[SBStU07] *Stefan Schulte, Rainer Berbner, Ralf Steinmetz, Mathias Uslar;* **Implementing and evaluating the Common Information Model in a relational and RDF-based Database;** Information Technologies in Environmental Engineering, ITEE 2007 - Third International ICSC Symposium, Oldenburg, März 2007, S. 109-118.

Implementing and evaluating the Common Information Model in a relational and RDF-based Database

Stefan Schulte, Rainer Berbner, Ralf Steinmetz[1]

Multimedia Communications Lab (KOM), TU Darmstadt, Germany

Mathias Uslar[2]

OFFIS, Business Information Management, Oldenburg, Germany

Abstract: During the last decade, the Common Information Model (CIM) has evolved to an extensive ontology for the domain of energy markets. As the CIM does only offer an UML model for the implementation of its objects, an ER model or relational database schema has not been released. Therefore, it is necessary to create a CIM based database schema in order to persist CIM data in a relational database. This schema could either be constructed based on the former mentioned UML model as a relational database schema or based on an already existing RDF/XML serialization of CIM as an RDF database. This paper evaluates these two implementations of the CIM.

Keywords: Common Information Model, CIM/XML, RDF, power system data exchange, energy management systems

1 Introduction

Not only the electrical power outages in the US and Switzerland in 2003 have shown the need for fast and automatic data exchange between companies which operate in power markets. Furthermore, factors like the increasing deregulation of such markets and the accompanied distinction be-

[1] {schulte|berbner|steinmetz}@kom.tu-darmstadt.de
[2] uslar@offis.de

tween network operators and companies accessing power networks resulted in a need for integrated energy management systems (EMS). Such systems are based on a common management and a capable data exchange format (Becker et al. 2000).

In this paper, different approaches for persisting CIM/XML data in a relational database are proposed and evaluated. In Section 2 we provide basic information about CIM and its serializations. In Section 3 we will propose a mapping from CIM/XML to standard SQL by using the Java API *HP Jena* based on a translation from the CIM UML model to a relational database schema. A performance evaluation of this approach is presented in Section 4. Finally, Section 5 presents a conclusion of the findings of this paper, some related work and recommendations for future work.

2 The Common Information Model and its serializations

As a data exchange format should also be suitable for the data interchange with other companies, it is beneficial to use an approved standard instead of proprietary formats. Nowadays, it seems as the Common Information Model (CIM) will be the most common data exchange format for energy markets in the future. Core elements of CIM have been adopted in International Standard IEC 61970-301, other elements are currently drafts and will be standardized as IEC 61970-302 and -303. CIM is an extensive ontology for the domain of electric utilities and is available in different formats, e.g., in Rational Rose UML, XMI and RDF Schema. The model has recently also been released in Web Ontology Language OWL (Britton and deVos, 2005). More information on the CIM can be found in (Uslar et al., 2005) and (Uslar, 2006).

Deploying CIM leads to a lot of fundamental decisions, for instance if it is advantageous to sustain the currently used database or to implement a new database schema. As CIM does not offer any official database model and data exchange is based on the RDF representation of CIM (called *CIM/XML*), it seems consequential to establish an RDF-based database instead of a relational database schema. However, it is quite unlikely that it is possible to introduce a completely new data management in a company; instead, it is necessary to map CIM/XML data from and to the existing database.

3 Mapping RDF to SQL and vice versa

As aforementioned, currently there is no officially released implementation of CIM in a relational database schema. Hence, it is necessary to create a database schema and map CIM/XML data from RDF to SQL in order to write the information into a database. In a standard scenario, one company A would extract data from its relational database, convert the data to CIM/XML and send it to another company B. B will reconvert the data to SQL and write it into its own database.

Converting data from RDF to SQL and backwards is the main challenge of the information exchange between the two parties. Unfortunately, this mapping between the two data types cannot be done automatically as database schema and RDF schema cannot be matched without multiple adjustments. In the following two subsections the mapping from RDF to SQL and vice versa will be briefly described.

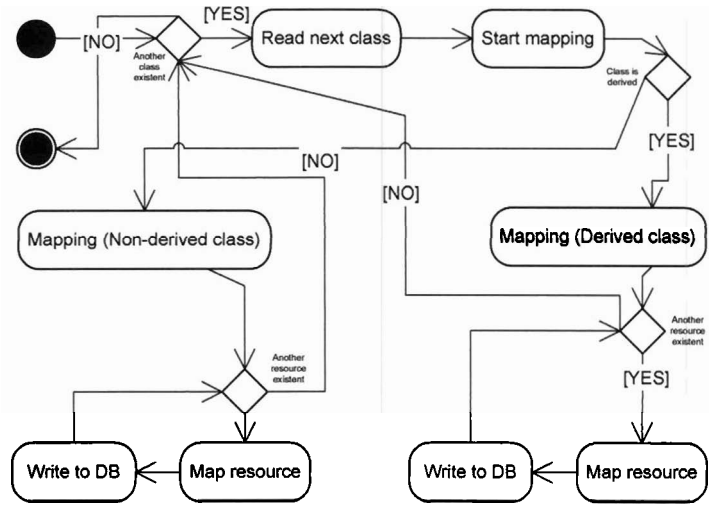

Fig. 1. Mapping data from CIM/XML to a relational database

3.1 Mapping data from CIM/XML to a relational database

The database schema is based on the approach suggested by (Podmore and Robinson 2000) – in real world examples it is highly probable that the data has to be converted from the original relational database schema to a more appropriate form first or that the mapping algorithms have to be extended.

The mappings as well as other functionalities (e.g. methods for performance measurement) have been implemented in the demonstrator program *JACIM*. JACIM uses Hewlett-Packard's Open Source Java-API *Jena Semantic Web Framework* (HP Jena) to read, parse and write RDF Data and is able to map CIM/XML data to SQL statements, write these statements into a relational database and backwards.

In the database, every CIM class is implemented using a separate table; attributes and relationships are represented by the rows of this table.

The mapping from CIM/XML to SQL and insertion of the data into the relational database is based on the activity diagram outlined in Figure 1.

First, the test data is read and parsed by Jena's RDF/XML parser *ARP*. Afterwards, all RDF data is available in form of one RDF model. This model is iterated and every resource (i.e., one RDF subject) is handled on its own. Resources include several RDF statements, i.e. objects and the ID of this particular subject. After iterating the statements, all RDF data is available in form of one (if the particular CIM class is not derived from another CIM class) or more (in every other case) SQL statement(s). These SQL statements are executed and written to the database.

As primary keys of derived classes are also foreign keys which point to the primary keys of the derived class' superclasses, statements regarding superclasses have to be carried out first in order to avoid insertion errors. Relationships to other classes are realized as foreign keys, too. This implies that classes which are the "destination" of such a relationship have to be inserted in the database prior to the actual class.

Primarily, the mapping of derived and non-derived classes differs in the point that in the case of derived classes, all superclasses have to be known before the mapping starts. Both activities "Map resource" (cp. Figure 1) include a case differentiation. These case differentiations refer to the statements of a resource:

- In the first differentiation, statements which are primary keys or part of a n..n-relationship are separated and handled separately from other statements.
- These other statements are distinguished in the second case differentiation: Statements representing relationships are handled different from other attributes.

3.2 Mapping data from a relational database to CIM/XML

The mapping from a relation database to CIM/XML is outlined in Figure 2. Data from the database is selected with a *Natural Join* which joins the actual class with all its superclasses. The result of this join is a Java Re-

sultSet which is processed result by result. Analog to the case differentiation presented in Section 3.1, a case differentiation has to be done when mapping data from the relational database to CIM/XML. However, in this case it is not complex to such an extent; basically it is necessary to verify if a row from the ResultSet is a primary key, foreign key or normal attribute.

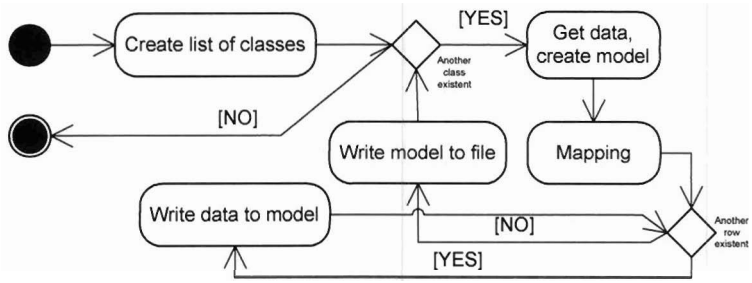

Fig. 2. Mapping data from SQL to CIM/XML

4 Evaluation

Based on the mapping introduced in Section 3, the implementation of CIM/XML in a relational and RDF-based database has been evaluated. The implementation using a relational database is named *SQL-approach*; its equivalent for a RDF-based database is called *RDF-approach*.

Test data is provided by the CIM Validation Tool (CIM VT, Areva60-2006-03-17.rdf) and includes 30 CIM classes and 23352 RDF statements in 3036 resources.

Both approaches are evaluated with the same test data. However, it is necessary to prepare the data for the SQL-approach. In the original test file, the data is listed according to the name of the actual CIM class. This presents no problem to RDF tools, as relationships between resources do not have to be existent at the moment a resource is parsed or written. Anyhow, if writing SQL statements to a database, all foreign keys have to be existent or insertion errors (cp. Section 3.1) will emerge. Therefore, the sequence of resources has to be changed so that every foreign key points to an already existing resource.

All implementation and performance tests are carried out on a modern notebook with 1.5 GHz and 768 megabytes main memory. As the objective of this paper is a comparison between RDF- and SQL-approach, there is no need for a separated database server etc. Microsoft Windows XP Home SP2 is used as operating system; Microsoft SQL Server 2000 De-

veloper Edition SP4 is employed as database. Java SDK is used in version 1.50_02, HP Jena in version 2.4 and jTDS (Open Source JDBC driver) in version 1.2. Furthermore, version 10 revision 7 of CIM is applied in our test environment.

The evaluation consists of two parts: In Section 4.1, the architecture and implementation efforts of both approaches are presented; Section 4.2 compares the performance of both implementations. Section 4.3 summarizes the finding of the evaluation and gives a recommendation for future handling of CIM integration.

4.1 Architecture and implementation efforts

Using a relational database to store CIM data implies that data has to be mapped from CIM/XML to the relational database schema and backwards. In the case of using a RDF-based database, such a mapping is obsolete. Therefore, the implementation efforts of the SQL-approach are significantly higher than those of the RDF-approach.

A higher implementation effort leads to several disadvantages of the SQL-approach in comparison to the RDF-approach. First of all, a complex implementation increases the probability of errors, especially if special cases have to be handled.

Secondly, the used mapping algorithms have to be extended if it is intended to store CIM classes which have not already been handled in our demonstrator program. The used test data contains only a small part of CIM classes but is already handling a lot of special cases. As a result, the SQL-approach is less flexible and maintainable. The flexibility is also reduced due to the fact that all foreign keys have to be defined first in order to avoid insertion errors.

Finally, the SQL-approach needs a defined database schema while the RDF-approach creates all necessary database tables by itself.

4.2 Performance

In this subsection the results of our performance tests are presented. The performance tests are carried out on three different test data sets which are all parts of the test data provided by CIM VT. The first data set contains 11 resources and 22 statements, the second 472 resources and 3470 statements. The last data set contains all data from CIM VT (3036 resources, 23352 statements). Every performance test is carried out three times in a new database in order to minimize measurement errors.

As it can be seen in Figure 3, the RDF-approach is slower than the SQL-approach when writing data from the CIM/XML-file to the database.

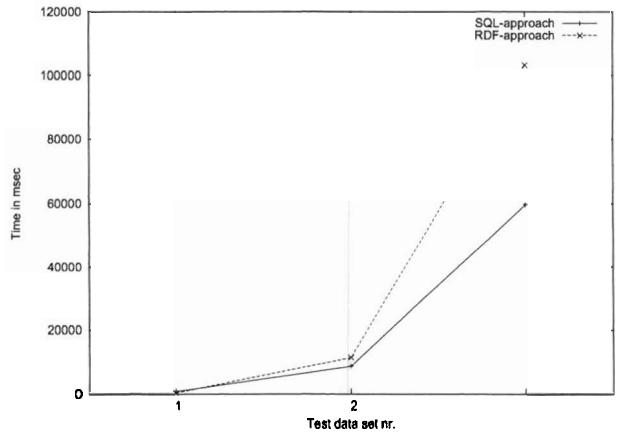

Fig. 3. Writing CIM data to database

The difference in the measured values increases disproportionately: The measured time for the SQL-approach grows roughly commensurate with the number of statements written to the database while the written statements per time unit decreases for the RDF-approach if a higher number of statements is examined.

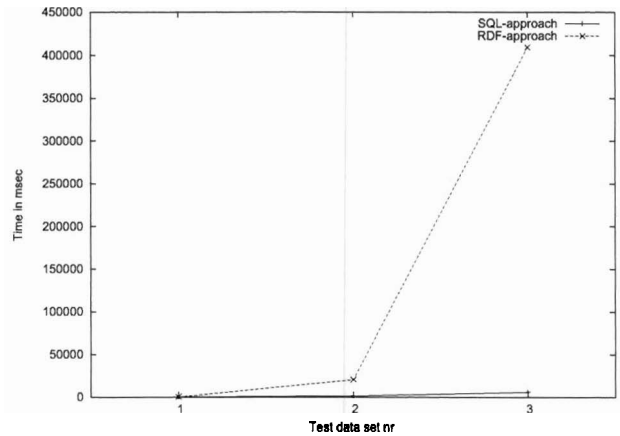

Fig. 4. Writing CIM data from database to file

As it can be seen in Figure 4, the results for writing from the database to a CIM/XML-file are quite similar, but the gap between the measured values is even bigger.

Again, the difference in the measured values increases disproportionately for the RDF-approach. In contrast, the measured values for writing the CIM/XML-file increase only slightly if using the SQL-approach.

Examining the present performance test results leads to some questions as it was not likely that the SQL-approach would come off better than the RDF-approach. The difference between the performances is especially large if writing from the particular database to the CIM/XML-file.

As both approaches invoke Jena's RDF parser and writer in the same way, the difference in the performance numbers has to result from other reasons.

Apparently, the database connection of Jena is much slower than the standard database connection provided by Java which is used in the SQL-approach. Furthermore, standard SQL statements seem to perform better than the statements Jena creates to write data to the RDF-based database.

All in all, the performance tests lead to an unexpected but unambiguous result: The SQL-approach is superior to the RDF-approach in terms of performance.

4.3 Conclusion

The evaluation leads to mixed results. While the RDF-approach is preferable due to its lower implementation efforts and better liability to errors, extensibility, maintainability and flexibility, the SQL-approach features a much better performance.

To assess which approach is easier to apply in a real-world scenario, more criteria should be taken into account: E.g., almost every company operates a SQL compatible database. Therefore, introducing a completely new technology like an RDF-based database is nearly impossible particularly with regard to other involved information systems. On the other hand, CIM/XML and similar formats gain in importance and need to be integrated in the information systems of companies involved in power markets.

The mapping between RDF-based CIM/XML and the SQL-based database has to take place somewhere: Either between the other systems and an RDF-based database or between CIM/XML and the relational database. However, from a business perspective the integration of the SQL-approach is easier as only the information system for im- and exporting CIM/XML

data has to be endorsed with a mapping to the existent information systems.

It should be noted, that the scenario in this paper is quite perfect as the relational database is designed in accordance with CIM/XML (cp. Section 3.1).

All things considered, the SQL-approach is preferable at the moment.

5 Summary and future work

In this paper we presented the CIM as a tool to exchange data between two companies. Even though the RDF-approach has got lower implementation efforts and better liability to errors, extensibility, maintainability and flexibility, the SQL-approach is recommended due to its better performance and easier integration into existing information systems.

To our knowledge, there are no published comparisons between the both approaches to integrate CIM as it was carried out in this paper. Nevertheless, there is some existent related work we want to mention. First of all, *D2RQ* (http://sites.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/) offers a way to treat non-RDF databases as virtual RDF graphs and could provide another way to integrate CIM and databases. Furthermore, (Theoharis et al., 2005) and (Grosse, 2003) have investigated the use of different RDF-based databases and tools.

Parallel to the research efforts presented in this paper, CIM was introduced in Web Ontology Language (OWL); as the name implies, OWL is better suited to represent ontologies than RDFS. OWL will probably be the preferred CIM format in the future and there will be only restricted support of CIM/XML. Therefore, we will examine the use of CIM/OWL in the future.

Furthermore, we will enhance the mapping from a relational database to CIM/XML data. At the moment, this mapping works but has got some shortcomings, e.g., with case sensitivity. Although the mapping provides semantically identical data (compared to the original test data), HP Jena uses a different RDF-syntax than the parser used by CIM VT. Accordingly, the resulting data is syntactically different.

As our mapping algorithms are too inflexible and have to be extended if mapping CIM classes which are not part of the test data from CIM VT, new algorithms will supersede the used ones.

Another aspect of our future work is the examination of alternative RDF-parsers and -writers. As it was outlined in Section 4.3, Jena's performance could be better when writing to the database and parsing data

from the database to RDF. Alternative frameworks like Sesame (http://www.openrdf.org/) could be used to improve the performance of the RDF-approach.

Last but not least, (Britton and deVos, 2005) propose a scenario where CIM could be used as the foundation for a Service-oriented Architecture (SOA). In the future, we will use CIM as a way to add a semantic base to the SOA reference architecture WSQoSX introduced by (Berbner et al., 2005).

References

Becker, D. et al. (2000) Standards-Based Approach integrates Utility Applications. IEEE Computer Applications in Power, pp 13-20

Berbner, R. et al. (2005) An approach for the Management of Service-oriented Architecture (SoA)-based Application Systems. In: Proceedings of the Workshop Enterprise Modelling and Information Systems Architectures (EMISA 2005), pp 208-221

Britton, J. P., deVos, A. N. (2005) CIM-Based Standards and CIM Evolution. IEEE Transactions on Power Systems. Volume 20, pp 758- 764

Grosse, J. (2003) XML Clearinghouse Report 6: Speicherverfahren und Werkzeuge für RDF/S

Podmore, R., Robinson, M. (2000) Introduction to CIM. PowerData Corporation. Incremental Systems Corporation

Theoharis, Y. et al. (2005) Benchmarking Database Representations of RDF/S Stores. In: 4th International Semantic Web Conference (ISWC 2005), pp 685-701

Uslar, M. (2006) The Common Information model for utilities: An Introduction and Outlook on Future Applications. In: Rainer Eckstein, Robert Tolksdorf (Hrsg.): Proceedings der XML-Tage 2006 in Berlin, XML-clearinghouse.de, pp 135-148

Uslar, M. et al. (2005) Interaction of EMS related Systems by Using the CIM Standard. In: Walter Leal Filho, Jorge Marx Gómez, Claus Rautenstrauch (Hrsg.): ITEE 2005: Second International ICSC Symposium on Information Technologies in Environmental Engineering: Proceedings, Otto-von-Guericke Universität Magdeburg, Shaker Verlag, Aachen, pp 596-610,