# Ontology Development in EEG/ERP Portal

Petr Ježek, Roman Mouček
Department of Computer Science and Engineering,
University of West Bohemia,
Univerzitní 8, 306 14
Pilsen, Czech Republic
Email: {jezekp, moucek}@kiv.zcu.cz

***Abstract*—Substantial difficulties related to collecting data/metadata from EEG/ERP experiments are presented. Because of suitable ontology describing EEG/ERP experiments does not exist authors present a custom ontology that is proposed to be discussed within the scientific community. The presented ontology is practically implemented within the EEG/ERP Portal. The EEG/ERP Portal simultaneously serves to interested community as a tool for sharing and managing of custom EEG/ERP experiments. Because of domain ontologies are expressed by the Semantic Web languages (OWL) authors' present briefly introduction into the Semantic Web technologies. Because of the Semantic Web languages expressivity is different in contrast with the object-oriented code, that is the most often used in current software development, authors' present the developed Semantic Framework. This framework ensures adding a missing semantic into the input object-oriented code and its transformation into appropriate OWL constructs. Integration of the Semantic Framework into the EEG/ERP Portal ensures an online serialization of stored experiments into an OWL document. A registration of the developed ontology in NIF validates authors' approach. The NIF registration also enables easy sharing of stored experiments.**

***Index Terms*—neuroinformatics, electroencephalography, event-related potentials, data management, Semantic Web, Semantic Framework, EEG/ERP experiment, EEG/ERP Portal, ontology**

## I. Introduction

Our research group at University of West Bohemia in Pilsen specializes in the research of brain activity. We use methods of Electroencephalography (EEG) and its derived technique Event-Related Potentials (ERP). The EEG/ERP experiments are time-consuming performed in a specially equipped laboratory. With increasing number of experiments we were facing the problem with their long-term storage and management.

We are a member of the Czech national node of International Neuroscience Coordinating Facility (INCF) that is concerned with the organization of neuroscience data, application of computational models and analytical tools, etc. The one of the objectives of INCF is to define standardized formats and an infrastructure for neurophysiological research. Our aim within the INCF is to provide a complete technological support for storing and sharing of EEG/ERP experiments.

Despite a lot of tools is being developed the potential benefits that the dissemination of knowledge among laboratories offers are still not used. The main reason is that the medical software is supplied by various mostly commercial vendors who are not interested in developing reusable standards. On the other hand, the suitable medium for sharing experimental data should be the Internet. Despite the popularity of the Internet, how it is growing, it contains of a huge amount of information with practically no classification. Such not classified data are not suitable for sharing. As a solution an extension called the Semantic Web that is intended to give the data semantic meaning is being developed. Despite of advantages that the Semantic Web brings common used systems are usually object oriented.

Since semantic gaps between common modeling techniques and the Semantic Web exist we needed to describe the most of serious of them and investigate their removal. We present the proposed mapping and its implementation that extends common object oriented code by missing semantics. The proposed mapping we practically implemented within a custom Semantic Framework.

The ontology that precisely describes the EEG/ERP experiment is presented. Our effort to provide a system for storing and management of EEG/ERP experiments resulted in a custom solution called the EEG/ERP Portal. The developed Semantic Framework generates experimental ontology from the EEG/ERP Portal that is integrated within the NIF registry.

## II. EEG/ERP Experiments

### A. Introduction

EEG/ERP is a technique for recording and interpreting the electrical activity of the brain. This technique is widely used in the research of the brain activity (e. g. reaction of comatose patients, monitoring of drivers' attention, etc.). We perform experiments based on classical oddball paradigm [16]. Such experiments typically contain two stimuli. Stimuli are presented in a random series such that one of them occurs relatively infrequently. The first one presented more often is called non-target and the second one is called target. The tested subject is instructed to be concentrated on the target stimuli. Simple signal averaging procedure is performed continuously during the session after each stimulus subside. Averaging extracts the ERPs from the signal.

### B. Difficulties

Since data obtained during the EEG/ERP experiments are raw data we need to extend them by metadata description. Data/metadata have to be further processable by software tools. It includes searching, interchanging or managing.

Several ways how data/metadata can be represented exist. Data/metadata are usually stored in a relational database. On the other hand, tools that work with data/metadata are usually based on the object oriented programming. Such stored data/metadata can be expressed by the Semantic Web languages to be searchable by automatic readers. Each approach uses a different way to access them. Because of different possibilities for expressing a semantic meaning of the data a need of suitable mapping is emerging.

### C. Experimental Ontology

Since a description of the EEG/ERP experiments is desirable we propose a custom ontology that precisely describes this domain.

When data from experiments are described by suitable metadata in a well-formed structure they may be reproduced and processed across laboratories.

The specification of the ontology originated from experience of our laboratory, co-workers from cooperating institutions, books describing principles of EEG/ERP design and data recording (e.g. [16]) and numerous scientific papers describing specific EEG/ERP experiments. It also corresponds to the effort of the INCF [17] in the field of development and standardization of databases in neuroinformatics.

The experimental data are supplemented by its description according to a defined protocol. The collected metadata can be divided into several semantic groups according to their semantic meaning. We defined the following semantic groups:

*Activity* - describes the activity that takes place according to the specific scenario. It includes information about video, pictures on the computer screen, or sound sloughed off into the headphone, or information about the instructions that received the tested subject when the experiment starts. The description of stimuli is also present.

*Environment* - describes surrounding conditions. It includes: weather, time of the day and temperature.

*Tested subject* - describes information about the tested subject. It includes laterality, education, age, gender, diseases, disability and drugs.

*Hardware equipment* - describes the type, producer and the serial number of the used hardware.

*Software equipment* - describes software that was used during the experiment. It includes the name of the software, the version and the producer, and configuration files if they are used.

*Used electrodes* - describes type, impedance, location, the used system and fixation of used electrodes.

*Data digitalization* - describes a set of parameters that influence conversion of the data using a specific analogue-digital converter. It includes filtration, sampling frequency and band-pass.

*Signal analysis* - describes the technique of analyzing the EEG/ERP signal. It includes the length of pre- and post-stimuli part of the signal, the number of epochs and the verbal description of signal processing procedure.

*Data presentation* - describes experimental results or assumptions needed to reproduce the experiment. It includes Averaged ERP waves (an image of averaged waves), Grand averages (an image of grand averages - an average over more epochs), Evolution of ERP in time and space (images how the ERP is being spread over scalp), Waves description (a description of the well-known or new waves which were formed during study), Link to raw experimental data (a link to all recorded raw data).

*Signal artifact* - Artifacts may originate from muscle activities, eyes movement, etc. It contains information describing a compensation method that prevent the formation of the artifact. When a method for removing of the formed artifact is used the description is also placed there. When some artifact totally degrades the signal the user can define conditions when it is possible to assume the signal as totally useless.

## III. EEG/ERP Portal

### A. Context

Because of hard manual work with large amounts of EEG/ERP data and metadata and difficulties mentioned in II-B, we decided to design and implement a custom software tool suitable for EEG/ERP data and metadata storage and management. As the result we developed the system called the EEG/ERP Portal intended not only for our local research but in general it contributes to advancements in human brain understanding.

The interpretation of EEG/ERP experimental data/metadata is solved by developing a custom ontology described in II-C. The practical implementation of the designed ontology is ensured by the database structure and the internal logic of the developed EEG/ERP Portal. The EEG/ERP Portal provides a possibility to store, get, manage or interchange EEG/ERP experiments among interested researchers. The internal data storage structure is implemented according to the developed ontology.

Several additional modules are also implemented. It includes a module for an interaction within social networks, a library of analytical tools, a semantic web module or web services for an interaction with other systems.

### B. Architecture

The EEG/ERP Portal is web based application. It uses three-layer architecture. This architectonic style is supported by selection of programming tools and technologies. We used Java and XML technologies to ensure a high level of abstraction (system extensibility) as well as a long term existence of the EEG/ERP Portal as open source.

*1) Persistence Layer:* The persistence layer uses the Hibernate framework. It means that a relational database and an object - relational mapping are supported. Oracle 11g database server is used to ensure the processing of large data files.

*2) Application and Presentation Layers:* The application and presentation layers are designed and implemented using the Spring technology. This framework supports MVC architecture, Dependency injection and Aspect Oriented Programming. The Spring Security framework is used to ensure management of authentication and user roles. User access to the system is realized through the web interface. Majority of users are familiarized with web applications and they do not need any additional software except a web browser. The user interface is divided into several parts (the main menu, the second level menu, the header, the footer and a content part). Figure 1 shows the preview of the EEG/ERP Portal.


Fig. 1. EEG/ERP Portal Preview

## IV. Semantic Web Technologies

### A. RDF and OWL

The Semantic Web is a layered architecture. The first layer is called Resource Description Framework (RDF). RDF is a simple metadata representation framework, using URIs to identify web-based resources and a graph model for describing relationships between resources.

Web ontology language (OWL) is a richer language and provides more complex constraints on the types of resources and their properties. OWL comes with a larger vocabulary, greater machine interpretability and stronger syntax than RDF. Because of richer semantic expressivity and needs that the registration in NIF (see. VI) requires we decided to transform data/metadata from the EEG/ERP Portal into OWL.

### B. OOP to OWL Mapping

A description of similar and different concepts of OOP (expressed by UML diagrams) and an OWL expression is given.

*1) Similar Concepts:* Both OWL and UML are based on classes. Relationships among OWL classes are called *properties*. Properties are not necessarily tied to a specific class. The UML properties are translated to *owl:ObjectProperty* if the type of Property is UML Class and *owl:DatatypeProperty* otherwise. Both support a separation into modules, called *package* in UML and *ontology* in OWL. Both support fixed enumeration of elements. Both languages allow the class to be a subclass of more than one class. OWL properties can be constrained by cardinality restrictions. UML also supports cardinality restrictions.

*2) Different Concepts:* In the common OOP the names within one namespace always refer to the same object, and different names always refer to a different object (unique name assumption). Names in OWL do not by default fulfill unique name assumption. Facilities of UML for supporting programs include *operations*, *responsibilities*, *static operations*, *interface classes*, *abstract classes*. In contrast OWL is intended to only represent data and additional data semantics that enables to infer an additional data meaning [3].

### C. Existing Tools

We tested several tools that transform common object oriented code into the Semantic Web languages. The most of tested tools are based on Jena [2]. It provides a program environment for RDF, RDFS, OWL, SparQL languages and technologies.

Mapping of OWL classes to Java Interfaces is described in [11]. Every OWL class is mapped into a Java Interface containing just the accessor/mutator method declarations (set/get methods) for properties of that class.

Back transformation is described in [12] where an OWL processor SWCLOS3 is developed. It is on top of Common Lisp Object System (CLOS). It allows lisp programmers to construct domain and task ontologies in software application fields.

Java2OWL-S is a tool which is able to generate OWL directly [4]. It uses two transformation. First one transforms a java code into a temporary WSDL file. The second one transforms this file into an OWL.

JenaBean [5] is a similar tool, it is a flexible RDF/OWL API to persist JavaBeans. It takes an approach to binding that is driven by the Java object model rather than the OWL or RDF schema.

Concerning one side transformations the tools above work quite satisfactorily because object-oriented code has poorer semantics than OWL. However, if we want to use more capabilities of OWL, an enrichment of object-oriented code by missing semantics is needed.

The ActiceRDF [14] is a library for accessing RDF data from Ruby programs. It provides a domain specific language for RDF models; it can address RDF resources, classes and properties programmatically.

The Semantic object framework (SOF) [15] utilizes embedded comments in source codes to describe semantic relationships between classes and attributes.

The eClass [13] is a solution that changes Java syntax to embed semantic descriptions into the source code.

## V. Semantic Framework

### A. Context

According to Subsection II-B mapping from common object-oriented code is not successfully solved.

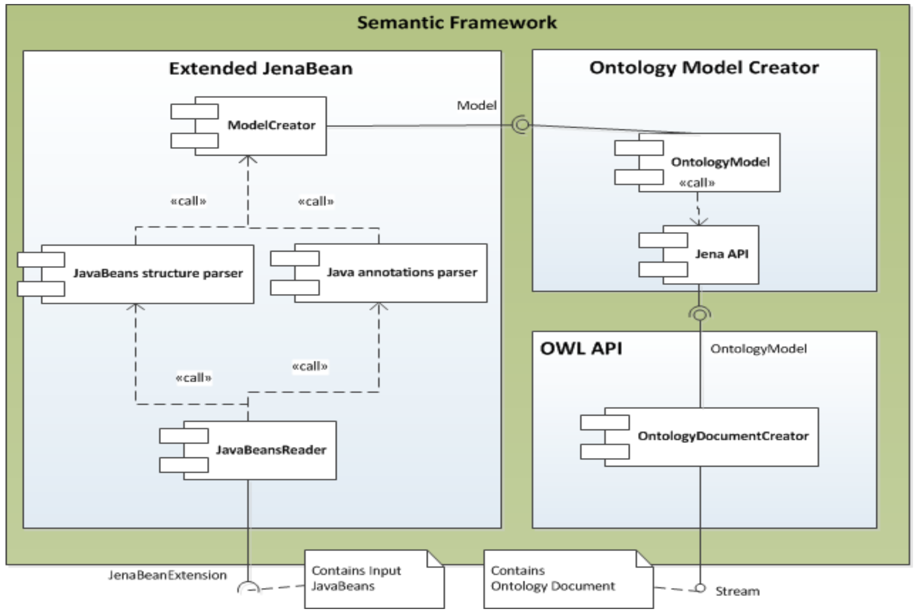


Fig. 2. Component Diagram of Semantic Framework

The tools described in Subsection IV-C are able to transform only a common semantics including classes definitions and its properties.

Some of tested frameworks try to enrich the input object-oriented code by missing semantics using either embedded comments or changing the JavaBean syntax. The usage of such tools is difficult because it requires a modified compiler and java interpreter. Because of the EEG/ERP Portal works with the data stored in common JavaBeans as well we decided to propose a mapping that enables adding missing syntax into the common JavaBean structure. The extension of the common JavaBean structure is based on the idea we initially presented in [10]. The complete set of proposed annotations and its mapping to appropriate OWL constructs we described in [9].

### *B. Implementation*

The Semantic Framework is provided as a library that user can integrate into a custom system. The input is the set of JavaBeans and output is an ontology document. The ontology document can be serialized into the several supported Semantic Web languages syntaxes. We currently support RDF/XML, OWL/XML, Turtle, Abbreviated OWL/XML formats.

Figure 2 shows the component diagram of the Semantic Framework. The framework contains three main subcomponents. The first subcomponent is the modified JenaBean. We extend the current JenaBean so that the output corresponds to the proposed mapping. Moreover, we added the processing of Java annotations into the processing of JavaBeans structure so we are able to transform the set of proposed annotations. The output of the extended JenaBean component is an internal model representation.

This internal model representation is submitted to the second, Ontology Model Creator, subcomponent. This subcomponent creates an Ontology model using an Ontology Model Factory. The internal JenaBean model is processed and an ontology document is created by calling Jena API methods. The result model can be further processed by the last subcomponent, OWL API, that transforms an ontology model into the supported ontology formats.

### *C. Integration within EEG/ERP Portal*

The input point of the EEG/ERP Portal internal logic is created by the set of controllers according to Spring MVC design. The controllers process a HTTP requests and generate a related response. We defined the set of URLs processed by a *SemanticMultiController* that processes an OWL reasoner request. The controller contains methods for getting an OWL document in the supported syntaxes.

When the *SemanticMultiController* processes input parameters it calls a *SemanticFactory* interface. This factory is responsible for getting JavaBeans from the database and handover them to the *Semantic Framework*. When the *Semantic Framework* gets the ontology document it returns a *Data Stream* back to the web browser.

The Semantic Framework is controlled by a build-in timer. The timer calls the Semantic Framework API in regular intervals. It generates the ontology document and stores them into a temporary file. When any document request appears the temporary file containing the actual set of stored experiments is immediately available. This approach ensures a quick response because the ontology document is pre-prepared.

## VI. Registration within NIF

### *A. Introduction*

NIF is a portal that serves as an online inventory of registered neuroscience data sources. Since the NIF is important framework within the INCF community developed according to INCF recommendations [17] we have choose it as the best choice for validating the developed ontology.

### B. Static Context Registration

NIF uses a proprietary framework *Disco* [8]. The registered resource is described using two XML files: *disco.xml* and *disco.rd.xml*. Both have RDF syntax so they are computer-processable on the NIF side. The first file provides a basic description of the registered resource as its URL, contact, etc. The second one provides a more detailed description of the source as a description of partial sections of the system, keywords, publication links, etc. These files are located in the root of our EEG/ERP Portal so they are accessible by the NIF automatic readers.

### C. Dynamic Context Registration

The dynamic content of the EEG/ERP Portal is accessed by the proprietary NIF protocol *Interoperability XML* [1]. The Interoperability XML is used to describe the structure of metadata instances stored in the ontology document. NIF reloads it in regular intervals. When the *Interoperability XML* file is reloaded the ontology document is reloaded as well.

## VII. Conclusion

Scientific papers usually do not solve interpretation, storage or interchanging of experimental data/metadata.

With a need of sharing EEG/ERP experiments relates a need to describe them by a well-defined metadata together with a need to find a suitable medium for data/metadata sharing. The Internet seems to be appropriate to share experiments. However, due to limits that the current web gradually reaches a parallel web called the Semantic Web is being developed. The Semantic Web that expresses meaning of data by domain ontologies is supposed to solve the problem of missing semantics of the current web.

Development of specific ontologies is crucial task in the creation of the Semantic Web. However, current software systems are usually based on object-oriented programming languages. Since fundamental differences between semantics of the object-oriented code and Semantic Web languages exist, we proposed and implemented an extension of current JavaBeans by Java Annotations that fill existing semantic gaps. Proposed mapping was implemented in the presented Semantic Framework.

Difficulties relating to description of EEG/ERP experiments were solved by designing the ontology that describes metadata. The EEG/ERP Portal serves the community as the practical tool for storing, managing and interchanging custom experiments. The internal structure of the system is designed to satisfy restrictions given by the ontology. The EEG/ERP Portal was registered in NIF. It ensures accessibility of stored experiments to interested researchers and the verification of the designed ontology.

The presented approach produces the ontology document that is generated in OWL. OWL is divided into three sublangues (Lite, DL, Full) with different semantic expressivity. The ontology fully satisfies OWL DL but it does not use all constructs of OWL Full. Since OWL Full e.g. does not enforce a strict separation of classes, properties, individuals and data values it is problematic to map object-oriented constructs (where such strictly separation is enforced) to equivalent OWL constructs. In the future we plan to investigate a strict separation between OWL specifications and clearly define which OWL constructs are possible to express by object-oriented languages and where it is fundamentally impossible.

Since many limitations of OWL exist the extension (called OWL2) has been started developed [7]. OWL2 aims is to remove issues of different syntaxes, improve datatype expressivity, provide better organization of imports or remove difficulties with different versions of OWL syntaxes [6]. As the significant next step we plan to start with transformation from the object-oriented code into the OWL2 syntax.

## Acknowledgment

This work was supported by the European Regional Development Fund (ERDF), Project ”NTIS - New Technologies for Information Society”, European Centre of Excellence, CZ.1.05/1.1.00/02.0090.

## References

[1] Ncbo bioportal, 2010. http://bioportal/bioontology.org/ontologies/40510.
[2] Apache Consorcium. Apache jena, June 2010. http://jena sourceforge.net/.
[3] OMG Consorcium. Ontology definition metamodel, 2011. Http://www/omg.org/spec/ODM/1.0/PDF/.
[4] DAML. java2owl, May 2012. http://www.daml.org/2003/10/java2owl/.
[5] Apache Foundation. jenabean, may 2010. http://code.google.com/p/jenabean/.
[6] I. Graua, B. C. Horrocksa, B. Motika, B. Parsiab, P. Patel-Schneiderc, and U. Sattlerb. Owl 2: The next step for owl, web semantics. In *Services and Agents on the World Wide Web, Volume 6*, pages 309–322. Elsevier, November 2008.
[7] W3C OWL Working Group. Owl web ontology language reference, 2011. http://www/w3.org/TR/owl-ref/.
[8] A. Gupta, W. Bug, L. Marenco, X. Qian, Ch. Condit, A. Rangarajan, H. M. Müller, P. L. Miller, B. Sanders, J. S. Grethe, V. Astakhov, G. Shepherd, P. W. Sternberg, and M. E. Martone. Federated access to heterogeneous information resources in the neuroscience information framework (nif). *Neuroinformatics 6*, pages 205–217, 2008.
[9] P. Ježek and R. Mouček. Semantic web in eeg/erp portal. In *4th International Conference on Biomedical Engineering and Informatics*, pages 2076–2080. New York: IEEE, 2011.
[10] P. Ježek and R. Mouček. Transformation of object-oriented code into semantic web using java annotations. In *13th International Conference on Enterprise Information System*, pages 207–210, 2011.
[11] A. Kalyanpur, D. J. Pastor, and J. A. Padget. Automatic mapping of owl ontologies into java. In *Software Engeering and Knowledge Engeering*, pages 98–103, June 2004.
[12] S. Koide, J. Aasman, and S. Haflich. Owl vs. object oriented programming. In *International Workshop on Semantic Web Enabled Software Engineering (SWESE)*, 2005.
[13] F. Liu, J. Wang, and S. T. Dillon. Web information representation, extraction and reasoning based on existing programming technology. In *Computational Inteligence 37*, pages 147–168, 2007.
[14] E. Oren, R. Delbru, S. Gerke, A. Haller, and S. Decker. Activerdf: object-oriented semantic web programming. pages 817–824, 2007.
[15] Ch. Po-Huan, L. Chi-Chuan, and Ch. Kuo-Ming. Integrationg semanic web and object-oriented programming for cooperative design. *Journal of University Computer Science,vol. 15, no. 9*, 2009.
[16] J. Luck Steven. *An Introduction to the Event-Related Potential Technique (Cognitive Neuroscience)*. A Bradford Book, 2005.
[17] J. van Pelt and J. van Horn. 1st incf workshop on sustainability of neuroscience databases. *Workshop report*, 2007. Stockholm.