SYSTEM FOR STORAGE AND MANAGEMENT OF EEG/ERP EXPERIMENTS

Generation of Ontology

Roman Mouček and Petr Ježek
Department of ComputerScience and Engineering, University of West Bohemia
Univerzitní 8, 306 14 Pilsen, Czech Republic

Keywords: Relational Database, Object Oriented Programming, Semantic Web, Ontology, OWL, RDF, Semantic Web Tools, Semantic Web Frameworks, Semantic Mapping, Electroencephalography, EEG, Event-related Potentials, ERP, Hibernate, Spring MVC.

Abstract: This paper shortly describes the system, which provides the possibility to store and manage data and metadata from EEG/ERP experiments. The system is planned to be registered as a source of neuroscience data and metadata. It is one of the reasons we need to provide the system ontology. The scientific papers often describe the domain by using a semantic web language and consider this kind of domain modelling as a crucial point of software solution. However, real software applications use up the underlying data structures such as relational database and object classes. That is why the fundamental differences in semantics between common data structures (relational database, object oriented code) were summarized. The existing tools in semantic web domain were studied and partially tested. The first transformations from the system relational database and object oriented code were performed.

1 INTRODUCTION

Our research group at Department of Computer Sciences and Engineering, University of West Bohemia in cooperation with other partner institutions (Czech Technical University in Prague, University Hospital in Pilsen, Škoda Auto...) specializes in the research of attention, especially attention of drivers and seriously injured people. We widely use the methods of electroencephalography (EEG) and methods of event-related potentials (ERP).

EEG and ERP experiments take usually long time and produce a lot of data. Because there is no usable software tool for long-term storage and management of data obtained during these experiments, we have developed our own software tool for EEG/ERP data storage and maintenance (called simply the system in the following text) (Ježek, 2010).

Registration of the system as a recognized data source occasionally requires providing data and metadata structures in the form of ontology in accordance with ideas of semantic web ("SWB", 2001). Representation of data and metadata using ontologies is also supported by scientific effort to integrate data from various data sources and to develop autonomous agents reading and transferring data into an ontology form.

This paper briefly introduces the system for storage and management of EEG/ERP experiments, describes its architecture and used technologies, and explains the basic approaches for building ontologies. Then some known approaches (and their advantages and drawbacks) to a mapping of knowledge from common data structures (relational database, object oriented code) to semantic web languages are summarized. Existing mapping tools and occasionally our experience with them are described. Finally, the preliminary solution for transformation of our system data and metadata to semantic web ontology is looked for and discussed.

The final aim of this paper is to open a wider discussion concerning both the theoretical background and practical mapping possibilities from common structures especially if modern open source frameworks and technologies are used and thereby some kind of inner semantic mapping is already performed.

Mouček R. and Ježek P. (2010).
SYSTEM FOR STORAGE AND MANAGEMENT OF EEG/ERP EXPERIMENTS – Generation of Ontology.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Databases and Information Systems Integration*, pages 415-420
DOI: 10.5220/0003019904150420

2 SYSTEM FOR STORAGE AND MANAGEMENT OF EEG/ERP EXPERIMENTS

System for storage and management of EEG/ERP experiments enables clinicians and various community researchers to store, update and download data and metadata from EEG/ERP experiments. The system is developed as a standalone product; the database access is available through a web interface (Figure 1).

My data

Date	ID	Title	
01.05.2009, 12:00	24	xml soubor	Detail
04.04.2010, 09:45	12	ahoj	Detail

EEGbase - database for data gained in encephalography research.
Copyright © The University of West Bohemia 2009-2009

Figure 1: User Interface preview.

The system essentially offers the following set of features (the set of accessible features depends on a specific user role):

- User authentication
- Storage, update, and download of EEG/ERP data and metadata
- Storage, update and download of EEG/ERP experimental design (experimental scenarios)
- Storage, update and download of data related to testing subjects

The system is based on tree layer architecture (MVC pattern) consisting of persistent layer (relational database), application layer (object oriented code, object relational mapping from persistence layer) and presentation layer (JSP). The persistence layer uses Hibernate framework; Oracle 11g database server is used to ensure the processing of large data files. Application and presentation layers are designed and implemented using Spring technology. This framework supports MVC architecture, Dependency injection and Aspect Oriented Programming. There were no significant difficulties with integration of both frameworks, Hibernate and Spring MVC. Spring Security framework is used to ensure management of authentication and user roles.

Since the system is thought to be finally open to the whole EEG/ERP community there is necessary to protect EEG/ERP data and metadata, and especially personal data of testing subjects stored in the database from an unauthorized access. Then a restricted user policy is applied and user roles are introduced.

The complete overview of the system features and user roles (use case diagram) is available in (Pergler 2009).

Concerning the architectural layers there is a question which layer is more feasible for mapping of its structure into ontology. Currently we have studied two possibilities:

- Mapping from the persistence layer (relational database)
- Mapping from the application layer (object oriented code)

The mapping from the application layer to an ontology includes the precedent object relational mapping provided by Hibernate framework. The next section discusses the differences between the semantics of ontologies and common data structures.

3 SEMANTICS OF ONTOLOGIES AND COMMON DATA STRUCTURES

Although a definition of ontology is still under debate we will consider the ontology definition given by Gruber (Gruber, 1993) and Stabb and Studer (Stabb, 2004) who consider ontology as a formal explicit specification of a conceptualization for a domain of interest.

3.1 OWL

The standard for expressing semantic web ontologies is nowadays W3C OWL 2 Ontology Web Language (OWL) (OWL, 2009), which is based on description logic. They are various syntaxes available for OWL 2 and they serve various purposes. The RDF/XML syntax is the only syntax that is mandatory to be supported by all OWL 2 tools.

Knowledge expressed in OWL documents (ontologies) can be reasoned with computer programs either to verify the knowledge consistency or to make implicit knowledge explicit (OWL Primer, 2009). The essential part of OWL ontology is vocabulary (a set of central terms) with interrelation information (the meaning of a term is characterized by its interrelation to other terms). Then there is important to emphasize that OWL is neither a database framework (although there is an analogy e.g. between assertional information and database content) or a programming language (OWL is a declarative language; e.g. algorithmic realization of inferences is not a part of OWL document).

OWL 2 also does not provide any means to prescribe the document syntactic structure. It means that there is impossible to enforce the presence of any information in a document.

Concerning the semantics of OWL document the correct answer to any question is predetermined by the formal semantics in two possible versions: the Direct Semantics and RDF-Based Semantics.

OWL of course cannot reflect and represent all aspects of human knowledge. On the other hand, an important OWL feature is that it captures the possibility to draw consequences from the knowledge. The automatic computation of these consequences is made by OWL reasoners. This is also the strength of OWL 2 because these OWL tools can discover information that can be hardly found by people. However, this also means that it is difficult to predict the future effect of various constructs and their combinations.

The basic OWL notions include axioms (the statements expressed by OWL ontology), entities (elements referring to real-world objects – individuals, classes and properties) and expressions (combinations of entities).

There are fundamental differences in richness of semantics between OWL (Description Logic based system) and relational database or object oriented systems. On the other hand, there are several approaches how to bridge at least some of these semantic gaps. These issues are discussed in the following sections.

3.2 OWL and Relational Database

What are the differences and similarities comparing relational databases and OWL? How to bridge the gap between ontologies and relational databases?

Except an analogy concerning assertional information (the previous section) an analogy between ontology terminological information and a database schema can be found. On the other hand, they are important differences in the underlying semantics. If some information is not present in a database, it is considered to be false (closed-world assumption). By contrast, if some information is not present in an OWL document, it may be missing and possibly true (open-world assumption).

LePendu (2007) aligns the expressiveness of existing ontology languages with the capabilities of RDBMS. His work on database integration focused on generating ontology from schemas and ontology-based data integration (Dou, 2006a; Dou, 2006b). Then he continued with the work on the system taking an ontology as input and generating SQL relational database schema definition (LePendu, 2007). It is maintained that many of the first-order features which are common to ontologies are also reflected in relational databases. A simple observation is made: database schemas are like simple ontologies. Simple rules of thumb are used and there is shown that "ontology schema" for one database can be merged with another one.

However, to bridge the semantic gap between relational database and ontology means to get a broad knowledge of several fields including relational models, database normalization, knowledge representation, logic, conceptual modeling, etc. Moreover, there are not only semantic gaps between ontologies and relational database but they are still gaps between the conceptual design of a database and its implementation in relational database management system (RDBMS).

A set of general statements and specific facts within an ontology are often referred to as intensional (inferred) knowledge in contrary to extensional (explicit) knowledge. Relational databases store and retrieve extensional data well but they generally do not perform inference. Intensional knowledge reduces the amount of required extensional data storage. On the other hand, inference is a time-expending process; it means if we store more extensional data then we need less time to answer queries is necessary. From this point of view data in the systems are extensional while mappings between systems are intensional rules. The current research in data integration brings some logics in databases again (LePendu, 2007).

A methodology describing mapping of ontology basic constructs and axiomatic features to SQL database relational schema is described in (LePendu, 2007). A wide discussion is dedicated to connections between databases and logics with respect to the various theoretical foundations.

(Hu, 2008) proposes a method of building domain specific OWL ontology from relational database automatically. Three mapping rules from relational database schema to ontology class property are introduced and a method prototype is implemented.

(Lam, 2006) describes translation of two neuroscience databases into OWL and the formal merging of the resulting OWL ontologies. An existing tool D2RQ was used.

(Juric, 2008) introduces a framework for an automatic mapping of relational database content and metadata to OWL domain ontologies. The constructed ontologies are enriched with additional semantics from the WordNet lexical database. Jena tool was applied.

(Astrova, 2009) proposes an approach to automatic transformation of relational database to ontologies written in OWL with the aim to integrate data scattered across many different domains.

3.3 OWL and Object Oriented Programming

As we already mentioned OWL as a description language is not considered to be a software programming language. Moreover, object oriented programming (OOP) is not captured as the way of ontology construction. On the other hand, the ontological representation of objects in OWL is, syntactically and semantically, very similar to the description of objects, classes and instances. Then the analogy of system analysis in software engineering process with building ontologies leads to the idea of system development based on the description logic, it means formalized ontological description.

(Koide, 2005) developed an OWL processor (SWCLOS) based on Common Lisp Object System (CLOS). CLOS allows lisp programmers to develop object-oriented systems. OWL processor itself allows lisp programmers to construct domain and task ontologies in software application fields. Koide also demonstrates the possibility of the integration of OWL and OOP, and discusses semantic gaps between CLOS and OWL.

The automatic mapping from OWL ontologies into Java is described in (Kalyanpur, 2004). The authors note the fundamental differences between description logic and object oriented systems, primarily related to completeness and satisfiability. They present ways to minimize the impact of these differences and show how to map richer OWL semantics into Java. HarmonIA framework is mentioned.

4 FRAMEWORKS AND TOOLS

There is a number of frameworks and software tools, which are considered to generate OWL (RDF) output from relational database or object oriented code. Some of these frameworks and tools exist only as initial proposals or prototypes described in scientific papers, while some of them have been really implemented. The following list includes a selection from existing frameworks and software tools, which were studied for the next possible usage. If a framework or tool was widely tested at our department, our experience is added.

4.1 Jena

Jena is a well known Java framework for building Semantic Web applications. It provides a program environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine. It is developed as an open source and grown out of work with the HP Labs Semantic Web Programme. The framework includes: RDF API, OWL API, Reading and writing RDF in RDF/XML, N3 and N-Triples, in-memory and persistent storage and SPARQL query engine (Jena, n.d.). Jena is integrated within variety of tools intended for semantic web purposes.

4.2 D2RQ

D2RQ is a declarative language to describe mappings between relational database schemata and OWL/RDFS ontologies. The D2RQ Platform uses these mappings to enable applications to access a RDF-view on a non-RDF database through the Jena and Sesame APIs, as well as over the Web via the SPARQL Protocol and as Linked Data (D2RQ, 2009).

There is important for practical usage that the installation process is straightforward; all necessary libraries are in a package. We were able to generate RDF output without substantial difficulties.

4.3 Virtuoso

Virtuoso is open source software, which enables to transform SPARQL queries to SQL queries. It is possible to combine SPARQL and SQL queries over one database. The Virtuoso RDF meta schema is a built-in feature of Virtuoso's SPARQL to SQL translator. It recognizes triple patterns that refer to graphs for which an alternate representation is declared and translates these into SQL accordingly (VIRTUOSO, 2009).

4.4 SquirrelRDF

SquirrelRDF is a tool which allows non-RDF data stores (not explicitly RDF) to be queried using SPARQL. It includes relational databases and LDAP servers. It provides a Query Engine, a command line tool, and a servlet for SPARQL http access. The result is the information looking like RDF. However, it makes no attempt, for example, to reveal implicit relations between objects (suggested by foreign keys), or normalise denormalised data (SqRDF, n.d.).

4.5 METAMorphoses

The data transformation processor METAmorphoses transforms data from a relational database into RDF documents according to a mapping. To achieve this goal it processes schema mapping and template documents written in XML languages. The processor employs an algorithm based on author´s data transformation model, which is maintained to have a higher performance than similar solutions in the field. The tool is designed to hide the complexity of the semantic web technologies into the schema mapping layer, while exposing the simple template layer to the programmer (Švihla, 2007).

This software tool works well with predefined data source but any change means a lot of manual work especially during templates construction. Data processor and supplemental software tools failed several times during operations without informing a user about the source of difficulties.

4.6 Sommer

Sommer is a simple library for mapping Plain Old Java Objects (POJOs) to RDF graphs and back. XML/RDF template designates the form of output RDF document. The XML/RDF template is extended about information from input JavaBeans (SOMMER, n.d.).

4.7 JenaBean

JenaBean uses Jena's flexible RDF/OWL API to persist Java Beans objects. Java Beans are transformed into RDF graph but the biding is driven by the Java object model rather than an OWL or RDF schema (JenaBean, n.d.). It also means that this tool does not require an input template (in contrast with Sommer) but RDF output is semantically poor and not well-formed.

4.8 Java2OWL-S

Java2OWL transforms JavaBeans directly into OWL representation by using two transformations. The first transformation is from JavaBeans into WSDL (Web Service Description Language). The input of this transformation is formed by Java class and the output is temporary WSDL file. The second transformation transforms temporary WSDL file into OWL (four OWL documents are created).

4.9 OWL API

There exist several syntaxes for representation of ontologies. OWL API is a Java API and reference implementation for creating, manipulating and serializing OWL Ontologies. It includes a number of components including RDF/XML, OWL/XML; Turtle parsers and writers, and interfaces for working with reasoners (OWLAPI, n.d.).

5 SOFTWARE TOOLS USAGE AND PRELIMINARY RESULTS

As we already mentioned in Section 1 our primary aim is to register our system as recognized data source and to provide the system ontology. It means that we need to perform only one-sided mapping from relational database (object oriented code) to OWL; then we need to use only a subset of semantic richness of RDFS and OWL.

We decided to try out two parallel approaches. The first approach includes the transformation of relational database into ontology using D2RQ tool and OWL API. The second approach includes the transformation from object oriented classes to OWL using Java2OWL-S tool.

We have encountered the following difficulties yet. EEG/ERP raw data are saved in the database as binary files. However, D2RQ tool does not work with the corresponding BLOB data type. We solved this problem by simple refactoring of D2RQ code. The original reaction (exception is thrown) was replaced by textual output using the attribute name.

The second approach only OWL output for smaller part of the Java application code was successful till now. The transformation of the whole code brings difficulties which are solved.

6 CONCLUSIONS

The presented paper shortly describes the system which provides the possibility to store and manage data and metadata from EEG/ERP experiments. Open source Java technologies were used for the development.

We plan to register our system as a source of neuroscience data and metadata within world known projects e.g. Neuroscience Information Framework (NIF, n.d.). It is one of the reasons we need to provide the system ontology. The scientific papers often describe the domain by using a semantic web language and consider this kind of domain modelling as a crucial point of software solution. However, real software applications use up the underlying data structures such as relational database and object classes. That is why the fundamental differences in semantics between common data structures (relational database, object oriented code) were summarized. The existing tools in semantic web domain were studied and partially tested. The first transformations from the system relational database and object oriented code were performed.

Our aim for the conference includes discussion concerning both the theoretical background of semantic richness using various expressing languages and practical usage of existing frameworks and software tools. These frameworks and tools are considered to be suitable for generation of ontologies over the systems that are already developed using current Java technologies.

ACKNOWLEDGEMENTS

This work was supported by Grant no. 2C06009 Cot-Sewing.

REFERENCES

Astrova, I. (2009). Rules for Mapping SQL Relational Databases to OWL Ontologies. *In 2nd International Conference on Metadata and Semantics Research (MTSR 07)* (pp. 415-424)

D2RQ (D2RQ) (2009). Retrieved November 18, 2009, from http://www4.wiwiss.fu-berlin.de/bizer/D2RQ/

Dou, D. and LePendu, P. (2006a). Ontology-based integration for relational databases. *In ACM Symposium on Apllied Computing (SAC)* (pp 461-466).

Dou, D., LePendu, P., Kim, S. and Qi, P. (2006b). Integrating databases into the semantic web through an ontology-based framework. *Proceedings of the 22nd International Conference on Data Engineering Workshops)*, (pp 54-63).

Gruber, T.D. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199-220.

Hu, Ch., Li, H., Zhang, X. and Zhao, Ch. (2008). Research and Implementation of Domain-Specific Ontology Building from Relational Database. *The Third ChinaGrid Annual Conference* (pp.289-293)

JenaBean (JenaBean), (n.d.) Retrieved January, 18, 2010, from http://code.google.com/p/jenabean/

Jena framework (Jena) (n.d.) Retrieved November 18, 2010, from http://jena.sourceforge.net

Juric, D., Banek, M., Skocir, Z. (2008). Uncovering the deep Web: Transferring relational database content and metadata to OWL ontologies. *In 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems* (pp. 456-463)

Kalyanpur, A., Pastor, D., Battle, S. and Padget, J. (2004). Automatic Mapping of OWL Ontologies into Java. *In Proceedings of the 16th Int'l Conference on Software Engineering & Knowledge Engineering (SEKE'2004)* (pp. 98-103)

Koide, S., Aasman, J. and Haflich, S. (2005), OWL vs. Object Oriented Programming. *In International Workshop on Semantic Web Enabled Software Engineering (SWESE)*.

Lam, H.Y., Marenco, L., Shepherd, G.M., Miller, P.L., Cheung. K.H. (2006). Using Web Ontology Language to Integrate Heterogeneous Databases in the Neurosciences. *In AMIA Symposium Proceedings* (pp. 464-468).

LePendu, P. (2007). *Ontology based Relational Databases*, University of Oregon.

Neuroscience Information Framework (NIF). (n. d.). Retrieved September 7, 2009, from http://www.neuinfo.org/

OWL API (OWLAPI) (n.d.). Retrieved November 18, 2009, from http://owlapi.sourceforge.net/

OWL 2 Web Ontology Language (OWL2). (2009). Retrieved February 22, 2010, from http://www.w3.org/TR/owl2-overview/

OWL 2 Web Ontology Language Primer (OWL Primer) (2009). Retrieved February 22, 2010, from http://www.w3.org/TR/2009/REC-owl2-primer-20091027/

Semantic web (SWB). (2001). Retrieved February 22, 2010, from http://www.w3.org/2001/sw/

SOMMER (SOMMER) (n.d.) Retrieved January 13, 2010, from https://sommer.dev.java.net/sommer

Staab, S. and Studer, R. (2004). *Handbook on ontologies (International Handbooks on Information Systems).* Springer.

Švihla, M. (2007). *Transforming Relational Data into Ontology Based RDF data.* Thesis. CTU, Prague.

SquirrelRDF (SqRDF) (n.d.). Retrieved January 11, 2010, from http://jena.sourceforge.net/SquirrelRDF/

Virtuoso Open-source Wiki (VIRTUOSO), (2009). Retrieved February 25, 2010, from http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSSQLRDF.

Ježek, P., Mouček R. (2010). Database of EEG/ERP experiments. *In HEALTHINF 2010 - Proceedings of the Third International Conference on Health Informatics*. (pp. 222-227)

Pergler, J. (2009). *Database of ERP experiments – business and presentation layer*. Thesis (in Czech). University of West Bohemia. Pilsen.