Unified data access framework for integrated systems

Karol Matiaško, Katarína Zábovská, Michal Zábovský

University of Zilina, Faculty of Management Science and Informatics,

e-mail: karol.matiasko@fri.utc.sk, katarina.zabovska@fri.utc.sk, michal.zabovsky@fri.utc.sk

Abstract: *In this paper we are introducing the concept for the unified data access framework. The main aim of our work is to allow the unified data access on the international level for educational, commercial and security purposes. The idea of the unified access is important in the current days mostly for the national/international security, international labor policy, market restrictions or diseases prevention.*

Keywords*:* RDBMS, XML, web, web services, semantic web services, data access

1. Introduction

Nowadays, Information systems integrated many databases. Each of the database is used as basic information storage elements. There are is possible recognized hundreds of databases owned by companies, state authorities and other subjects. Information stored in each of them is similar for each country, mostly in databases used by national authorities. Currently and very often they are not interconnected, since the law restrictions or the different implementation.

In this paper we specify the framework for the unified database access, which could be used for ease information exchange.

Requested interoperability may be defined as the ability for multiple software components to interact regardless of their implementation, programming language or hardware platform. The available mechanisms for software interoperability are [5]:

- Data-type interoperability: distributed and disparate programs support structured exchange of information through Application Programming Interfaces (APIs) invoked over a computer network.
- Specification-level interoperability: the same as the previous one but encapsulates knowledge representation differences at the level of abstract data types (e.g. a Table, Tree etc.). This enables programs to communicate at higher levels of abstraction and increases the degree of information hiding. CORBA and Enterprise Java Beans (EJB) fall into this category.
- Semantic interoperability: unlike the above two types of interoperability which are concerned with the form (structured description) at the integration interface, semantic interoperability represents design intent and predicted behavior as well as the form of the shared entities. It assumes that different information sources store information on related issues but each may offer a different meaning (semantic) of it.

The semantic interoperability problem may be defined in general as “the ability of user to access, consistently and coherently, similar (though autonomously defined and managed) classes of digital objects and service distributed across heterogeneous repositories, with federating or mediating software compensating for site-by-site variations" [5]. Thus the semantic interoperability between various heterogeneous information sources continues to pose serious challenges to database, artificial intelligence and other related communities [1].

The aims of our framework are as follows:

- To integrate information from multiple data sources quickly and easily with less programming and adopt application access to data.
- To transform, integrate or aggregate data easily.
- To provide possibility to create scalable, service oriented architecture without disruption or major up-front investment.

2. Problem definition

Existing approaches for integration of heterogeneous databases include resolution of structural differences between underlying databases. In conceptual or global schema approach, there is a need for standardization of data structures and definitions [5, 6]. Typical problems of this approach are [5]:

- The emphasis is given on schematic (i.e. syntactic) rather than semantic heterogeneity.
- It assumes global knowledge is available which is not always possible
- The development costs of coding the system and the data definitions is huge
- The autonomy of individual databases is lost

Additionally, each local database provides an export schema (a portion of its overall schema) which it is willing to share with other sources. It implies the need for good security background of designed solution.

By the problems specified above we can call this approach *centralized* architecture (because of existence of global knowledge – scheme). We will transform this kind of model into the *decentralized* model during the next phases of our development. Because of the complexity of mentioned transformation, we need to realize specific research for this problem area. Main research activities will be concerned with solutions of following problems [5]:

- It is very difficult to represent semantic information outside a specific domain. Even in a specific domain, human (user) intelligence is usually required to be aware of the context so as to assess semantic information such as in a federated approach.
- A specific action can have different results depending on the context.
- When using a global schema, it is hard to maintain the autonomy of individual databases and keep the overall system's development costs as low as possible.
- Application-specific solutions that compromise the autonomy, flexibility and scalability of the entire distributed application should be avoided.

We consider using results from research for continuous implementation. Extensions would be involved in the framework and framework will be used in the real environment (such as educational information exchange etc.).

3. Framework architecture

Current technologies allow us to create some kind of service oriented framework for database interconnection. First we decided to create framework using service oriented concept. This solution should be easy adopted by existing systems, even for data sources or for the applications in the consumer roles

The whole framework architecture could be divided into the following parts:

- Data transformation framework
- Communication framework
- Security and access control extensions
- Intelligent services.

We will follow the items above in our research and development. For better efficiency we are going to establish two major branches of the development. First one will be focused to *communication architecture* and *security extensions*. The second will be concerned to *database interfaces* (e.g. transformation, data semantic, intelligent data/information processing etc.).

3.1 Communication framework

We have already mentioned that our framework architecture will be based on services. This approach follows our expectations for the transparent information sharing. It is important to emphasize, that we are concerning to information not to data. This approach will be more evident in the database interfaces specification and is main motivation for the intelligent services research.

Since we are able to use well defined and standardized techniques and technologies, we decided to build whole communication architecture on the web services basics. The *web services* architecture is based mostly on standards or de-facto standards and allows us to use XML language for the information exchange. Dynamic character of our framework and its semantic orientation must be also followed in the web services concept. Semantic web services concept must be implemented and then major issues of this scenario are [7]:

- Automatic web service discovery
- Automatic web service composition
- Automatic web service execution

In this context, the web services oriented framework aims at providing an appropriate conceptual model for developing and describing services and their composition in order to support the major issues presented above. Major architectural elements for the semantic web services are [7]:

1. Ontologies
2. Goal repositories
3. Web services description
4. Mediation

Ontologies interweave human understanding of symbols with their machine-processability, gluing together two essential aspects:

1. Ontologies provide a formal semantic for information, consequently allowing information processing by a computer.
2. Ontologies provide real-world semantics, which make it possible to link machine-processable content with meaning for humans based on consensual terminologies.

Goal repositories store requester goals. A goal specifies the objective a given requester might have when looking for a service. A goal specification consists of two elements:

1. Pre-conditions describe what a web service expects for enabling it to provide its service, i.e. requirements over the input.
2. Post-conditions describe what a web service returns in response to its input, i.e. the relationship between the input and output.

Web services description give details about concrete web service. This concrete service description relies on the previous two elements: ontologies provide the necessary terminology, and goal repositories express (more generic) goals that are fulfilled by the service.

Mediation is the process of enabling heterogeneous parties to automatically inter-operate. Several kinds of mediation are identified:

- Vocabularies have to be mediated through ontology mapping, which ensures heterogeneous services can be invoked and composed.
- Process mediation deals with different interaction styles and conversation patterns.
- Mediation of message exchange protocols overcomes the differences on underlying messaging protocols between the communication parties.
- Dynamic service invocation, which takes care of providing the invocation of appropriate services at run-time to fulfill a declarative goal specified by the requester at design-time.

3.2 Security extensions

Security extensions of the framework are critical part for the real-live implementation. Aggregation of data from different locations gives the user powerful information source and the potential security compromising leads to serious problems in the international level. Hence we are using security design principles in the whole time of the framework design. The secure distributed architecture is discussed also in [8].

3.3 Intelligent services

The most important research phase for unified data access framework is creation of the artificial intelligence extensions for semantic data access architecture. If we want to transform centralized model based on well formed centralized data description structure, we must simulate “human activity” for the information recognition inside data. For very common used data quite simple model should work well, but construction of this framework portion will be critical for the rest of the framework and then it must be designed and implemented very carefully. We want to implement this important framework part by using techniques for mathematical modeling, genetic and evolution programming and neural networks implementation.

4. Data interfaces

Existing databases are using different structures for the same information. The differences are caused by different national requirements, habits and history. Data stored inside these database systems is under strong pressure to be accessible directly in the semantic form. We are able to provide XML mapping for data to become information. This approach will be implemented in the first phase of our framework development. By this mapping we want to make environment transparent and the architectural dependencies manageable on the local level. Then the main, semantic model should be managed separately on the conceptual level.

Imagine the situation that we are going to interconnect databases used by police departments for the car information holding. We will find out that the structures for different databases are almost the same. E.g. for common car features is easy to define common data structure. Hence we can define transformation *T* by which is possible to transform dataset *Ds* for the database *D* into the unified structure/form *U.*

Previous example should work quite well for other databases (e.g. citizens, animals etc.). But we must have in our mind that is impossible to define complete transformation, because of impossibility to transform semantic of different languages. So we must create model which will solve problem of the maximal coverage of the specified data set. By this solution we will be able to provide maximal possible level of automation and hence the user interaction required by the system will be minimized.

Main research activities are concerned to previously described “information recognition”. Because it is not task for common systems based on the mathematical model solution, we will implement solution based on artificial intelligence concept. This approach would provide “intelligent transformations”. It means that our database interfaces will be able to learn and they will be able to adapt current unified structure by the user actions.

Our unified structure could be defined by the XML document. XML structure allows us to define structured information and then we can use standardized techniques for the data structure checking and for the appropriate data transformation. XML based architecture could be used for the distributed architectures as well.

5. General model requirements

Current research on the field of semantic webs gives us good background for our database interfaces design. Used techniques are based mostly on XML language. XML, as the subset of SGML, provides well designed tool for the structured documents definition. By using XML we are able to describe data in data source, verify data transferred from/to unified framework and quite easy manipulate data in the heterogeneous environment. XML documents have a mechanism for self-description as well: the DTD.

Traditional databases are strongly typed. The producer and consumer have prior agreement on the structure of the information units. Since current necessities for information interchange it is not sufficient to use this concept for the long-term. The semantic orientation, and hence information exchange, is much more interesting approach. The unified data access framework must be based on facility that can expand as a human need expands. Hence information must be provided without dependency to the current representation.

Since the problems with the abstract model definition, we must define fundamental constraint for our basic research and development. In the very first phase we have to specify the syntactic constraint called *well-forming*. This constraint is essential tool for allowing information to include extended information while remaining processable by originated “down-level” database systems [4]. Future work will be oriented to minimization of the *well–defined* environment especially from the user’s point of view. This definition would be replaced by artificial intelligence subsystem.

XML based data representation is quite easy because any given XML document is finite, as is any table in a relational database. We can then specify some restrictions for our solution.

6. ALGOritmization of using data access framework

The unified data access framework must provide a way of exposing information from different systems. These systems may use a variety of internal data models so this implies a requirement for some generic concept of data at a low level that is in common between each system.

Another challenge is to support the mapping of the existing and the future database systems, preserving the universality and also properties of the local systems. Fortunately, XML concept is very closely connected with the relational database model. We should expect that the basic structures that support serializing relational databases can be shared with our framework.

Main activities of the unified data access framework research and development could be divided into two parts – communication framework development and the database interface definition. First part is mostly applied part; second one is more concerned with the research and its implemented outputs.

We suppose the following phases for the communication interface development:

- *Phase 1:* To create communication background for the Framework. The communication architecture will be based on web services and will include exchange interfaces (for the remote system connection) and information interfaces (for exchange system and information storage connection). Communication architecture must be accessible for different types of target devices e.g. personal computers, PDAs, cellular phones etc.
- *Phase 2:* To create SDK (Software Development Kit) for ease-of-use. Initial version could be implemented without GUI (Graphical User Interface) by XML, XSLT and DTD specifications.
- *Phase 3:* The main aim of this phase will be to extend existing communication architecture for security and access control mechanisms. The aim of this part must be followed in the previous phase, because security extension will be fundamental part of the whole architecture.
- *Phase 4:* To create business strategy for the application framework providing.
- *Phase 5:* To extend implemented framework for non service oriented functionality. Also much more sophisticated SDK, GUI based, should be implemented here.

Research tasks mentioned in the data interface section could be solved in following steps:

- *Phase 1:* To define transformation rules for the very common information – citizens, cars, students, animals. This phase consists of the following steps:
  1. XML structure specification
  2. DTD specification
  3. XSL transformation specification for frequently used RDMS (e.g. Oracle, MS SQL, Informix, MySQL, PostgreSQL). Also SQL dialect transformation must be solved here.
  4. Connection to the WS provider.

  Each step must follow the idea of the semantic oriented architecture.
- *Phase 2:* To prepare basic SDK (Software Development Kit) for the ease-of-use of the architecture from the Phase 1. SDK must be platform independent.

- *Phase 3:* To extend or re-implement architecture from the Phase 1. To add subsystems for
  1. information recognition (expert system definition)
  2. artificial intelligence

1. CONCLUSION

We decided to verify and demonstrate our concept on the solution for the student's information exchange. The main reason is that participation of educational institutions on this project allows us to use significant data background for our solution. To make sense to our solution, architecture will be initially designed for the international students/teachers information exchange and as a background solution for the e-learning extensions.

REFERENCES

[1] Ouksel, M. A.: A Framework for Scalable Agent Architecture of Cooperating Heterogeneous Knowledge Sources, Intelligent Information Agents, Mathias Klush (Ed.), Springer, 1999, pp 100-124

[2] Martincová, P., Matiaško, K.: Query processing decomposition principles and parallel scheduling of the execution. In: Studies of the Faculty of Management Science and Informatics. Vol. 8, 1999, pp 43-57. ISBN 80-7100-701-3

[3] Zábovský, M.: Web Services – The Way to Integrated Future. In Journal of Information, Control and Management Systems, Vol. 1, 2003, pp 89-92. ISSN 1336-1716

[4] Berners-Lee T., Connoly D., Swick R. R.: Web Architecture: Describing and Exchanging Data, W3C Note 7 June 1999, http://www.w3.org/1999/06/07-WebData

[5] Ganguly P., Rabhi A. F., Ray K. P.: The Semantic Interpreter Pattern, http://www.sistm.unsw.edu.au/people/rabhi/publications/koala01.pdf

[6] Ramesh v. C. K., Quirologico S., Silva M.: An Intelligent Agent-based Architecture for Interoperability among Heterogeneous Medical Databases, http://hsb.baylor.edu/ramsower/ais.ac.96/papers/ramesh2.htm

[7] Bruijn de J., Lara R., Arroyo S., Gomez J. M., Han S-K., Fensel D.: A Unified Semantic Web Services Architecture based on WSMF and UMPL, http://deri.semanticweb.org

[8] Ansper A., Buldas A., Freudenthal M., Willemson J.: Scalable and Efficient PKI for Inter-Organizational Communication