University of West Bohemia in Pilsen
Department of Computer Science and Engineering
Univerzitni 8
30614 Pilsen
Czech Republic

Database of EEG/ERP experiments

The State of the Art and the Concept of Ph.D. Thesis

Petr Ježek

May, 2010

Distribution: public

Abstract

This work summarizes problems occurred with storing data from EEG/ERP experiments. It address problems with EEG/ERP data formats, metadata description or sharing experiments between laboratories. Work shortly introduces background of EEG/ERP experiments and laboratory equipment. Existing formats including EDF, ARFF, WEKA or VDEF are presented with description of its advantages and difficulties. Difficulties of neuroscience databases and existing databases are also presented. There is an organizatoin called INCF that recommends how to make databases sustainable, those recommendation are presented. Because the internet is suitable medium for sharing of experiments and the semantic web provides possibilities how to represent metadata of experiments, work is either focused on the semantic web technologies. It describes common languages with their interpretative capabilities and difficulties. Since nowadays data are usually stored in the relational databases or represented by object oriented model hence possibilities how to represent and semanticaly describe sense of data by relational or object-oriented model is presented in comparison with the semantic web model. Mapping between those representations is analyzed together with description of existing tools. In the last part of developed portal is presented.

Contents

I Opening 6

1 Introduction 7

1.1 Problem Overview 8

1.2 Document Structure 9

II Background and State of The Art 11

2 EEG/ERP Experiments 12

2.1 EEG/ERP Introduction 12

2.1.1 Biological Background 12

2.1.2 Electroencephalography 13

2.1.3 Event-Related Potentials or Evoked Potentials 14

2.1.4 ERP Components 14

2.2 EEG/ERP Experiments 15

2.2.1 The Oddball Paradigm 15

2.2.2 Simple Example Experiment 15

2.2.3 ERP Laboratory 17

3 EEG/ERP Data Formats 19

3.1 Formats Overview 19

3.2 European Data Format 19

3.2.1 Specification 20

3.2.2 Disadvantages 20

3.3 Vision Data Exchange Format 21

3.3.1 Specification 21
3.3.2 Disadvantages 22
3.4 Attribute-Relation File Format 22
3.4.1 Specification 22
3.4.2 Disadvantages 23

4 Neuroscience Databases 24
4.1 Sustainability 24
4.1.1 INCF Recommendations 25
4.2 Available Databases 26
4.2.1 CARMEN Portal 26
4.2.2 INCF Japan Node - Portal of Neuroinformatics 28
4.2.3 Neuroscience Information Framework 29
4.3 Neuroscience Databases Conclusion 31

5 Semantic Web Technologies 32
5.1 Introduction 32
5.2 Semantic Web Overview 33
5.3 Technologies 35
5.3.1 Resource Description Framework 36
5.3.2 Ontology Web Language 40
5.3.3 Relational Model 43
5.3.4 Mapping Between Relational and RDF Model 44
5.3.5 Mapping Between OWL and OOP 46
5.4 Existing Tools and Frameworks 49

III Current Status and Future work 52

6 EEG/ERP Portal 53
6.1 System Context 53
6.2 Specification of Requirements 53
6.3 Project Scope and System Features 54
6.4 User Roles 54

6.5 Definition of Metadata 55

7 Conclusion 57

7.1 Aims of PhD Thesis 58

List of Figures

2.1.1 Neurone structure [3] 13
2.2.1 Graph segment for non-target stimulus 16
2.2.2 Graph segment for target stimulus 17
2.2.3 Laboratory Equipment [21] 18

4.2.1 JNode activities 29

5.3.1 the semantic web layered architecture [22] 35
5.3.2 Example of RDF graph describing person named Joe Smith [24] 37
5.3.3 RDF and RDFS layers [22] 39
5.3.4 OWL variants 42

Part I

Opening

Chapter 1

Introduction

Nowadays the methods of Electroencephalography (EEG) or especially Event-Related-Potentials (ERP) are widely used in the research focused on driver's attention, prediction of micro sleep or reaction of comatose patients or seriously injured people. These methods are relatively cheap and non-invasive to tested subject. Naturally this research requires doing a lot of experiments with scenarios focused on e. g. attention, tiredness or concentration of tested subject.

There are a lot of data obtained from these experiments. The data should be stored in order to couls be used in the future or interchange between various laboratories.

Many books and articles focused on doing experiments was written (e. g. [4]) but they don't solve how to data store and manage. Data from EEG/ERP experiments are usually stored in the neuroscience databases. Neuroscience databases provide a diverse collection of communities with access to meta- and raw-data. Data in the neuroscience databases are stored in the diverse data formats.

Because neuroscience databases broaden and extent the scope of data stored there is need to provide some standard how to store them effectively. Data stored in those databases should be reachable, viewable, and suitable for secondary exploration far beyond the purpose of their original collection. Although effort of groups of interested researchers is to design and develop some standards how to store data, nowadays an universal format or an universal database not exist.

When data from experiments are obtained there is also a problem with their description by suitable metadata. Raw data without their description are useless because interpret their meaning is difficult. Metadata description relevant with experimental scenario, hence there are several requirements to metadata items relevant with requirements to experiments and vice versa. Although some existing formats provide possibilities to provide some metadata description, a well formed structure does not exist.

1.1 Problem Overview

Nowadays there is no problem to obtain data from EEG/ERP experiments but there is a problem how to store and manage data. EEG and ERP experiments take usually long time and produce a lot of data. With the increasing number of experiments carried out there is necessary to solve their long-term storage and management. With storing EEG/ERP data and metadata there relevant series of disadvantages:

- There is no widely spread and generally used standard for EEG/ERP data and metadata format within the neuroscience community.
- Results of EEG/ERP experiments are usually more important than raw data. Data without their description are difficult to evaluate.
- There is no reasonable and easily extensible tool for long-term EEG/ERP data/metadata storage and management. General practice is to organize data and metadata in common files in directories.
- Generally there is no practice to share and interchange data between EEG/ERP laboratories. EEG/ERP data are supposed to be secret or unimportant to share them.
- Data from experiments are usually not published therefore they are not available to researchers interested in EEG/ERP research, data mining

or signal processing or to researchers who don't have their own laboratory.

This work summarizes this disadvantages and is trying to find a solution. The following issuses are described:

- Existing data formats and their advantages and disadvantages.
- How experiments are performed and metadata which are need according to experiments requirements are found.
- Describtion of existing experiments and metadata definition.
- Design a suitable ontology for ERP domain.
- Existing neuroscience databases and their possibilities.
- Possibilities how to make experiments available by using web browser.
- Since internet continuously grows. Possibilities of semantic web in ERP domain were described in order to find of experiments were easier.

1.2 Document Structure

Chapter 2 describes what EEG and ERP exactly means and what EEG/ERP experiments involve. In the short description is noted how is arranged laboratory and which is used equipment, how is scenario of experiments and which are expected brain responses. Naturally short overview of biological background about working of human brain have to be included .

This chapter is followed by Chapter 3 describes available data and metadata formats. In each format is described its advantages and disadvantages and why no one is not used as a standardized formats. With formats description relevant their internal structure definition. It includes both raw data and metadata description (if metadata are available).

There are many international organizations producing neuroscientific data. Some of them have developed their own databases where they publish data from their research. There is also a diverse group of neuroscientific interfaces where data from experiments are not published directly but own data source is possible to register there. These databases collect registered data sources.

Chapter 4 is focused on international neuroscience databases. It describes international nodes which participate in the neuro research and describes how they want to solve storing, preserving and interchanging data and metadata. This chapter also describes recommendations for creating neuroscience databases [15].

Chapter 5 describes technologies of the semantic web. It describes interpretative possibilities of the semantic web in comparison with relational databases and object-oriented model. It provides description which languages are used in the semantic web, what techniques are used to transform data from object-oriented or relational model into semantic web. Description of existing tools with their advantages, disadvantages and differences is mentioned as well. Finally a set of suitable tools for transforming data from neuroscience databases into semantic web representation is presented.

Chapter 6 describes the developed portal for management of EEG/ERP experiments. This portal serves as a base tool for management of EEG/ERP experiments. It also serves as a base for developed semantic web engine.

Part II

Background and State of The Art

Chapter 2

EEG/ERP Experiments

2.1 EEG/ERP Introduction

Before discussing experiments it is necessary to introduce electroencephalography and event-related potentials and how the brain works.

2.1.1 Biological Background

The core component of the nervous system (including brain, spinal cord, and peripheral ganglia) is a neuron. It is an electrically excitable cell that processes and transmits information by electrochemical signaling via connections with other cells called synapses. Neurons are called nerve cells. A neuron is basically an on/off switch. It is either in a resting state or it is shooting an electrical impulse down an axon. On the very end of axon path there is a little part that shoots out a chemical. This chemical goes across a gap (called synapse) where it triggers another neuron to send a message. Figure 2.1.1 on page 13 shows a structure of a typical neuron [3].

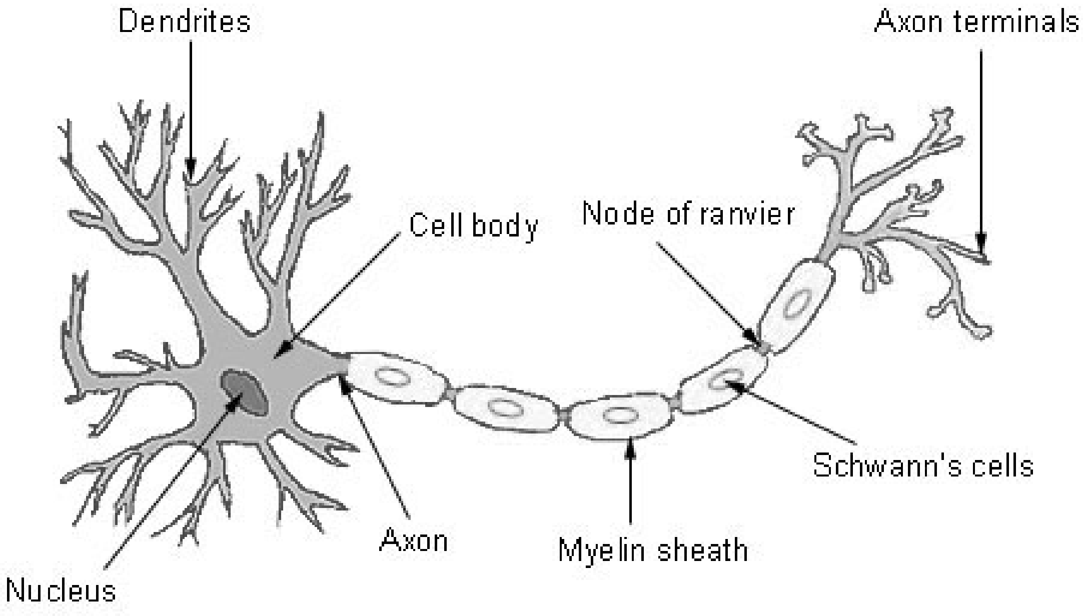

Figure 2.1.1: Neurone structure [3]

2.1.2 Electroencephalography

Electroencephalography (usually abbreviated EEG) is a technique for recording and interpreting the electrical activity of the brain. It is a non-invasive method. The nerve cells of the brain generate electrical impulses that fluctuate rhythmically in distinct patterns. To record the electrical activity of the brain, pairs of electrodes are attached to the scalp. Each pair of electrodes transmits a signal to one of several recording channels. This signal consists of the difference in the voltage between the pair. The rhythmic fluctuation of this potential difference is shown as peaks and troughs on a line graph by the recording channel dependence on time. This graph is named electroencephalograph (extracted from [1]).

2.1.3 Event-Related Potentials or Evoked Potentials

Event-related brain potentials or Evoked Potentials[1] (usually abbreviated ERP resp. EP) are derived techniques from EEG. The methods are non-invasive, brain activity during cognitive processing is measured. The transient electric potential shifts (so-called ERP components) are time-locked to the stimulus onset (e.g., the presentation of a word, a sound, or an image). Each component reflects brain activation associated with one or more mental operations. In contrast to behavioral measures such as error rates and response times, ERPs are characterized by simultaneous multi-dimensional on line measures of polarity (negative or positive potentials), amplitude, latency, and scalp distribution. Therefore, ERPs can be used to distinguish and identify psychological and neural sub-processes involved in complex cognitive, motor, or perceptual tasks. Moreover, unlike next technique used for registering brain activity as is magnetic resonance imaging (MRI) or functional magnetic resonance (fMRI) (even Event-Related fMRI, which precludes the need for blocking stimulus items), ERP provides extremely high time resolution, in the range of one millisecond (extracted from [2]).

2.1.4 ERP Components

The method of averaging is used for obtaining ERP from EEG. When ERP experiment is recorded simultaneously with brain activity a position of stimulus is stored (by creating markers in the signal). The single-trial waveforms, is creating averaged ERP waveforms for each type of stimuli at each electrode site. By doing this averaging at each time point folowing the stimulus its end up with highly replicable waveforms for each stimulus type.

The resulting averaged ERP waveforms consist of a sequence of positive and negative voltage deflection, which are called components.

The components are designated by letters P, N or C. P is used for positive signal, N for negative signal and C for components which are not completely positive or negative but their polarity vary. The letter is typically folowed

[1] In this work it supposed that there is no difference between Event-Related Potentials and Evoked Potentials terms

by number which quantifies latency of the wave in milliseconds. For instance there is a component named P300 which is very often used in experiments based on oddball paradigm described in the next section. It signifies component with positive amplitude detected after 300ms stimuli onset. Notation of components is sometimes shorten so that we can see P3 instead of P300 but the meaning is the same (extracted from [4]).

2.2 EEG/ERP Experiments

2.2.1 The Oddball Paradigm

The experiments based on oddball paradigm 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 second one is called "target". Stimuli could be audio (two different tones, beeps or voices) or video (two different signs, pictures, letters or digits on the screen). The rate between stimuli is approximately 20 percent for target to 80 percent for non-target. Tested subject is instructed to be concentrated to target stimuli or to do nothing [4, 5].

2.2.2 Simple Example Experiment

This section describes simple EEG/ERP experiment used for demonstation how to obtain P3 component from EEG signal. This experiment is done in our laboratory according to experiment described in [4].

The experiment is a variant on the classical oddball paradigm. Subjects view sequences of 80 percent letters Os (non-target stimuli) and 20 percent Qs (target-stimuli) and they calculate how many times Q (target stimuli) occurs. Each letter is presented on a video monitor for 100ms, followed by a 1 400ms blank interstimulus interval. While the tested subject perform this task, EEG from several electrodes embedded in an electrode cap is recorded. The EEG is converted into digital form and is stored on a hard drive. Whenever a stimulus is presented the stimulation computer sends a marker code to the

EEG digitization computer, which stored them along with EEG data.

A simple signal averaging procedure is performed continuously during session after each stimulus. It extracts the ERPs elicited by the Os and the Qs. Specifically, the segment of EEG surrounding each Q and each O is extracted and lined up these EEG segments with respect to the marker code.

Figure 2.2.1 on page 16 shows how ERP signal looks for non target stimulus O. Onset of stimulus is inserted into coordinate origin, there is evidently that signal has still a similar amplitude.

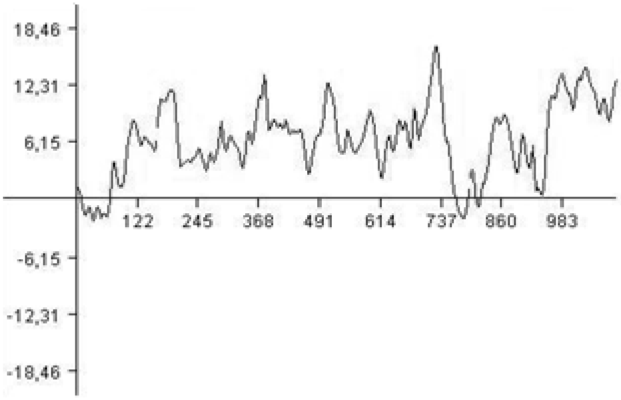

Figure 2.2.1: Graph segment for non-target stimulus

Figure 2.2.2 on page 17 shows how ERP signal looks for target stimulus Q. Onset of stimulus is inserted into coordinate origin as well. Approximately after 300ms it is possible to see positive peak with much higher amplitude then neighboring extremes.

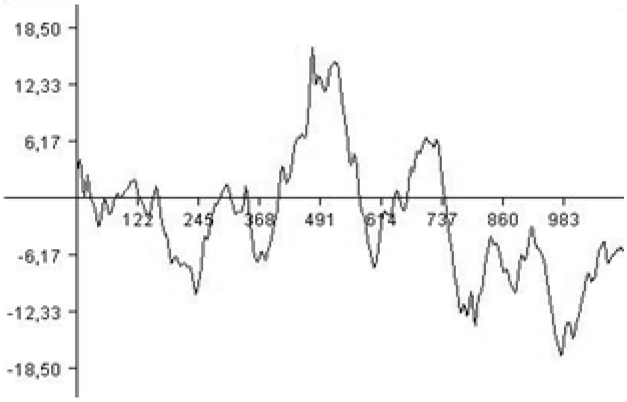

Figure 2.2.2: Graph segment for target stimulus

2.2.3 ERP Laboratory

In order to perform ERP experiments we have laboratory with special equipment. In this laboratory we use 32-channels EEG recorder BrainAmp with BrainVision recording software and our own software for presenting experimental scenarios. We use two computers. The first one is for playing scenarios and second one for storing EEG data and watching progress of experiment. Both computers are connected together by USB adapter in order to store markers from scenario. Tested subject is sits on the seat, he/she has an EEG cap on the head and is watching scenarios of experiment on the screen. Attendant person is present during experiment in order to instruct a tested subject. Laboratory equipment is presented in the Figure 2.2.3 on page 18

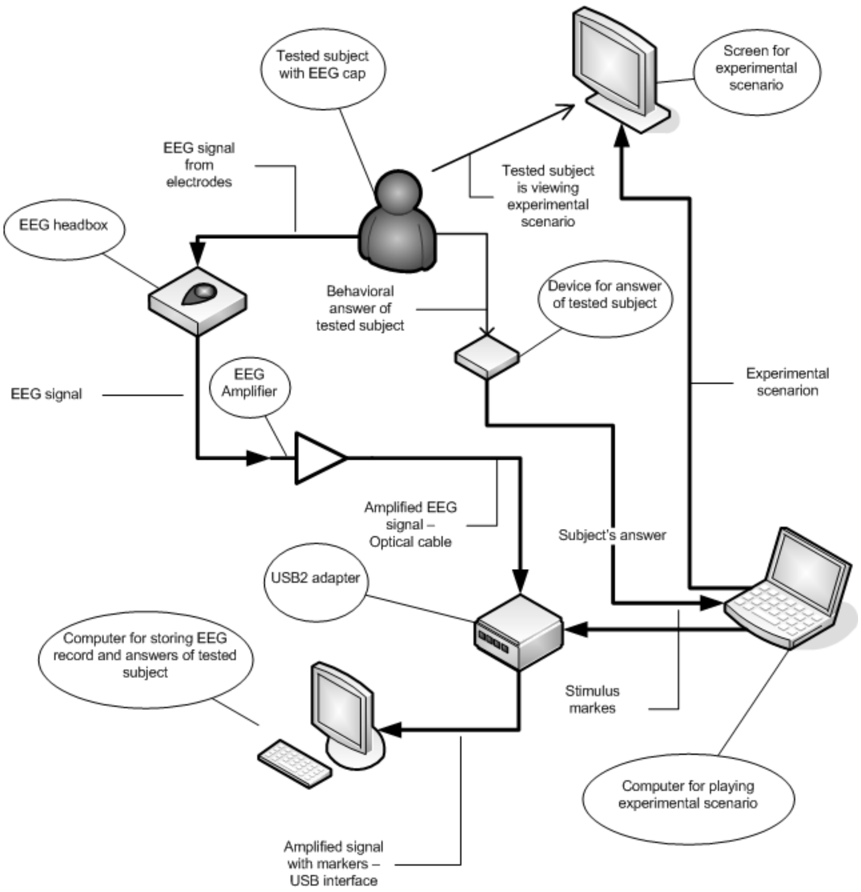
Tested subject with EEG cap
Screen for experimental scenario
EEG signal from electrodes
Tested subject is viewing experimental scenario
EEG headbox
Behavioral answer of tested subject
Device for answer of tested subject
EEG signal
EEG Amplifier
Experimental scenarion
Amplified EEG signal – Optical cable
Subject's answer
USB2 adapter
Computer for storing EEG record and answers of tested subject
Stimulus markes
Computer for playing experimental scenario
Amplified signal with markers – USB interface

Figure 2.2.3: Laboratory Equipment [21]

Chapter 3

EEG/ERP Data Formats

3.1 Formats Overview

When EEG potentials are obtained from scalp of tested subject, they have to be digitalized for machine processing. Special devices called analog-digital converter convert data into digitalized form. Producers of this converter are responsible for output format specification. Since there are many of producers of EEG recording devices and they profit from selling own solution there is no general endeavor in order to make it compatible with each other producers or make it as an open source.

In this chapter most often formats used for storing EEG data and metadata are described. Some from described formats was mostly developed by commercial companies. Reading or storing data usually requires using among supplied commercial software. Other described formats are open source.

3.2 European Data Format

The European Data Format (abbreviated EDF or EDF+ used for its extension) is a simple format for exchange and storage of multichannel biological and physical signals. It was developed by a few European medical engineers in 1987 who met on international Sleep Congress in Copenhagen. With the support of professor Annelise Rosenfalck, the engineers initiated the European project on Sleep-Wake analysis (1989-1992). They wanted to apply their sleep analysis algorithms to each others data and compare the analysis results. So, in Leiden in March 1990, they agreed upon a very simple common data format. This format became known as the European Data Format first introduced in 1992 published in [8] (extracted from [7]).

3.2.1 Specification

One data file contains one uninterrupted digitized polygraphic recording. A data file consists of a header record followed by data records. The first part of header contains a set of metadata that identify tested subject, contains recording identification, time information about the recording, the number of data records and finally the number of signals in each data record.

The first part of header is 256 bytes length and it is followed by the second part of header record that specifies type of signal, amplitude calibration or number of samples in each data record. The length of the second part is 256 bytes for each signal so total header length is possible to express by (3.2.1). Header is followed by data record where each sample is represented by two bytes integer.

$$header\,length = 256b + (ns * 256b)\,;\,ns = signals\,count \tag{3.2.1}$$

Although this format is used in some commercial (e.g. Walter Graphtek [9] or xltech [10]) and in many of open source readers and writers (e.g. Brainlab [11] or OpenXDF [12]) this format has several disadvantages.

3.2.2 Disadvantages

Firstly, raw data and metadata are in one file together. In common formats there is no general habit to mix binary and text data together. Secondly, metadata contain only a restricted set of information about tested subject. Further, format is not determinates for ERP experiments directly that is why there is no possible to store markers into signal. Finally, information about experimental scenario is missing totally.

Despite its drawback this data format has been probably the most hopeful attempt to standardize description of EEG data.

3.3 Vision Data Exchange Format

Vision Data Exchange Format (VDEF) is produced by BrainAmp device designated for reading EEG/ERP. This format could be read using the Vision Recorder developed by BrainProduct company [13]. Software and hardware equipment are used in our laboratory where we do EEG/ERP experiments. The Vision Recorder has the following features:

- User can controll different amplifiers, also program enables new EEG/ERP formats to be integrated with the aid of independent components.
- The number of channels is only restricted by the amplifier that is in use. The internal structure supports an unlimited number of channels.
- Segmentation based on event markers is available to reduce the space required by EEG/ERP files.
- Averaging based on event markers is available to form ERP during recording.
- The data can be filtered separately for display, for segmentation or averaging and for storage.

Text in this section was extracted from [13]

3.3.1 Specification

The format consists of three files (the header file, the marker file and the raw data) that have to be stored in one folder together. The header file describes the EEG/ERP. This file is an ASCII file with the extension ".vhdr". It will normally be given the same base name as the raw data EEG/ERP file that is described in it. It also contains name of marker and raw data files, data format, number of channels, sampling interval and for each channel number, reference channel name, channel name, resolution and resolution unit. The format of the header file is based on the Windows INI format. The last, marker file, contains name of data file, used encoding and for each marker their number, type, description, position, size and channel number (Extracted from [13]).

3.3.2 Disadvantages

Although this format solves many disadvantages of EDF data format, especially that data and metadata are stored separately into diverse files and format is directly used for ERP experiments (provides possibility to store markes), several disadvantages remains open.

Format does not define metadata about scenario of experiment thus they cannot be stored. Because the format is a commercial its acceptance as a standardized format is questionable.

3.4 Attribute-Relation File Format

Attribute-Relation File Format (ARFF) is used internally by the Weka Machine Learning Project [14]. WEKA is a collection of machine learning algorithms for data mining tasks written in Java. It contains tools for regression, association rules, clustering, data pre-processing, classification, and visualization. It is also suitable for developing a new machine learning schemes. In our department we have use several tools for WEKA Software.

3.4.1 Specification

ARFF file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF file contains two section; Header and Data. Header part is marked by “header” annotation and contains the name of the relation, a list of attributes and their types.

Data part is marked by “data” annotation and contains a set of values separated by comma. Attributes in the header part have to be ordered and they define the name of the attribute and its data type. The order of the attributes define the column position in the data section of the file. For example, if an attribute is the third one declared then Weka expects all that attribute values will be found in the third comma delimited column in data section.

3.4.2 Disadvantages

Although ARFF is an open source format publish under the General Public License as well as whole WEKA project they could be more extended, real situation is that is used only in the WEKA project locally.

The format does not provide almost not possibilities how to store metadata of experiments. Because data from each channel are stored in one text file together with metadata searching and seeking in text file is problematic. Also, there is no possibility how to store markers from ERP experiments.

Chapter 4

Neuroscience Databases

4.1 Sustainability

Neuroscience databases are young and dynamic field with many developments still have to be done. Databasing already gives a new flavor to the term neuroinformatics emphasizing high-throughput technologies for data generation, systematic large-scale data collation and presentation, and the development of computational tools that allow researchers to extract features and relation ships among ever-grooving amounts of data.

Neuroscience databases are provided by a diverse collection of neuroscientists. These databases provide a set of analytical tools or computational models and some of them provide possibilities for storing raw data and metadata from experiments. These resources could be useful in new research, development of methods and scientific education. The development of these databases requires several years of work focused on researchers needs with active researchers cooperation.

Nowadays there is a question how these databases sustain their activities in the long therm. There is an organization called International Neuroinformatics Coordinating Facility (INCF) which organized the 1stINCF Workshop on Neuroscience Database Sustainability. The goal of this workshop was to discuss issues related to the sustainability of neuroscience databases, identify problems and discuss solutions or approaches to these problems, and formulate recommendations (extracted from [15, 16]).

4.1.1 INCF Recommendations

INCF formulated several recommendations that should be followed when neuroscience databases are created in order to ensure long term sustainability. Extraction of recommendation useful for this work is in the next text. INCF recommends [15]:

- Clearly define the community (audience for the resources), identify roles and needs of each, provide mechanisms for incorporating feedback (wiki, bulletin, boards, etc.).
- Develop focused but flexible standards, follow best practices, make standards open to community.
- In developing of infrastructure for data sharing and sustainability it is critical to understand how neuroscience community is organized and how it works with data.
- Data can be safely expressed in relational schema. A comprehensive data model, integrating datasets, documents and annotations are needed. Large neuroscience datasets should be isolated.
- To use open source solutions in the maximal range, including XML, Web-Services or semantic web technologies, adherence to standards (ISO) is important.
- Datasets could be replicated at the central site, have to be formulated on ethical and patent/copyright issues, and users identificational requirements for integrated datasets.
- Technical issues include grid and web service security, access control, single sign on, etc. should not be missied out.
- INCF could identify the data resources with highest information value, and the interconnections between these resources. Then, INCF can

specify which resources shall be preserved and at which schedule, which resources are not sustained, and which resources have a low information value and do not need to be sustained.

- Databases should be based on defined ontologies and schemata that are portable (in visible formats). They should allow import/export of database data in exchange formats. Query engines must be integral to databases and be defined explicitly. Languages and source code specifications must be provided for database applications.
- Data should have a markup language with metadata info for formats, experimental information, granularity, description of terminology, and minimal standards. It should be portable, scalable and extensible, and needs an ontological framework on which the data is based.
- The Web should be taken as a standard for interfaces (user interface). Each interface must have a defined API, with specifications for graphical interfaces, portability, query, and use cases.

Generally INCF should establish and moderate web-based infrastructure, identifiyng specific types of data/databases and investigate existing neuroscience data. Full text of recommendations is available in [15]. Our effort in current and future work is to respect the recommendations in the maximal range.

4.2 Available Databases

There are several databases developed for storing and getting together neuroscientific data. This section introduces solutions that are available. The main advantage of introduced databases is that they provide possibility to register own data source within these databases.

4.2.1 CARMEN Portal

CARMEN is a project funded by the Engineering and Physical Sciences Research Council (UK). The system CARMEN has been designed to allow neuroscientists to share data and programs from neurophysiological experiments amongst collaborators, in a secure and formally annotated manner. Core of the CARMEN is a data storage resource which is available to end-user through web interface.

The portal provides to user a set of following objectives:

- To search achieved data
- To upload, annotate and store own experimental data
- To run processes and routines on the stored data on the CARMEN computers.

Searching the data stored in the portal is possible by using search box in the system. This search box provides text field where user puts entry key words and relevant set of results is obtained. Data could be signed as a private or as a public. Not logged user can see only public data.

The system provides possibility to show the metadata associated with archived data and download the data for local processing.

Registered users can upload experimental data to the CARMEN system. Uploading process consists several forms where user fills metadata describing inserted data.

The Portal also enforces a privacy on archived data. Through a simple user interface the end user can specify who has access to the stored data/metadata that they have uploaded. Data and metadata can be signed as public, private. accessible only to the logged user, or protected via access control lists, such that only predefined set of registered users have access to the data.

In addition there is possible to store and achieve analysis tools that were used with data processing. It allows collaborators to share tools, methods and algorithms, and provides means to run the analysis on the CARMEN computer resources. Uploaded tools are implemented as a web-services in order to could be called locally from user's computer without their downloading. There is also an access control list where is defined who can call particular services. Services could not be uploaded directly by user but user has to contact the CARMEN system support staff.

4.2.2 INCF Japan Node - Portal of Neuroinformatics

The Japan Node of the INCF (JNode) coordinates neuroinformatics activities within Japan and represents Japanese efforts in INCF. Japan Node mainly domestics neuroinformatics research and directions, advises on Intellectual Property Rights and protects experimental subjects, develops and publishs brain science databases, coordinates database management, disseminates neuroinformatics information via the web portal, develops the infrastructure for brain science information and neuroinformatics and supports the development and diffusion of neuroinformatics technology.

Activities of Japan node with relation to INCF are shown in Figure 4.2.1 on page 29 [18].

Except mentioned activities JNode has developed the portal of neuroinformatics where is possible to find links to web sites of other organizations with participants on the neuro research.

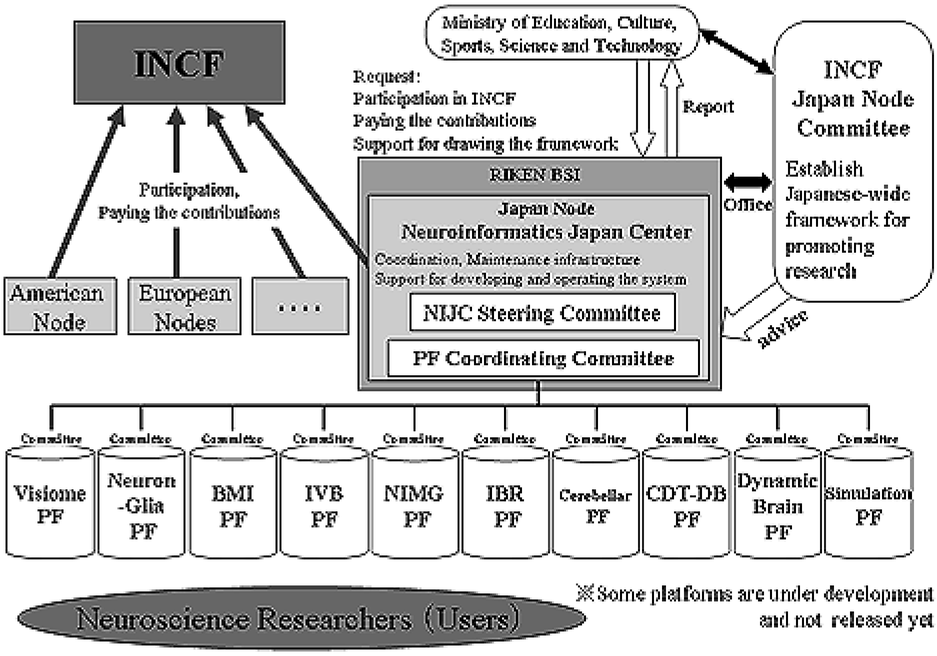
INCF
Ministry of Education, Culture, Sports, Science and Technology
INCF Japan Node Committee
Establish Japanese-wide framework for promoting research
Request:
Participation in INCF
Paying the contributions
Support for drawing the framework
Report
Participation, Paying the contributions
RIKEN BSI
Office
Japan Node
Neuroinformatics Japan Center
Coordination, Maintenance infrastructure
Support for developing and operating the system
NIJC Steering Committee
PF Coordinating Committee
advice
American Node
European Nodes
....
Committee
Visiome PF
Neuron -Glia PF
BMI PF
IVB PF
NIMG PF
IBR PF
Cerebellar PF
CDT-DB PF
Dynamic Brain PF
Simulation PF
Neuroscience Researchers (Users)
※Some platforms are under development and not released yet

Figure 4.2.1: JNode activities

4.2.3 Neuroscience Information Framework

Neuroscience Information Framework (NIF) is a dynamic inventory of Web-based neuroscience resources. It includes data, materials, and tools accessible via any computer connected to the Internet.

Effort of NIF is to advances neuroscience research by providing possibilities to access public research data and tools through internet with requirements to use open source.

NIF is created by several participant universities including University of California, San Diego, California Institute of Technology, George Mason University, Yale University Medical College, and Washington University.

In the portal it is possible to connect to web seminars arranged by NIF community where users could connect through internet into the arranged seminar time and talk about seminar topic. They have developed a comprehensive vocabulary for annotating and searching neuroscience resources. The vocabularies are available for download as an OWL[1] files and also through the NCBO BioPortal[20]. For informing about news they publish community news and provide Neuro Wiki. Many more tools in the current version of NIF portal are available in [19].

Probably the most usefull feature is possibility to register own data source. Registered resources are actively seeking to be available through NIF. The goal of NIF is to enable users to register his/her database within portal. NIF portal has indexed registered data sources. When an interested user wants to search some data he/she will accesses NIF portal, put key words into searcher and NIF portal searches data in registered databases, so he/she can search over a lot of databases by using uniform interface. NIF does not maintain any resources locally.

User who wants to register own data source can make a choice from three levels (extracted from [19]):

1. Level 1 - Registration requires providing URL of user's data source and basic information about the type of data source. This level places data source into NIF registry where is available through NIF web portal but does not provide direct access to dynamic content.

2. Level 2 - It uses XML-based script to provide a wrapper to a web site that allows searching for key details about a requested data source including dynamic content. Content wrapping is ensured by special tool named DISCO[2].

3. Level 3 - This level knits independently maintained databases into a virtual data federation by registering of a schema information and databases views within NIF portal. This concept maps tables fields and values into the NIFSTD ontology[3]. Data within a source database can be combined with other databases by defining an integrated view

[1] Ontology Web Language is described in the Chapter 5

[2] It is the tool used as a gateway to the neuroscience database, it provides machine understandable information to integrator servers (developed by Dr. Luis Marenco at Yale University)[19].

[3] NIF Standard Ontology is composed of a collection of OWL modules covering distinct domains of biomedical reality

across databases. It means that individual databases may be small but user access this data source as one virtual large database.

4.3 Neuroscience Databases Conclusion

Several most known databases in the neuroscience were introduced in this section. CARMEN is well-designed portal where user can make own user account and share data from experiments. When user uses some additional tools for data processing he/she can provides this tools as a web service. Data and services can be public or private according to owner decision. The portal provides a good solution for users who want to share own experiments and don't have own portal where they could provide own experiments for download. Portal is also suitable for users who are interested in neuroresearch but they don't have laboratory but they are interested in data processing. An disadvantage of CARMEN portal is that software tools are implemented as web services; it could be obstruction for not advanced users.

Japan portal provides a set of usefull information and news from neuroscience. It contains several links to existing data sources and a set of available software tools therefore could serve as a good guidepost. Although there is possible to add own data source there is not possible to do it automatically (e.g. by filling registration form).

Probably the most promising project is NIF portal where user can find a lot of usefull informations, tools and communities from neuroscience area. The main idea of NIF portal is not to serve as a global database but it enables users to register their own databases. This partial databases are maintained by their owners but data in those databases are available by unified interface (through NIF registry). Their basic users have possibility to only register URL and provide description of own portal, advanced users can register his/her OWL structure. Data in databases registered within NIF portal are searched by full text search engine. Despite all advantages this solution is not addressed to users they don't have own portal where they could share experiments.

Chapter 5

Semantic Web Technologies

5.1 Introduction

Nowadays the World Wide Web (WWW) resp. Internet[1] is the largest knowledge database which is available for human readers over the world. Its boom changed the way of people communication with each other.

Internet was based on 1960s when the US funded military agencies research projects to build robust, fault-tolerant and distributed computer networks. For this purposes they formed a small agency called Advanced Research Projects Agency (ARPA) in order to develop military science and technology. After approximately 10 years still more civilian organizations (e. g.: NASA or Harvard University) were connected into the Internet as well. After several years in the same decade the Internet was expanded into the Europe. Since middle of 1990s the Internet is used for commercial purposes more often. Nowadays it is estimated that quarter of Earth's population uses the services of the Internet.

Nowadays how the Internet is shooting up it consists of a huge amount of information, with practically no classification. It is extremely difficult to effectively handle this enormous amount of information.

Today's Web content is mostly suitable for human readers. It typically

[1] Between meaning of Internet and WWW phrases is a difference. The Internet is a network of all subnetworks over the world against WWW is way of accessing information over the internet. Nevertheless for this work difference between these two terms is irrelevant.

involves people's seeking and making use of information, searching for or getting in touch with other people, reviewing catalogs of on-line stores and ordering products by filling out forms and so on. Main tools used for finding relevant information are search engines such as Google, Yahoo or Alta Vista. Although search engines are widely used though it has several disadvantages:

- They have high re-call and low precision. Even if the main relevant pages are retrieved, relevant or irrelevant documents were also retrieved.
- Low or no recall. Often user does not any relevant result for they request.
- Results are highly sensitive to vocabulary. Offer user's initial keywords don't get the results they want because relevant documents use different terminology from original query.
- Results are single Web pages. If user needs information that is spread over various documents, they have to initiate several queries to collect the relevant documents.

One solution how to solve those disadvantages is to develop increasingly sophisticated techniques based on artificial intelligence and computational linguistics.

An alternative approach is to represent Web content in a form that is more easily machine-processable and to use intelligent techniques to take advantage of these representation. One of these approaches is Semantic Web (extracted from [22]).

5.2 Semantic Web Overview

A semantic web is not a separate web but it is an extension of the current one. The phrase Semantic Web was firstly introduced by inventor of WWW, URIs, HTTP and HTML sir Tim Berners-Lee in [23].

The idea is to enrich web content by semantic metadata that describe content in order to be computer-understandable. Metadata should by expressed by special languages intended to represent data that could be understood by various kinds of software tools (often called software agents). Ontologies and set of statements translating information from various data sources into common terms and rules have to be defined. With that those agents can understand information in those terms. Data formats, ontologies and software agents should operate as one big application on the World Wide Web.

Since a lot of sceptics said that the semantic web was too difficult for people to understand it. It is a little bit true but there are several organizations as a consortium W3C[2] they are working to improve, extend and standardize the system of tools, languages, publications and so on in order to make the semantic web easy to use.

This chapter introduces available languages and tools intended to express information in the semantic web form. Data in the WWW are typically stored in relational databases. Databases are made available in several forms on the Web where users or applications are end-users. In such cases, the semantics of data has to be made available along with the data. For human readers there are appropriate formats (e.g. HTML) but for application programs this semantic has to be provided in a formal and machine processable form.

Data from databases are typically translated from relational model into object oriented model by using object-relational mapping. When we will transform data into the semantic web we can do it in two ways. The first way is from relational model and the second way is from object-oriented model. Each transformation has issues mentioned in this chapter. This chapter also describes what possibilities for describing semantic of engaged data provide mentioned models.

[2] World Wide Web Consortium is the main international standards organization for the World Wide Web founded and headed by Tim Berners-Lee.

5.3 Technologies

The semantic web technologies is a layered architecture, often represented using a diagram first proposed by Tim Berns-Lee. Typical diagram representation is in Figure 5.3.1 on page 35. This schema is quite old (proposed in 1999) but it can still serve as a simple illustration of the semantic web architecture.

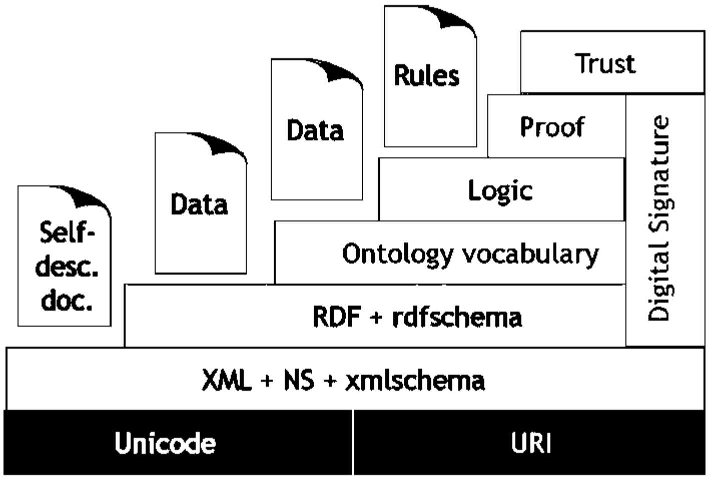

Figure 5.3.1: the semantic web layered architecture [22]

Description of layers is:

- *UNICODE and URI*: Unicode is the standard for computer character representation, URI is the standard for identifying and locating resources.
- *XML*: XML and its related standards, such as Namespaces, and Schema, form a common means for structuring data on the Web but without communicating the meaning of the data.
- *Resource Description Framework*: RDF is the first layer of the semantic web proper. RDF is a simple metadata representation framework, using URIs to identify Web-based resources and a graph model for describing relationships between resources. Several syntactic representations are available, including a standard XML format.

- *RDF Schema*: a simple type modelling language for describing classes of resources and properties between them in the basic RDF model. It provides a simple reasoning framework for inferring types of resources.
- *Ontologies*: a richer language for providing more complex constraints on the types of resources and their properties.
- *Logic and Proof*: an automatic reasoning system provided on top of the ontology structure to make new inferences. Thus, using such a system, a software agent can make deductions as to whether a particular resource satisfies its requirements and vice versa.
- *Trust*: The final layer of the stack addresses issues of trust that the semantic web can support. This component has not progressed far beyond a vision of allowing people to ask questions of the trustworthiness of the information on the Web, in order to provide an assurance of its quality.

In the next section technologies from Figure 5.3.1 on page 35 up to the Ontologies layer are described. Layers above ontologies as well downmost layer are beyond the scope of the work.

5.3.1 Resource Description Framework

Resource Description Framework (RDF) is a standard model for data interchange on the Web. RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this relation is called triples). More formal definition is represents Definition 1.

Definition 1. (RDF triples)

Assume an infinitive set of RDF URI references marked U;
an infinitive set of blank nodes marked B where B = *{b_j: j$\epsilon\mathbb{N}$}* ;
and an infinite set of RDF literals marked L
A triple (v_1, v_2, v_3) $\epsilon(U \cup B) \times U \times (U \cup B \cup L)$ *is called an RDF triple.*

Because this linking structure is directed, labeled graph, from definition 1 could be written definition 2.

Definition 2. (An RDF document is directed labeled graph)

G = (N, E, l_N, l_E)

E (Edges) represent named links between two resources.
N (Nodes) represent resources.
l_N,l_E represent their labels.

Graphical representation of RDF graph looks like an example in Figure 5.3.2 on page 37. Ellipses represent URI-identified resources, rectangles are literals and arcs are URI-identified predicates.

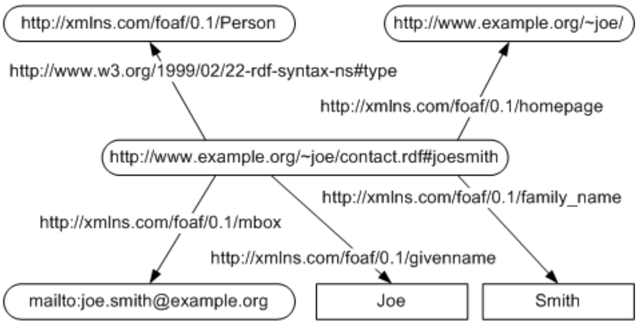
http://xmlns.com/foaf/0.1/Person
http://www.example.org/~joe/
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://xmlns.com/foaf/0.1/homepage
http://www.example.org/~joe/contact.rdf#joesmith
http://xmlns.com/foaf/0.1/family_name
http://xmlns.com/foaf/0.1/mbox
http://xmlns.com/foaf/0.1/givenname
mailto:joe.smith@example.org
Joe
Smith

Figure 5.3.2: Example of RDF graph describing person named Joe Smith [24]

Because XML provides a uniform framework, there are several parsers, structure could be validated according to DTD or XSD[3] scheme, hence XML is a good language for interchange of data/metadata between applications. However XML does not provide any information about semantics of data/metadata.

Although RDF is essentially a data-model there was need to give RDF syntax in order to could be represented and transmitted. RDF model has

[3] DTD or XSD schema languages express set of rules to which an XML must conform in order to be valid.

been given a XML syntax. As a result was RDF/XML language with XML benefits and with possibilities to express RDF triples (Another but not so common and not XML based formats are N3 or Turtle.). The formal grammar for the syntax is annotated with actions generating triples of the RDF graph.

RDF contains several elements and attributes. Basic primitives are: *rdf:Resource, rdf:type, rdf:Description, rdfs:Class, rdfs:SubClassOf, rdfs:Domain, rdfs:Range, rdfs:Literal, rdfs:Property, rdfs:ConstraintResource* and so on. These primitives provide possibilities to describe classes, their data types, restrictions etc.

In addition there are two layers; RDF and RDFS. RDFS describes classes compared to RDF which describes instances of those classes. An example is in Figure 5.3.3 on page 39. The schema contains classes lecture, academic staff member, first-year courses, and properties are taught by, involves, phone, employee id. In the figure properties are blocks, ellipses above the dashed line are classes and ellipses below the dashed line are instances.

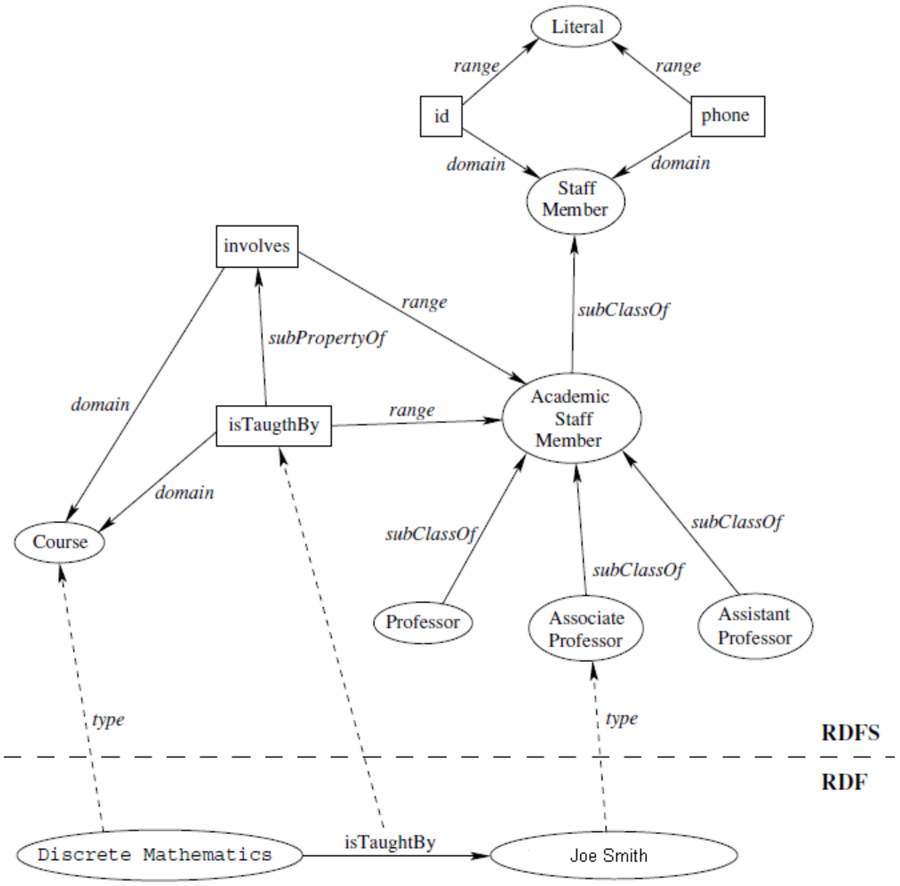

Figure 5.3.3: RDF and RDFS layers [22]

Limitations of the Expressive Power of RDF

RDF contains primitives to concern about classes and subclasses, properties and subproperties, subclasses and subproperties relationships, domain and range restrictions and instance of classes. However a number of features are missing.

In the RDF there are no properties with local range. For instance in the RDF can be defined the range of property using rdf:range for all classes but not only for some classes. For instance if there is property range “eat", there is no possible to define that cows eat only plants, while other animals may eat meat, too.

Next in the RDF there is no possible to define disjointness classes. For example it could not be said that male and female are disjoint. RDF only enables to define that female is subclass of person.

Also in the RDF boolean combination of classes like an union, intersection or complement are not available, so there is no possible to define that class person to is disjoint union of the classes male and female.

RD is not able to solve special characteristics of properties. There is no possible to say that property is transitive (like “greater than”), unique (like “is mother of”), or the inverse of another property (like “eats” and “is eaten by”).

There are many more limitation of RDF, more described in [22]. OWL solves many of them.

5.3.2 Ontology Web Language

The expressivity of RDF is very limited. RDF schema is limited to a subclasses hierarchy and property hierarchy with domain and range definition. Because for the semantics is required much more expressiveness then RDF offers hence W3C has defined a more powerful language named Ontology web language (OWL). An OWL has more facilities for expressing meaning and semantics than RDF. The OWL also facilitates greater machine interpretability of web content than XML and RDF.

There are various syntaxes available for OWL; one of them is RDF/XML. It is only one that is mandatory to be supported by all OWL tools.

Against RDF OWL allows users to write explicit formal conceptualizations of domain models. The main requirements are a well-defined syntax, efficient reasoning support, a formal semantics, sufficient expressive power and convenience of expression.

Well-defined syntax is necessary for machine-processing of information. A formal semantics describes the meaning of knowledge precisely. It means that the semantics does not refer to subjective intuitions, nor it is open to different interpretations by different people or machines.

A formal semantics and reasoning support are usually provided by mapping an ontology language to a known logical formalism, and by using automated reasoners that already exist for those formalisms.

Reasoning support is important because it allows one to check the consistency of the ontology and the knowledge, to check for unintended relationships between classes and automatically classify instances in classes. OWL is a richer vocabulary description language for describing properties and classes, such as relations between classes (e.g., disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g., symmetry), and enumerated classes. Relationships between classes are described by next detonations.

Definition 3. (Class membership)

We will suppose two classes A and B

If we will suppose that there is x an instance of A.

Next we will suppose that B is subclass of A ($B \subset A$)

$\Rightarrow$*x is an instance of B.*

The next definition deals with transitivity of classes.

Definition 4. (Equivalence of classes)

We will suppose three classes A, B and C.

If A is equivalent to B and

B is equivalent to C then

$\Rightarrow$*A is equivalent to C*

The next definition deals with consistency of the ontology

Definition 5. (Consistency)

We will suppose x which is instance of class A and

A is subclass B and C ($A \subset B \cap C$) and A is subclass of D ($A \subset D$),

B and D are disjoint

$\Rightarrow$*Ontology inconsistency because A should B empty, but has the instance x*

$\Rightarrow$*We have indicated an ontology error*

The next definition deals with classification of an individual to class

Definition 6. (Classification)

We have suppose that certain property-value pairs are a sufficient condition for membership in a class A then

if an individual x meet all such conditions

⇒x must be an instance of A

Three Species of OWL

Because the full set of requirements for an ontology language is extensive, W3C defines OWL as three different sublanguages, each geared toward fulfilling different aspects of this full set of requirements.

1. OWL FULL - OWL full uses all the OWL language primitives. It also allows the combination of these primitives with RDF scheme. OWL full is fully compatible with RDF
2. OWL DL - OWL DL is a sublanguage of OWL Full. OWL DL for example restricts how the constructors from OWL and RDF may be used.
3. OWL Lite - OWL Lite contains more restrictions than OWL DL. For example, OWL Lite excludes enumerated classes, disjointness statements, and arbitrary cardinality. The advantage of this ontology is that is easy to use and implement.

Relation between OWL languages is in Figure 5.3.4 on page 42.

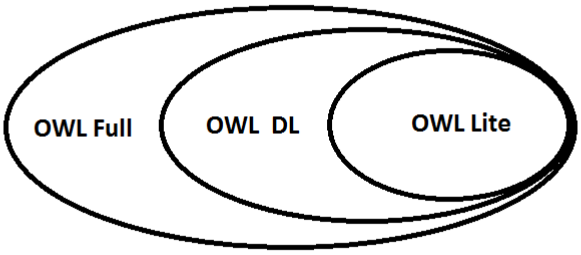

Figure 5.3.4: OWL variants

5.3.3 Relational Model

As was mentioned in the Section 5.2 data in the web are stored in relational databases. So data from those databases have to be transformed into OWL or RDF model. Before describing transformation mechanism is necessary to define a set of relational databases formalisms.

Relational databases are based on relational model. Next five definitions describe constitution of relational model (extracted from [25]).

Definition 7. (Domain)

A domain is non empty set of values with unique name commonly referred to as a data type

Definition 8. (Scheme)

A scheme for Relation R of arity n is a list of unique attribute names A where

$R=\{A_1, \dots A_n\}$

Definition 9. (Relation)

A relation r on scheme R is a subset of the Cartesian product

$R \subseteq A_1 \times \dots \times A_n$

We can say that R has arity n

Definition 10. *(Relational Database)*

A relational database DB is a finite set of relations R_1, R_2, ..., R_n. The schema for R_1, R_2, ..., R_n comprise the database schema for DB.

Definition 11. (Key)

We will suppose key K, relation r and schema R.

A key for relation r in schema R is subset of R such that, for any two tuples in r, they are the same if they have same value for K.

Definition 12. (Attribute)

For a relation R of arity k, each element $X_i (i \leq 1 \leq k)$ of some tuple $t \epsilon R$ can be referenced either by the ordinal value ($X_i = t[i]$), or by some predefined string s_i called an attribute ($x_i = t[i] = t[s_i]$). Because elements can be referenced by attribute value in this way, a relation is often called a table.

5.3.4 Mapping Between Relational and RDF Model

RDF triple can describe a simple fact such a relationship between two things where the predicate names the relationship, and the subject and object denote the two things. A familiar representation of such a fact might be as a row in a table in a relational database. This table has two columns, corresponding to the subject and the object of the RDF triple. The name of the table corresponds to the predicate of the RDF triple. In this table each row represents a unique instance of the subject. Such a row has to be decomposed for representation as RDF triples. Such designed table must be further normalized in order to will be at least in the third normal form [25].

Furthermore in RDB model, every table has a primary key. This key is typically additional column with unique row id, so a form of mapping from a row of a table to RDF triples is presented in [25] as follows .

- The primary key value corresponds to the common subject of collection of triples and the subject has an rdf:type property whose value is the table name.
- The column name of each table corresponds to the predicate of the RDF triple.
- The value in the cell corresponds to the object.
- A more complex fact is expressed in RDF using a conjunction of simple binary relationship.

Algorithm how to get an equivalent RDF model from relational model described in [25] is

- Create an RDF class for each entity-table.
- Convert all primary keys into IRI[4] class.
- Assign a predicate IRI to each non-primary key attribute.
- Assign an rdf:type predicate for each row, linking it to an RDFS class IRI corresponding to the table
- For each column that is neither part of primary or foreign key, construct a triple containing the primary key IRI as the subject, the column IRI as the predicate and the column's value as the object.

The next approach was described in [31]. There is described framework named OntoGrate combines ontology-based schema representation, first order logic inference, and some SQL wrappers. There are defined several mapping rules from the first order logic to relational scheme needed for developing SQL wrappers. There are:

Relation ↔ Type
Attribute ↔ Predicate
Integrity Constraint ↔ Axiom
Primary Key ↔ Fact

By using the set of developed features described in [31, 32, 33] it is possible to express simple ontologies by using first-predicate logic and according to mentioned rules to transform it to relational schema. In addition there is described how it is possible to merge ontologies consisting of common elements from a source and target ontology. Given merged ontology between two sources it is again expressed in the first order logic language. There is also defined data integration model where integration of ontologies is done in two steps.

[4] IRI - Internationalized Resource Identifier is generalization of URI but may contain Unicode characters against URI that can contain only ASCII characters.

- Query Translation: The process of extracting data expressed by one schema to answer a query posed using another schema, also known as query answering.
- Data Translation: Translating data from a source schema to a target (or integrated) schema for the purpose of information exchange.

5.3.5 Mapping Between OWL and OOP

Nowadays object oriented programming (OOP) is the main stream in the software development. There are many profits from using object oriented languages like a code reuse, better structured programs and easier transition from analysis to implementation. It is ensured by class definition, using objects, inheritance or polymorphism. These features ensure a high level of data abstraction. More is desribed in publication [26].

Similarities and Differences Between OOP and Ontology

Semantic web technologies associate three types of features used in the object oriented world. They describe reality in the conceptual level independent to technological restrictions so they are similar to UML representations in OOP. They also constitute database schema for base of facts (RDF). Eventually they are processed by software tools in the implemented application so they are part of the implementation.

At the first sight there are several similarities between OOP (expressed by UML) and OWL. Both they have a classes, an instances or an inheritance. In both it is also possible to define cardinality restrictions etc.

But in detailed view there are many differences. Substantial difference is a meaning of properties and individuals. In in the UML instances and properties are removal from classes, in the OWL properties are double types; object and datatype properties. The first one links an individual to an individual and the second one links individuals to data values. The UML also does not provide support for describing anonymous classes. Further ontologies are static so they don't provide possibilities how to reflect changes in the time while in the UML it is possible by using by state model.

Infrastructure for Developement of Ontologies

Although tools for development ontologies did a progress in last years (from text editors till graphical user interfaces) todays tools still don't provide so user comfort as existing tools for object oriented modeling. One of reasons is that ontology discipline was later formalized but larger problem is the complicated essence of ontological models. Also there are not many tools for serialization ontology into relational databases. Some of existing tools are described in Section 5.4.

Practical Way How to Map OWL to OOP and Back

Because todays majority of software tools are written in OOP languages it is desirable to map ontology languages into OOP languages as well. Since approach of this work is to respect INCF recommendations from 4.1.1 where is recommended to use open source if it is possible hence as a representant of OOP language is chosen Java[5] in the next text.

Java API generated from an ontology can be used to readily build applications (or agents) whose functionality is consistent with the design-stage specifications defined in the schema. Other benefits of this mapping include the use of any Java IDE to debug (or customize) the application or ontology easily and the use of javadoc to generate an on-line documentation of the ontology automatically.

Fundamental differences in understanding OWL and OOP systems are described in [30]. For instance, a class definition in an ontology, which consists of restrictions on a set of properties, implies:

> *An individual which satisfies the property restrictions, belongs to the class.*

However, its equivalent class definition in Java (OO system) containing a set of fields with restrictions on field-values enforced through listener functions in its acceding methods implies:

[5] http://java.sun.com/

> *A declared instance of the class is constrained by the field restrictions enforced through the class acceding methods.*

The above two definitions represent dual views of the same model, and hence they are not semantically equivalent.

In [30] every OWL class is mapped into a Java Interface containing just the acceding method declarations (set/get methods) for properties of that class. Using an interface instead of a Java class to model an OWL class is the key to expressing the multiple inheritance properties of OWL, because Java class language is single inheritance. A corresponding Java class that embeds each interface (corresponding to an OWL class) wherein there are explicitly defined the fields (properties of the class) and implemented the acceding methods is defined. Using interfaces allow to map various set of OWL operators like subClassOf, intersectionOf and oneOf. A summary is shown in table 5.1.

	OWL	Java
Basic Class	A	interface IntA class A implements IntA
Class Axioms	A equivalentClass B	interface IntAB extends IntA, IntB class A/B implements IntAB
	B subClassOf A	interface IntB extends IntA
Class Descriptions	A = intersectionOf(B,C)	interface IntA extends IntB, IntC
	A = complementOf / disjointWith B	interface IntA { IntA ABBlocker()} interface IntB { IntB ABBlocker()} (Overridden blocking method ABBlocker)
	A = oneOf(I1, I2)	Enum A{I1, I2}

Table 5.1: OWL Class Mappings [30]

Situation is more complicated with properties. Properties in OWL assumes multiple-cardinality so Collection type has to be in the Java fields. But in Java each variable can be of one type this contrasts with the permitted multi-range properties in OWL. For avoiding this Java insufficiency in [30] special set of listeners with range checkers is implemented. More accurately description is out of scope of this work.

Back transformation is described in [34] where an OWL processor is developed, SWCLOS3, which is on top of Common Lisp Object System (CLOS). CLOS allows lisp programmers to develop Object-Oriented systems, and SWCLOS allows lisp programmers to construct domain and task ontologies in software application fields.

In SWCLOS a resource node in RDF graph is represented by a CLOS object, and a labeled arc from a node to another is represented by a slot that belongs to an arrow-tail node and has an arrow-head node as slot value, but rdf:type relation is replaced with instance-class relation and rdfs:subClassOf relation is replaced with class-superclass relation in CLOS.

With OWL mapping is situation better because OWL representation is much more likely for objects. Especially, the property restrictions that provide the local constraints on property values for a specific domain may be straightforwardly implemented by CLOS slot definitions that belong to a class.

5.4 Existing Tools and Frameworks

Tools which are considered to generate OWL or RDF from object oriented model or relational database and vice-versa are implemented. This section describes selection of tools which were studied and tested. Selection of tools suitable for future use will be done .

The base of majority tested tools is Framework Jena [35]. It is Java Framework for building the semantic web applications. It provides a program environment for RDF, RDFS and OWL, SPARQL [36] and includes a rule-based inference engine. Jena is open source and grown out of work with the HP Labs Semantic Web Programme. The Jena Framework includes: A RDF API Reading and writing RDF in several RDF formats (RDF/XML, N3 and N-Triples), An OWL, API In-memory and persistent storage SPARQL query engine. Jena is a parser which is able to read/write mentioned formats and store them into internal model. This model could be read by encapsulated frameworks.

SquirelRDF [37] is a tool which allows relational databases to be queried using SPARQL. It is just an implementation of RDB to RDF mapping, thus ontology is not considered.

A very promising approach for mapping from RDB to RDF migration is D2RQ [38]. This framework uses a declarative language to describe mappings between relational database schema and RDF. D2RQ Platform provides possibilities how to query a non-RDF database using the SPARQL query language, how to access information in a non-RDF database using the Jena API or the Sesame API [39], how to access the content of the database as Linked Data over the Web or how to ask SPARQL queries over the SPARQL Protocol against the database. Further D2RQ consists of D2RQ engine, a plug-in for the Jena and Sesame, which uses the mappings to rewrite Jena and Sesame API calls to SQL queries against the database and passes query results up to the higher layers of the frameworks. The last part of D2RQ platform is D2R Server, HTTP server that can be used to provide a Linked Data view, a HTML view for debugging and a SPARQL Protocol endpoint over the database.

Further tool METAMorphoses [40] is data transformation processor from RDB into RDF according to mapping in the template XML document. 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.

Next tool Sommer [41] is a very simple library for mapping Plain Old Java Objects (POJOs) to RDF graphs and back. It uses XML/RDF template in the input. This template is extended about information from input POJOs.

JenaBean is similar tool, it is flexible RDF/OWL API to persist java beans. It takes an unconventional approach to binding that is driven by the java object model rather than an OWL or RDF schema. Jenabean is annotation based and does not place any interface or extension requirements on Java object model. By default JenaBean uses typical JavaBean conventions to derive RDF property URI's, for example, the java bean property "name" would become RDF property ":name". Jenabean allows for explicit binding between an object property and a particular RDF property. So JenaBean against Sommer does not need any input template but generates RDF/XML representation according to JavaBean structure.

Java2OWL-S is tool which is able to generate OWL directly [42]. It uses 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).

There exist several syntaxes for representation of ontologies. The OWL API [43] 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.

Part III

Current Status and Future work

Chapter 6

EEG/ERP Portal

6.1 System Context

Because of hard manual work with large amount of EEG/ERP data and metadata and in face of difficulties mentioned in previous parts (especially no suitable data format, metadata description or suitable software tool for storage and management of experiments), we decided to design and implement own software tool for EEG/ERP data and metadata storage and management.

The developed EEG/ERP data store (called simply the system in the following text) is a prototype which is developed in order to increase both the efficiency and the effectiveness of neuroscientific research. Simultaneously the system is a base tool for research in the semantic web field, so designed data representation in the OWL format will be integrated within the system as well.

6.2 Specification of Requirements

The specification of requirements originated from experience of our laboratory, co-workers from cooperating institutions, books describing principles of EEG/ERP (e. g. [4]) design and data recording and numerous scientific papers describing specific EEG/ERP experiments. It also corresponds to mentioned efforts of INCF in the field of development and standardization of databases in neuroinformatics.

6.3 Project Scope and System Features

System is developed as a standalone product. The database access is available through a web interface. We need a web server supporting open source (Java and XML) technologies and a database system, which is able to process huge EEG/ERP data. The system is easily extensible and can serve as an open source.

The system essentially offers the following set of features (the number 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 crucial user requirement is the possibility to add an additional set of metadata required by a specific EEG/ERP experiment.

6.4 User Roles

The system is developed in order to serve not only locally in our department but it will be open to whole EEG/ERP comunity. One step how to ensure enlargement of the system is to register it in the NIF portal described in 4.2.3. 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.

On the basis of activities that a user can perform within the system the following roles are proposed:

- *Anonymous user* has the basic access to the system (it includes essential information available on the system homepage and the possibility to create his/her account by filling the registration form).

- *Reader* has already his/her account in the system and can list through and download experimental data, metadata and scenarios from the system, if they are made public by their owner. Reader cannot download any personal data or store his/her experiments into database.

- *Experimenter* has the same rights as Reader; in addition he/she can insert his/her own experiments (data and metadata including experimental scenarios) and he/she has the full access to them. This user role cannot be assigned automatically, a user with the role reader has to apply for it and the new role must be accepted by group administrator. Every experimenter is member at least of one group.

- *Group administrator* is user who established a new group or received this privilege from other group administrator. He/She can change privileges of users in group where he/she is administrator.

- *Supervisor* has an extra privilege to administer user groups and change their user roles according to the policy.

6.5 Definition of Metadata

The data obtained from EEG/ERP experiments are senseless if they are not supported by more detailed description of testing subjects, experimental scenarios, laboratory equipment etc. Metadata are also necessary for an interpretation of performed experiment and for data search and manipulation. Metadata are organized in several semantic groups:

- Scenario of experiment (name, length, description, ...)

- Experimenters and testing persons (given name, surname, contact, experiences, handicaps, ...)
- Used hardware (laboratory equipment, type, description, ...)
- Actual surrounding conditions (weather, temperature, ...)
- Description of raw data (format, sampling frequency, ...)

There is important that only a small predefined set of metadata is optional to fill in. In addition, a user with the role experimenter has the right to define his/her own metadata.

Chapter 7

Conclusion

It were written a lot of works describe how to do experiments. Although describing experiments is important to understand how brain works. No less important is to provide suitable neuroscience database where these experiments could be stored. With increasing a number of experiments there is necessary to store data effectively. Nowadays there are not many publications focused on how data should be organized and how to design neuroscience databases. With storing data relevant their metadata description. Metadata should be clearly defined, but common format with description of metadata for ERP domain also does not exist.

This work summarized existing neuroscience databases and formats which attempted to be standard formats mainly for EEG. The majority of introduced format are designed by producers of measuring devices. In order to in incurred chaos was established an order there is INCF organization, published some recommendations how neuroscience databases should be organized and maintained. These recommendations are described in this work as well.

Nowadays making data from databases available through Internet is popular. The reason is clear. Work with web applications is easy for users, they cannot install any software on the computer and they can access data from diverse places. So if data from ERP research are available through the internet then they will be easier manageable and shareable with another researchers.

Although Internet is still more popular and more extensive it is still less well-arranged, hence idea of the semantic web has gone. The goal of the semantic web is to enrich current content about semantic meaning. Semantically annotated data are findable easily, but data transformation from current databases is not without difficulties, so work also describes difficulties and possible solutions how to data from databases transform (represented by relational scheme or object oriented model). A many software tools solves those transformation to a certain degree is described, hence improvement is still necessary.

Because we need a system for storing experiments from our laboratory we have developed portal including database and user web interface. This portal serves for management of our experiments, but also it is a prototype which serves as a base tool for development of a mechanisms that provides data in the semantic web representation. So there is a need to extend this portal with possibility to provide data in the semantic web form. When ERP experiments were described by metadata and those experiment were available in the semantic web representation, it would be possible to register these data sources in the NIF portal and make them available for whole researchers comunity. So developed system could be used as a standardized portal for sharing and managing ERP experiments.

7.1 Aims of PhD Thesis

1. A proposal ontology that represents ERP domain.

2. The ontology will express ERP experiments resulting to a semantic web.

3. Proposal of extension and implementation of a mechanism that transforms the data stored in our database into the semantic web.

4. Registration in the NIF portal that evaluates the design and the practical contribution of the approach.

Bibliography

[1] Elektroencephalography, Encyclopedia Britannica http://www.britannica.com/EBchecked/toic/183075 /electroencephalography, Online, 2010

[2] Electroencephalography/Event Related Potentials (EEG/ERP) Laboratory, Georgetown University, http://brainlang.georgetown.edu/erplab.htm, Online, 2010

[3] G. Johnson, Understanding how to brain works. Traumatic Brain Injury survival guide, http://www.tbiguide.com/howbrainworks.html, Online, 2010

[4] S. J. Luck, An Introduction to the Event-Related Potential Technique (Cognitive Neuroscience), The MIT Press, August 2005.

[5] A. Dean, D. Voss, Design and Analysis of Experiments, Springer Verlag, New York, USA, 1999.

[6] J. Polich, A Kok, Cognitive and biological determinants of P300: an integrative review, Elsevier Science B.V., 1995

[7] European Data Format, http://www.edfplus.info/index.html, Online, 2010

[8] B. Kemp, A. Värri, A. C. Rosa,K. D. Nielsen, J. Gade. A simple format for exchange of digitized polygraphic recordings. Clinical Neurophysiology 1992;82 p. 391–3.

[9] Walter Graphtech, http://www.walter-graphtek.com/, Online, 2010

[10] Natus, http://www.natus.com/index.cfm?page=company_1&crid=139, Online, 2010

[11] Brainlab, http://www.brainlab.be/, Online, 2010

[12] OpenXDF, http://www.openxdf.org/, Online, 2010

[13] Brain Product, http://www.brainproducts.com/, Online, 2010

[14] Weka Machine Learning Project, http://www.cs.waikato.ac.nz/~ml/, Online, 2010

[15] J. van, Pelt, J. van, Horn, Workshop report, 1st INCF Workshop on Sustainability of Neuroscience Databases, Stockholm, 2007

[16] R. Kötter, Neuroscience Databases: A Practical Guide, Kluwer Academic Publishers, USA, 2003, ISBN 1-4020-7165-5

[17] CARMEN Portal, http://www.carmen.org.uk/, Online, 2010

[18] INCF Japan node, http://www.neuroinf.jp/, Online, 2010

[19] Neuroscience Information Framework, http://neuinfo.org/, Online, 2010

[20] NCBO BioPortal, http://bioportal.bioontology.org/ontologies/40510, Online, 2010

[21] R, Mouček, P. Mautner, Driver attention while double stress - EEG/ERP experiment (Pozornost řidiče při dvojí zátěži – EEG/ERP experiment in Czech), Kognice a umelý život IX, Opava 2009, ISBN 978-80-7248-516-1

[22] G. Antoniou, F. van Harmelen, A Semantic Web Primer, The MIT Press, April 2004, ISBN 0-262-01210-3

[23] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American Magazine, May 2001

[24] Semantic Web Example, http://www.obitko.com/tutorials/ontologies-semantic-web/, Online, 2010

[25] Yanhui Lv Ma, Z. M. Transformation of relational model to RDF model. Systems, Man and Cybernetics, 2008

[26] J. de O. Guimarães, The object oriented model and its advantages, ACM SIGPLAN OOPS Messenger, January 1995, Pages: 40 - 49, ISSN:1055-6400

[30] A.Kalyanpur, D. J. Pastor, S. B., J. A. Padget. Automatic Mapping of OWL Ontologies into Java, Software Engeering and Knowledge Engeering, June 2004, Pages. 98-103.

[31] P. LePendu, Ontology based Relational Databases, University of Oregon, 2007

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

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

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

[35] Jena Framework, http://jena.sourceforge.net/, Online, 2010

[36] E. Prud'hommeaux and A. Seaborne, SPARQL Query Language for RDF. W3C Recommendation, http://www.w3.org/TR/2005/WD-rdf-sparql-query- 20050217/, February 2005.

[37] D. Steer, SquirrelRDF, http://jena.sourceforge.net/SquirrelRDF/, Online, 2010

[38] The D2RQ Plattform - Treating Non-RDF Databases as Virtual RDF Graphs, http://www4.wiwiss.fu-berlin.de/bizer/d2rq/, Online 2010

[39] Sesame sematic web toolkit, http://semanticweb.org/wiki/Sesame, Online, 2010

[40] Švihla, M. Transforming Relational Data into Ontology Based RDF data, Thesis, CTU, Prague, 2007

[41] Sommer, https://sommer.dev.java.net/sommer/index.html, Online, 2010

[42] Java2OWL-S, http://www.daml.org/2003/10/java2owl/, Online, 2010

[43] The OWL Api, http://owlapi.sourceforge.net/, Online, 2010