneity across countries, data sources and data types, many challenges are faced. The integration system has to cope with: lack of technical interoperability, including different hardware platforms, operating systems, database management systems, access protocols, transport formats and programming languages [2]; lack of semantic interoperability within the different data sources [3]; management of heterogeneous data quality, especially when it comes to statistical analysis, sensitive to heteroscedasticity [4]; and finally, security, privacy and confidentiality across regions and countries, which breaches can lead to unacceptable and unforeseen risks for the patient and citizen's privacy rights [5–7].

To handle the issues aforementioned, different integration systems were proposed in the literature. Usually, those systems can be classified into three different

[1] Corresponding Author: Douglas Teodoro, University Hospitals of Geneva, 4 rue Gabrielle-Perret-Gentil, 1211 Geneva, Switzerland; E-mail: douglas.teodoro@sim.hcuge.ch.

approaches: *data warehousing* such as in Genome Database (IGD) [8], ATLAS [9], BioWarehouse [10] and BioDWH [11] projects; *view integration* such as in HEMSYS [12], TSIMMIS [13], BioKleisli [14], TAMBIS [15], SIMS [16] and KRAFT [17] projects; or *link integration* such as in SRS [18], Entrez [19] and Integr8 [20] projects.

This paper focus on presenting a data integration architecture that will be used to build interoperability between heterogeneous data storages of a network of hospitals in the DebugIT (Detecting and Eliminating Bacteria Using Information Technology) [21] project. In the next section, the DebugIT Clinical Data Repository architecture is depicted. In section 3, issues concerning integration are discussed. And finally, concluding remarks are presented in the section 4.

2. The DebugIT Clinical Data Repository

The DebugIT project will use clinical and operational information from existing clinical information systems (CIS) located in several hospitals across Europe in order to advance healthcare battle against bacteria through the use of information technology. The access to these distributed and heterogeneous data will be achieved through a virtualized, fully integrated clinical data repository (CDR).

The DebugIT project requires a unique homogeneous view of the data sources, featuring transparent access. Centralizing the raw data in a permanent storage is not allowed because of ethical and privacy issues. The sources must be accessed through SPARQL and the results must be presented in the RDF format. In order to meet these requirements, the system architecture (Figure 1) is composed of three main components: wrappers, local CDRs and a central virtual CDR – a federated database instance.

Wrappers are responsible for the ETL – extract, transform and load – process. They extract the data from the local CIS, performs a model transformation from the CIS into the DebugIT model and then, load the data into the local CDR. With the data already stored, they perform still two tasks: 1) normalise the content using ontologies – NEWT for bacteria, WHO-ATC for drug, SNOMED CT for culture, Time.OWL for time attributes, among others; 2) convert the data from the local CDR "input model", an EAV/CR model, to the local CDR "interface model", which provides a customized physical schema.

The Talend suite is used in the wrappers' development. It provides a semi-automatic source representation as well as ready-to-use models for several source types, including RDMS, delimited files, xml files, among others. Additionally, it provides means to semi-automatically map the CIS content to the DebugIT model, using a friendly GUI interface.

Local CDRs are MySQL database instances that store DebugIT data extracted from the CIS. They are set up inside the demilitarised zone (DMZ) of each data provider, being the interface between the data provider and the whole system. The data stored in the local CDRs may be validated and annotated if needed. They are based on DBMS because of scalability and robustness reasons.

The local CDR contains two distinct schemas: an EAV/CR, used as an input schema for the data extracted from the CIS, and another customized for DebugIT data, used as query interface. The EAV/CR schema provides a robust layer between the CIS and the CDR, avoiding changes on both sides when one of them needs to be modified. In addition, it solves the problem of a model capable of hosting all the DebugIT data coming from several centres. The customized schema is populated with the normalized data previously stored in the EAV/CR schema. It provides the query interface for the system and solves problems with query performance and expressiveness found in the EAV/CR model.

Local CDRs are used to perform the following tasks: 1) normalise the content of the CDR; 2) build the data catalogue and information model; 3) draft the DebugIT core ontology; 4) apply data mining methods and 5) draft end point use cases. In addition, the local CDRs reduce the load on the production systems and will be of great help when dealing with privacy issues.

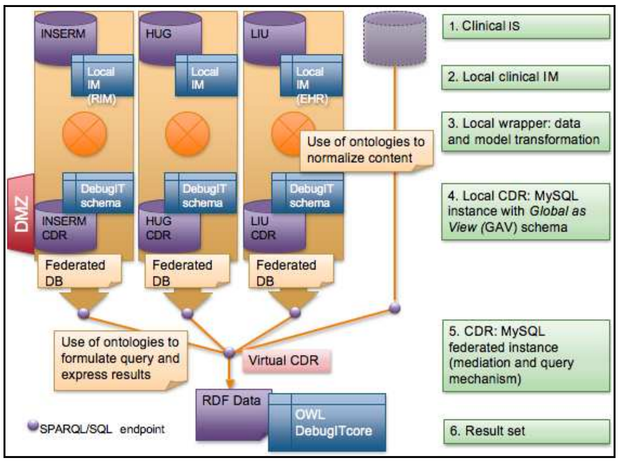
INSERM
HUG
LIU
Local IM (RIM)
Local IM
Local IM (EHR)
Use of ontologies to normalize content
DebugIT schema
INSERM CDR
HUG CDR
LIU CDR
DMZ
Federated DB
Use of ontologies to formulate query and express results
Virtual CDR
RDF Data
OWL DebugITcore
SPARQL/SQL endpoint
1. Clinical IS
2. Local clinical IM
3. Local wrapper: data and model transformation
4. Local CDR: MySQL instance with Global as View (GAV) schema
5. CDR: MySQL federated instance (mediation and query mechanism)
6. Result set

Figure 1. DebugIT CDR architecture

The core of the system is a MySQL *federated database instance*. It links all the data sources in order to create a global view on top of them. With that, it enables the DebugIT data to be accessed through a single query point but without storing any data centrally. The CDR as well as the local CDRs are being wrapped as web data services using D2R in order to provide SPARQL endpoints.

The system uses a global schema, thus following the tightly coupled integration approach [2]. Global schema is the easiest way and the key component for the semantic interoperability when ready-to-use global ontology is not present. The common schema is able to accommodate all the data pertinent to the project and their relations.

The data flow in the system can be divided into two parts, *CIS → local CDR* and *local CDRs → CDR*, as shown in Figure 2 and Figure 3 respectively.

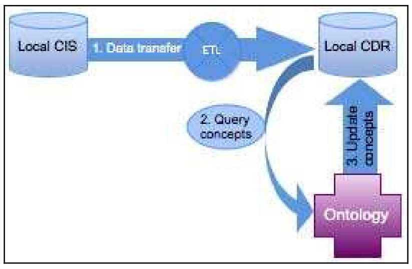

Figure 2. The data are extracted from the local CIS and stored into the local CDR. Then they are normalised using ontology concepts.

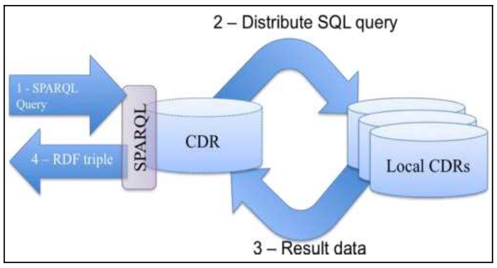

Figure 3. The CDR receives a SPARQL query that is transformed in the standard information model, using D2R. Then, the query is sent out to the local CDRs and the result is presented in the RDF format.

The system architecture has characteristics of both data warehousing and view integration approaches. Data transformation and common schema are typical of data warehousing while the inexistence of central storage is the main characteristic of view integration. Thus, the CDR presents up and down sides of those approaches. In the up side, there are high performance (data warehousing) and no data exposure (view integration). In the downside there are cost of data model updates (data warehousing and view integration) and data synchronisation (data warehousing).

3. Discussions

The major promoter of data integration process is to present to the end user a homogeneous view of the data, containing essential features of each individual system, in a unified system. As a consequence, the system must be able to handle complex queries that are spanned across the entire source datasets. From the analytical side, knowledge of the existence, physical location, access mechanism or schema of the underlying local databases shall not be required.

To have a flexible system, the content of the local databases must be autonomously and locally maintainable. Those schemas change quickly on average, two or three times per year. The databases are designed and maintained to meet local needs and changes are made independently of the integrated structure.

In addition, it is interesting to have an automatic or semi-automatic source representation. It would help the integration of new sources and would be very important for systems in which the sources sites have frequent schema updates.

The design of DebugIT CDR tries to follow these concepts mentioned above. Table 1 illustrates some comparative aspects of relevant integration systems found in the literature in the last two decades against the DebugIT CDR:

Table 1. Comparison of data integration systems

System	Year	Approach	Query interface	Downside	Upside
HEMSYS	1987	View integration	HEMSYS query language	Does not handle transactions.	Update via global schema.
SRS	1993	Link integration	Query form/API-keyword based	Source specified by user.	Easy to use and performance.
TSIMMIS	1994	View integration	LOREL	Not a full integration system.	Ease development of mediator.
TAMBIS	1996	View integration	Concept navigation	Unable to specify source. Inflexible schema change.	Usage transparency. Easy to query.
VOTES	2005	View integration	SQL	Based in a unique key identification across all data sources.	Information security.
caGrid	2005	View integration	Query API	Performance.	Processing of workflows.
@neurIST	2007	View integration	SPARQL	Schema maintenance.	Information security.
BioDWH	2007	Data warehousing	Query API	Integrates only RDB sources.	Data synchronization.
DebugIT CDR	2009	Mix approach	SPARQL/SQL	Data synchronization.	Robustness and response time.

4. Conclusion

In this paper, the pilot architecture for the CDR to be used in the DebugIT project was described. It constitutes a simple data integration model via the use of the MySQL federated engine to integrate distributed data and Talend and ontologies for homogenising models and data types. Despite of that, it has to be evaluated if the facilities provided by the federated approach are worth in performance and easiness of implementation when compared to the rigidness of the single conceptual model and the high load on the wrappers that they bring.

The next step towards the implementation of the integration system is to finalize the semantic interoperability and attack the privacy issue with creative solutions.

References

[1] Galperin, M.Y. (2008) The molecular biology database collection: 2008 update. *Nucleic Acids Research* 36:D2–D4.

[2] Sheth, A.P., Larson, J.A. (1990) Federated database systems for managing distributed, heterogeneous and autonomous databases. *ACM Computing Surveys* 22:183–236.

[3] Karasavvas, K.A., Baldock, R., Burger, A. (2004) Bioinformatics integration and agent technology. *Journal of Biomedical Informatics* 37:205–219.

[4] Mattes, W.B. et al. (2004) Database development in toxicogenomics: Issues and efforts. *Environmental Health Perspectives* 112:495–505.

[5] Sinnott, R.O., Stell, A.J., Ajayi, O. (2008) Supporting grid-based clinical trials in Scotland. *Health Informatics Journal* 14:79–93.

[6] Saltz, J. et al. (2006) caGrid: Design and implementation of the core architecture of the cancer biomedical informatics grid. *Bioinformatics* 22:1910–1916.

[7] Iavindrasana, J. et al. (2007) Design of a decentralized reusable research database architecture to support data acquisition in large research projects. *Studies in Health Technology and Informatics* 129:325–329.

[8] Ritter, O., Kocab, P., Senger, M., Wolf, D., Suhai, S. (1994) Prototype implementation of the integrated genomic database. *Computers and Biomedical Research* 27:97–115.

[9] Shah, S.P. et al. (2005) Atlas–A data warehouse for integrative bioinformatics. *BMC Bioinformatics* 6:34.

[10] Lee et al. (2006) BioWarehouse: A bioinformatics database warehouse toolkit. *BMC Bioinformatics* 7:170.

[11] Töpel, T., Kormeier, B., Klassen, A., Hofestädt, R. (2008) BioDWH: A data warehouse kit for life science data integration. *Journal of Integrative Bioinformatics* 5:93.

[12] Pillai, S.V., Gudipati, R., Lilien, L. (1987) Design issues and an architecture for a heterogenous multi-database system. *Proceedings of the 15th ACM Computer Science Conference*, ACM, St. Louis, 74–79.

[13] Molina, H.G. et al. (1997) The TSIMMIS approach to mediation: Data models and languages. *Journal of Intelligent Information Systems* 8:117–132.

[14] Davidson, S.B., Overton, C., Tannen, V., Wong, L. (1997) BioKleisli: A digital library for biomedical researchers. *International Journal of Digital Libraries* 1:36–53.

[15] Goble, C.A. et al. (2001) Transparent access to multiple bioinformatics information sources. *IBM Systems Journal* 40:532–551.

[16] Arens, Y., Knoblock, C.A., Hsu, C. (1996) Query processing in the SIMS information mediator. In Tate, A. (Ed.) *Advanced Planning Technology*, AAAI Press, Menlo Park, CA, 61–69.

[17] Preece, A.D. et al. (1999) The KRAFT architecture for knowledge fusion and transformation. In *Proceedings of 19th SGES Conference on Knowledge-Based Systems and Applied AI,* Springer, Berlin.

[18] Etzold, T., Argos, P. (1993) SRS – An indexing and retrieval tool for flat file data libraries. *Computer Applications in the Biosciences* 9: 49–57.

[19] Schuler, G.D. et al. (1996) Entrez: Molecular biology database and retrieval system. *Computer Methods for Macromolecular Sequence Analysis* 266:141–162.

[20] Kersey, P., Bower, L., Morris, L. et al. (2005) Integr8 and genome reviews: Integrated views of complete genomes and proteomes. *Nucleic Acids Research* 33:D297–D302.

[21] Lovis, C. et al. (2008) DebugIT for patient safety – Improving the treatment with antibiotics through multimedia data mining of heterogeneous clinical data. *Studies in Health Technology and Informatics* 136:641–646.