Uniwersytet Ekonomiczny w Katowicach

Studia Ekonomiczne. Zeszyty Naukowe
Uniwersytetu Ekonomicznego w Katowicach
ISSN 2083-8611 Nr 308 · 2016

Informatyka i Ekonometria 8

Wojciech Górka

Instytut Technik Innowacyjnych EMAG
wgorka@ibemag.pl

Michał Socha

Instytut Technik Innowacyjnych EMAG
msocha@ibemag.pl

Adam Piasecki

Instytut Technik Innowacyjnych EMAG
apiasecki@ibemag.pl

THE USE OF D2RQ IN THE INTEGRATION OF DEVELOPMENT TOOLS

Summary: The aim of the article is to present an approach to the integration of heterogeneous data sets from independent computer systems. The article presents the use of integration by means of the virtualization of data sets in a consistent semantic graph. The integration was carried out in accordance with the ATOM methodology, which had been worked out earlier, using the D2QR tool. The tool is presented in the paper as a representative of a class of tools for integrating multiple heterogeneous data sources into one coherent data space. The article describes the experience of using the various options of D2RQ for integration of data sources and different approaches to the issues related to semantic integration. The authors also presented their modification of the tool and suggestions for its improvement. The research and experience are directly related to the CCMODE project (Common Criteria compliant Modular Open IT security Development Environment).

Keywords: semantic integration, development tools, semantic web, D2RQ.

Introduction

Purpose of semantic integration

During the CCMODE project it was necessary to build a development environment. One of the objectives was to make the environment on the basis of many different types of tools (already available on the market) cooperating with

each other, as if it were a coherent system. One of the requirements of the project was also to provide the possibility to easily expand the environment. So it precludes the integration based on hard-coded links between different systems. Building a heterogeneous environment with available tools does not allow to introduce any major changes to the components of the environment. Even if most of the components would be open source tools, it would overwhelm the main objective of the CCMODE project by its complexity. Larger changes would require an increased effort associated with getting to know the technical details of the modified tools. A compromise solution was to create and use a database schema in which each tool saved its configuration and data.

Another important objective is the way of using the data. Different tools work in order to support the user in achieving a particular purpose – running a software development project. Each system has part of "knowledge" about the project. Some of the data in each system are duplicated and some complementary. The development environment is also needed to use the data collected in the various systems [Dittrich, Ziegler, 2004]. It is therefore necessary to establish a common data space, allowing to look at the data stored in these systems in a consistent and uniform manner [Levy, 2000; Dittrich, Ziegler, 2004].

The best way to deal with a variety of data and their meaning, according to the authors, seemed to be semantic integration. The use of semantic integration techniques will give meaning to individual atomic data elements. This will allow to create a homogeneous space where data can be searched and analyzed. This approach allows to solve the problem of data redundancy, integration and metadata [Magnani, Montesi, 2007; Press, 2008]. The semantic approach to integration allows to shift the burden of the work from the tasks issues involved in combining the data to the tasks associated with the increased usability of integrated data.

1. State of the art in the field of integration

In the area of systems integration there are various solutions available now [Lenzerini, 2002]. Depending on the needs and technological advancement different solutions [Bussler, 2003] are applied:

– simple data export and import made by the user or automated using batch scripts, special agents or schedulers etc. In [Bussler, 2003] called Point-to-Point integration;

– one-to-one integration based on direct database connections to the system which is integrated. In [Bussler, 2003] is Hub-and-Spoke integration;
– data bus based on the exchange of messages between integrated systems, using middleware software. In [Bussler, 2003] named Process-Based integration.

The first option is the simplest and basic. It allows to integrate two different systems. The condition is that these systems (or applications) provide possibilities to export and import data. Depending on the way the implementation is made, the participation of the operator is necessary (who will export the data from one system and import to another), or it can be done automatically with batch scripts – and running them with the appropriate schedule or using special software agents. Such a process leads to duplication of data – integrated information is simply copied from one system to another and is strongly influenced by the demand for current data – the data is up-to-date but only when copying. After the coping process is completed, the data become obsolete. It should be noted that while using such data, historical data are used. The accuracy and validity of this information depends on the frequency of export and import operations [Hull, Zhou, 1996]. This is the fundamental weakness of this method. However, it is still quite widely applied because of its simplicity and easy use. Additionally, the export-import method is time consuming and the integration process should take into account normally functioning systems.

Integration at the level of the database requires that the integrated systems are implemented based on a relational database [Auer et al., 2009]. The requirement is the knowledge of the database schema and the manner to use the database by the integrated system. This method of integration has the potential to ensure the consistency of data and their full synchronization [Bussler, 2003]. It is important, however, that the integrated system actually saved the data into the database – here it is all about situations of using some elements of the data cache. The weakness of this approach is the integration of the two particular points – the two systems that have to be integrated. Another problem is the lack of response to changes. If the source system changes, it does not notify another system about some data immediately. It is not necessary to use such functionality in every case but sometimes it is useful. Of course, we can deal with this problem using database triggers or cyclic polling but it is related to changes in the database schema or additional queries have to be executed – it is extra load of the database.

Another way to integrate is to use the data bus. This solution has many advantages. It does not bind together two integrated sites, but allows data exchange between any systems. Each system can "bind" to the data bus and provide some

functionality. The data are exchanged in the form of messages, so at the time of the data change the information is disseminated to all interested systems. Building a data bus enables to integrate systems in different ways. The solution is open to new ideas and does not require fundamental changes, once you change the functionality of systems integration, development (adding a new system). This type of integration, however, requires additional middleware software and implementation of modules that provide and consume the data for the data bus in each of the integrated systems.

A relatively new idea is semantic data integration [Dittrich, Ziegler, 2004]. It is a kind of compromise between integration based on the database and integration which uses a data bus. The semantic integration is based on the construction of a common data space, based on the engine that maps each database [Lenzerini, 2002; Press, 2008] to a relational form of an RDF graph [Bizer, Seaborne, 2004], in accordance with a defined ontology [Magnani, Montesi, 2007]. Mapping is intended to provide information on how to "translate" the data from a relational database to RDF triple [Press, 2008]. Mapping is not just about one database – it is being developed globally and thus it is possible to provide multiple databases in a uniform way. Database schemas are thus aligned in such a way that the same data (yet existing in different databases and tables) are represented by the same class of data and its properties. Additionally, the data that are complementary but exist in different data sources (different databases with different schemas) are presented and made available together. So we get an integration method providing synchronous access to multiple databases without the need to copy data or to have a cyclic operation of the export-import. The data are at once integrated with each other in terms of their relevance. This method of integration is flexible – it is possible to extend the data on new data sources without interfering with the existing structures and relationships. It is also possible to use the data for various applications wishing to make use of them. It is thus a way of integration in terms of flexibility similar to that of using the data bus. The difference is the method of exchanging data – in the case of the data bus the data exchange is asynchronous, while in the case of data space it is synchronous. In the latter case a request translates directly into a series of queries to connected databases [Press, 2008]. It is not a good solution for exchanging messages between systems – in the case when it is necessary to immediately notify systems between each other. However, the solution seems to be perfect to get data from other systems (data collection on request).

There are a number of products that support building a virtual data space and semantic integration [Auer et al., 2009]. The authors of this publication have drawn attention to the following:

- D2RQ [Bizer, Seaborne, 2004] – an open source freeware tool that allows to build the mapping from relational databases to RDF triples [Auer et al., 2009]. It enables to poll the data using the SPARQL language as well as to reach out directly to the semantic data through the HTTP protocol and properly constructed URL references [Auer et al., 2009] (Rest API).
- Virtuoso [Erling, Mikhailov, 2009] – a commercial solution that comprehensively assists the user in the integration of various data sources; the tool enables to share data both as semantic data possible to poll using the SPARQL language and relational data. It is also possible to interleave SQL and SPARQL queries.
- JBoss Teiid – an open source freeware tool that integrates multiple databases in one data space. It is not directly related to Semantic Web. An integrated data space is provided as another virtual database that can be queried by SQL queries using a JDBC driver.

During the development of the CCMODE system the authors decided to choose the D2RQ tool. This solution is available free of charge with a source code and technical documentation. Thanks to this, the tool is flexible in the use and development. In addition, the D2RQ architecture allows to build scalable solutions appropriate for the intended load. The preliminary analysis showed that D2RQ allows to change the functionality without having to interfere with the source code. The changes can be made by embedding the original D2RQ code in our own code functionality. The activity of D2RQ relies on the mapping between the source schemas and databases made available to the RDF graph. The mapping used in D2RQ became the basis for the W3C recommendation *R2RML: RDB to RDF Mapping Language* [www 1]. The recommendation was published by W3C when the CCMODE project was already in the final phase. Therefore, the article does not refer to this recommendation, as our experiences are related only to the mapping language supplied together with D2RQ.

Regardless of the work carried out in the CCMODE project, the authors trace the evolution of semantic integration capabilities and analyze the W3C recommendations in this area. Next projects related to the issue of integration will take into account the possibility of using the latest W3C recommendations. The following part of the article will describe the experiences of using this tool to integrate systems in the CCMODE project.

2. How to use D2RQ

As part of the CCMODE system it was necessary to integrate multiple systems to support software development. Most of these systems were built based on a relational database. The data collected in these systems had to be included in the documentation required by the Common Criteria standard [www 2]. This required access to data from multiple sources in such a way that the data were always safe and current. From the point of view of integration, it was not necessary to notify the system about the changes but it was enough to obtain reliable data on demand. It was also important to have the system integration flexible – to easily add a new system or replace the initially proposed one. The D2RQ tool fulfils all of these assumptions. In addition, it was necessary to make some modifications to the tool to ensure that it fits the needs of the built environment.

Therefore, the following features were used:

- the possibility to bind multiple data sources – ability to use multiple systems that collect their data in relational databases;
- the possibility to bind the same data from different sources – some data (data class) occur simultaneously on multiple systems. For example, information about the used parts of the implementation of the product can be stored both in text files placed somewhere in SVN as well as in another system to manage this type of data;
- the possibility to supplement the information about some data from different sources – some are complementary. For example, information about bug reports can be fully completed by the user who made or repaired the problem (full user information is stored in the system for personnel records and report bugs are stored in the bug tracking system). Alternatively, information about test cases (stored in the test management system) may be supplemented by information on the module to which these tests relate (information about modules are stored in the UML tool);
- building the ontology – one common schema, free of technical data (tables with relationships, keys, chains, etc.). This allows to present the schema to the user in a uniform manner for the entire system, and more comprehensibly. It is also possible to navigate between the data with the use of the properties of objects that somehow hide the technical aspects of the relationship and database joins.

The works that were carried out during the project touched many aspects but the article describes only those aspects which relate directly to the issues of data integration. In the case of the performed integration, the works were associ-

ated with the construction process of mapping and ontology. For this purpose, the following tools were used:

– Java platform with Eclipse and Maven – in the course of the project, it was necessary to implement certain parts of the system. The Java platform and the Eclipse and Maven tools were chosen;
– Semantic Works ontology editor – it was chosen (commercial license) due to the simplicity and clarity of use. Another proposal was to use the Protege tool but it seemed to be too complicated (too much functionality);
– Tomcat application server – D2RQ was embedded and run in Tomcat.

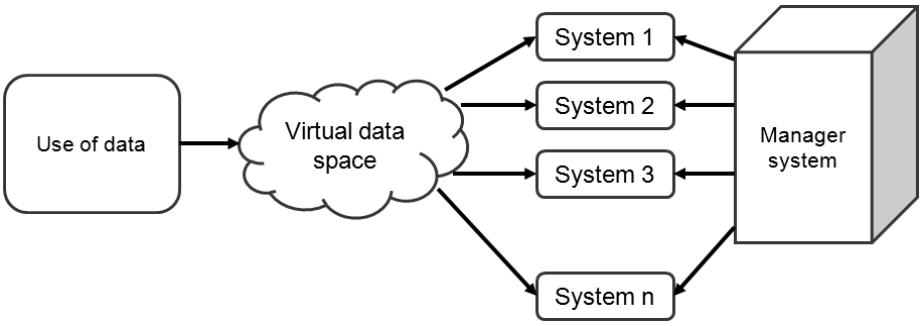

Fig. 1. The architecture of CCMODE system

The architecture (from the integration point of view) of the CCMODE system is presented on Fig. 1. There are several systems: System 1 to System n which have to be integrated. One part of integration is the manager system. The manager system has to manage projects in whole CCMODE system and in individual component systems. It is done by execution of different operations which add/update project, add user, grant privileges etc. The second part of integration is virtual data space. This part is based on D2RQ. It treats all systems as data sources and make them available as one coherent data space. This virtual data space can be used by different clients.

3. Experiences from use and modification proposals

Using D2RQ gave some insights and experiences as well as allowed to come up with some proposals. In addition to relational databases it was necessary (CCMODE project requirement) to integrate data from the SVN system. SVN does not store the data in a relational database. It was necessary, therefore,

to simulate an operation in a relational database, so far as this was required by the D2RQ system and resulted from the requirements. For this purpose, the JDBC driver has been implemented in which simulation logic was saved. Of course, due to the fact that SVN has a different data structure (tree structure) and for performance reasons, the simulation applies only to a limited set of data (limited set of tables, lack of relationships and joins between tables, appropriate response to specific requests from D2RQ).

The D2RQ developers made available an additional set of support tools such as tools for generating the mapping. This generator reads the database structure and generates the mapping information according to the meta data read from the database (table structures, constraints, reference, etc.). The generated mapping also creates its own ontology which matches the structure of the database [Levy, 2000; Halevy, Ordille, Rajaraman, 2006]. From our point of view, the direction of the napping implementation seemed wrong. Due to multiple data sources with different schemas, it was necessary to first build an ontology that described the data in a manner consistent with the functional requirements and then generate the appropriate mapping for each database separately, in order to adapt a different database to the ontology. This direction provided consistency and uniformity of data structures despite the various data sources. For this purpose, a tool to generate a mapping example was implemented, however, the tool generates the mapping which is not based on the relational database schema but on the ontology and its properties. The reason, why we developed the additional tool, was the fact that in the initial phase of the project a data analysis had been carried out. The summary of the analysis was the domain ontology which describes data in the project. This allowed to create simultaneously software components in the CCMODE project, despite of the lack of actual data access. Particular software modules had semantic descriptions of the data under development and necessary test data for the production of software . The actual data from the operating system components were gradually supplied to the common data space as far as the mapping corresponding to the predetermined ontology was implemented. The ontology helped to stabilize the data layer at a relatively early stage of the project life. Possible problems related to the provision of data availability and requiring some modifications in the data layer do not affect the source code in which the data are used. The ontology fulfils the role of a layer which separates the data from the data management layer.

In the case of building an ontology the authors proposed to divide the described areas to multiply namespaces for the ontology which represents data in databases. The division was supposed to help to organize work on the ontology

and the mapping between the ontology and the database. Each sub domain was named and the name transformed into the name space. Unfortunately, it was a wrong direction. The division of data (for clarity) should be made at the level of division between different RDF files describing different aspects of the domain and not by multiplying the namespaces. The multiplicity of namespaces turned out to be cumbersome in this case, and difficult to maintain and manage.

The D2RQ tool was also extended with the ability to support multiple independent data domains under a single server instance. Each domain had its own mapping file stored in the database. The mechanism that generates the mapping based on the current settings of the domain was also developed. The mapping generating process was based on suitable matching of the mapping file fragments. During the matching operation, the relevant data were added, some of the mapping templates were used several times with different parameters. To implement this functionality, the XSLT engine was used. Some portions of the mapping were saved as files with appropriate *xsl* tags and further processed by the XSLT engine involving data stored in the form of XML. The format of the mapping – n3 files – well fitted the .xsl format.

Working with D2RQ suggested various ideas for its development. First of all, the tool lacks the ability to control the mapping data based on the proposed ontology. Even the easiest way to verify the correctness of the entered properties and classes (e.g., in terms of possible spelling errors, lack of description of the properties in the ontology, etc.) would be useful. Inheritance relationships stored in the ontology (reasoning) are not used, probably due to poor performance. The D2RQ version used during the project did not serve binary files (data type CLOB, BLOB in the database). This function was implemented by the project team. In subsequent versions of D2RQ this possibility was also introduced.

Conclusions

To sum up, it seems that the proposed method of integration and the D2RQ tool itself may be an interesting alternative and a slightly different way to the methods for the integration of information systems. Although, the main purpose of this tool is to publish data on the Internet for the needs of new ideas related to Web 3.0, our application brings many extra benefits. It is would be useful to apply it in certain cases – in situations where it is necessary to access data in a synchronous manner, as a kind of reference to the data, or while using the data

on demand. The solution based on D2RQ as an integrator is not suitable in situations where it is necessary to notify about changes in integrated systems.

A limitation to consider while designing new solutions is one-way flow of data from the source system to the common space. In many cases it may be an advantage – it ensures that integration does not change the data, and therefore does not affect the consistency and integrity of data in the source system. However, solutions that require cross-updating of the data, will not work with D2RQ. In the case of an RDF graph which is a format to share the data, a primary language to access data is SPARQL.

From the point of view of the CCMODE project team, it would be useful to have a JDBC/ODBC interface for D2RQ. Though it is not the main purpose of this product, it would enrich D2RQ functionality and usability. On the other hand, mapping expressiveness seemed to be sufficient (as far as it was necessary). Although in most cases standard language constructs for mapping were sufficient, in a few cases it was necessary to create additional views in the database schema, but this was due to the specific nature of the database. Another way was to use our own JDBC driver or to extend mapping to serve binary files. Generally, D2RQ is a good tool and can be used in the field of enterprise systems integration.

Ongoing work from design to implementation of the system allowed the authors to read through the issue of data integration using semantic techniques. Among customers there is demand for even more advanced systems for data integration. Potential customers are becoming more aware that systems from different vendors often constitute complex heterogeneous systems and without advanced integration tools, aimed to describe the meaning of the data rather than the transfer data, it would not be possible to describe the complex environment in which these systems run.

The decision to use D2RQ for the integration of data sets has been taken by the team who were preparing the solution architecture in the CCMODE project. It was the practical use of previous research related to semantic databases. It is worth mentioning that 10 years ago the number of publications in this field and the number of developed tools pointed that semantics was seen as a very promising alternative to relational databases. However, over the last 10 years the interest in semantics has been declined. There are no more practical applications and the development of many tools has been abandoned.

It can be concluded that semantic technologies in the field of practical solutions did not come true. In the field of NoSQL databases (general), other than semantic solutions are developed [Han et al., 2011]. These other approaches to

NoSQL are currently used in practice. This is evidenced by solutions supported by commercial providers [Leavitt, 2010, Hossain, Moniruzzaman, 2013].

From today's point of view, the decision which was made in the CCMODE project seems to be misguided because of the lack of support from both academic and commercial solutions. There are no further interesting studies in the field of semantic data and the attached references reflect the material on which the CCMODE project team has relied.

References

Auer S. Dietzold S., Lehmann J., Hellmann S., Aumueller D. (2009), *Triplify: Light--weight Linked Data Publication from Relational Databases* [in:] Proceedings of the 18th International Conference on World Wide Web, ACM, 2009, s. 621-630.

Bizer Ch., Seaborne A. (2004), *D2RQ-treating non-RDF Databases as Virtual RDF Graphs* [in:] Proceedings of the 3rd International Semantic Web Conference (ISWC2004), s. 26.

Bussler Ch. (2003), *B2B Integration: Concepts and Architecture*, Springer Verlag, Berlin, Heidelberg.

Dittrich K.R., Ziegler P. (2004), *Three Decades of Data Integration-all Problems Solved* [in:] 18th IFIP World Computer Congress (WCC 2004), Toulouse, France, s. 3-12.

Erling O., Mikhailov I. (2009), *RDF Support in the Virtuoso DBMS* [in:] T. Pellegrini, S. Auer, K. Tochtermann, S. Schaffert (eds.), *Knowledge-Networked Media*, Springer, Berlin–Heidelberg, s. 7-24.

Halevy A., Ordille J., Rajaraman A. (2006), *Data Integration: The Teenage Years* [in:] Proceedings of the 32nd International Conference on Very Large Data Bases, VLDB Endowment, s. 9-16.

Han J., Haihong E., Le G., Du J. (2011), *Survey on NoSQL Database* [in:] Pervasive Computing and Applications (ICPCA), 6th International Conference, IEEE, s. 363-366.

Hossain S.A., Moniruzzaman A.B.M. (2013), *Nosql Database: New Era of Databases for Big Data Analytics-classification, Characteristics and Comparison*, arXiv, preprint arXiv:1307.0191.

Hull R., Zhou G. (1996), *A Framework for Supporting Data Integration Using the Materialized and Virtual Approaches*, ACM, New York.

Leavitt N. (2010), *Will NoSQL Databases Live up to Their Promise?* "Computer", No. 43(2), s. 12-14.

Lenzerini M. (2002), *Data Integration: A Theoretical Perspective* [in:] Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, ACM, s. 233-246.

Levy A.Y. (2000), *Logic-based Techniques in Data Integration* [in:] J. Minker (ed.), *Logic-based Artificial Intelligence*, Springer US, New York, s. 575-595.

Magnani M., Montesi D. (2007), *Uncertainty in Data Integration: Current Approaches and Open Problems* [in:] VLDB workshop on Management of Uncertain Data, s. 18-32.

Press R. (2008), *Ontology and Database Mapping: A Survey of Current Implementations and Future Directions*, "Journal of Web Engineering", No. 7(1), s. 001-024.

[www 1] http://www.w3.org/TR/r2rml/ (access: 06.05.2013).

[www 2] http://www.commoncriteriaportal.org/ (access: 06.05.2013).

WYKORZYSTANIE D2RQ W INTEGRACJI NARZĘDZI ROZWOJOWYCH

Streszczenie: Artykuł opisuje doświadczenia związane z realizacją semantycznej integracji systemów informatycznych. Prezentuje wykorzystanie integracji w rozumieniu wirtualizacji zbiorów danych w spójny graf semantyczny. Integracja została zrealizowana zgodnie z wypracowaną wcześniej metodyką ATOM, z wykorzystaniem narzędzia D2RQ. W artykule zostało przedstawione omawiane narzędzie jako reprezentant klasy narzędzi służących do integracji wielu źródeł danych w jedną spójną przestrzeń danych. Ponadto zostały opisane doświadczenia z wykorzystania poszczególnych możliwości D2RQ pod kątem integracji źródeł danych oraz różne sposoby podejścia do zagadnień związanych z tym tematem. Autorzy przedstawili również swoje modyfikacje narzędzia oraz propozycje jego udoskonalenia. Przeprowadzone badania i zebrane doświadczenia były bezpośrednio związane z realizacją projektu CCMODE (Common Criteria compliant Modular Open IT security Development Environment).

Słowa kluczowe: semantyczna integracja, narzędzia, semantyczna sieć, D2RQ.