SYSTEM FOR EEG/ERP DATA AND METADATA STORAGE AND MANAGEMENT

*Petr Ježek, Roman Mouček**

Abstract: The purpose of this paper is to introduce a system for EEG/ERP (electroencephalography, event-related potentials) data and metadata storage and processing and to summarize the authors' research in this field. Since researchers have difficulties with a suitable long-term storage and management of electrophysiology data the presented system helps them to increase both efficiency and effectiveness of their work by providing the means for the storage, management, search and sharing of EEG/ERP data. The requirements specification including the system context, system requirements, project scope, basic features, system users, and data formats and metadata structures are presented. The database structure is proposed; upload, download and interchange of EEG/ERP data and metadata using the web interface are described. The system architecture, used technologies and the final realization are described. Data and metadata search over the system and user accounts including system security management are also presented. Additional tools and structures as converters of data formats and semantic web ontology are mentioned.

Key words: *Electroencephalography, event-related potentials, EEG/ERP experiments, relational database, web interface, data search, user accounts management, system security, data format converters*

Received: January 29, 2010
Revised and accepted: June 11, 2012

1. Introduction

EEG and ERP experiments usually take a long time and produce a lot of data. However, there is no usable software tool especially for long-term storage and management of data obtained during these experiments. Experiments carried out in our laboratory were initially recorded and managed using a common folder structure (commercial software tools are used during recordings). Nevertheless, with an increasing number of experiments it was necessary to develop a custom software tool for data maintenance. This EEG/ERP database is described in [1]. Since

*Petr Ježek, Roman Mouček
Department of Computer Science and Engineering, University of West Bohemia in Pilsen, E-mail: `jezekp@kiv.zcu.cz, moucek@kiv.zcu.cz`

then the former EEG/ERP database has incorporated a number of changes and improvements. Currently not only EEG/ERP experiments but also data and metadata structures in various formats, scenario templates of EEG/ERP experiments, groups of people involved in experiments, etc. are covered by the features of our software tool[1]. In the paper the basic system context, users' requirements, and data formats and metadata structures are singled out. Then, new or updated features of the system are described.

2. EEG/ERP Data Storage And Management

There are many difficulties with long-term storage and management of EEG/ERP data and metadata.

- There is no software tool for easy EEG/ERP data management, which is accessible to the research community.
- There is no EEG/ERP data format generally accepted or used by the community.
- Data and metadata are often considered as unimportant for long-term storage and management once their interpretation is completed.
- Metadata are usually not organized.
- There is no practice to share experimental data between various labs.

As a result, we have incorporated a set of frequently used data formats and proposed a metadata structure within the development of the EEG/ERP system.

3. Specification and Requirements

3.1 System context

With regard to previously mentioned difficulties with data formats and metadata structures and also with regard to intensive manual work to manage large sets of EEG/ERP data we decided to develop a custom software solution, which provides tools for storage and management of data and metadata obtained during EEG/ERP experiments.

The developed software solution pursues not only our local research but it generally contributes to advancements in human brain understanding. In addition, we believe that such an advanced software tool increases both the efficiency and the effectiveness of neuroscientific research.

The system is not intended for direct usage of neural networks but it can be very useful for better understanding of neural networks activity presented e.g. in [2, 3].

[1] The software tool is called simply the system in the following text.

3.2 System requirements

The requirements specification originated primarily from experience gained during experiments performed in our laboratory, from working experience of our collaborative partners, from books describing principles of EEG/ERP design and data recording [4], and from a numerous set of scientific papers describing specific EEG/ERP experiments. It also corresponds to the effort of the International Neuroinformatics Coordinating Facility (INCF)[2] [5] in the field of development and standardization of data formats and metadata structures in neuroinformatics. The system has to be useable for a wide research community. Moreover, it has to be easy to use without special needs on user hardware and software equipment. The system requires only standard computer skills.

3.2.1 Project scope

The main goal of this project is to enable clinicians and groups of interested researchers to store, update, manage and download raw data and corresponding metadata resulting from EEG/ERP experiments. This goal also includes the possibility to set and manage a personal constitution of their experiments and research groups.

The system is developed as a standalone server side application. Open source technologies (based on the Java platform) and standardized data structures (XML, relational model, OWL) are used. A web server supporting open source technologies and a relational database system, which is able to process large EEG/ERP data, is necessary. The architectural style based on separated layers and supported by used technologies ensures easy and user-friendly future management and extensibility of the system. The system also provides the possibility to download data in a form suitable for subsequent machine processing.

3.2.2 System features

The system includes the following set of features:

- Registration of new users
- User authentication and authorization
- Storage, download, update and management of EEG/ERP data and metadata (recordings, scenario templates, and testing people)
- Download of EEG/ERP data and metadata, which are provided by other system users (excluding personal data of testing subjects)
- Identification and management of users groups
- Download of metadata in the XML format and the OWL structure

The set of features available for a specific user is limited by the policy of user roles.

[2]We are the members of Czech National Node for Neuroinformatics.

3.2.3 System users

The system was primarily developed for researchers in our department and for our close collaborative partners. However, since we cooperate with several other institutions we decided to open it to a larger community, especially to those who are interested in the research of brain activity, especially in the field of EEG/ERP experiments. The system is now widely tested to guarantee safety of personal data and availability of EEG/ERP resources. Finally, the system will be usable for both clinicians and researchers.

3.2.4 User roles

Since the system is going to be open to the whole EEG/ERP community of researchers it is necessary to protect EEG/ERP data and metadata and especially personal data of testing subjects stored in the database from an unauthorized access. Therefore, the restricted user policy is applied and user roles are introduced. The user policy has changed since the last description published in [1]. A new role 'group administrator' has been introduced and the privileges of the user roles 'reader', 'experimenter' and 'supervisor' have been changed.

The following roles are proposed on the basis of activities, which the user can perform within the system:

- **Anonymous user** – has not created an account yet. He/she has access only to basic information resources. This includes information available in the system homepage and the possibility to register into the system by filling a registration form.
- **Reader** – has already created an account in the system. He/she can go through, read and download data and metadata of EEG/ERP experiments stored in the system by the members of his/her group. Reader cannot read or download any private data and store his/her experiments into the system.
- **Experimenter** – has already created an account in the system. He/she is simultaneously the group administrator or he/she is qualified as the experimenter by his/her group administrator. He/she has the same basic set of privileges as the reader. In addition, experimenter can add his/her own experimental data and metadata into the system and he/she has full access to them.
- **Group administrator** – is a registered user who has created a new group. This user has the same privileges as the experimenter; in addition he/she can change user roles of group members (reader, experimenter) or accept new members of his/her group.
- **Supervisor** – has the same privileges as the group administrator, in addition he/she administrates the whole system and checks if system users follow the system rules.

By introducing the system of research groups a self-management of each group by its group administrator is possible. This solution enables the system supervisor to concentrate only on the essential issues connected with the system operation and maintenance.

3.2.5 System sustainability

The purpose of the system is not to serve only as a local managing tool for our research but to serve as a system, which enables sharing and interchange of data/metadata between various research groups. Currently EEG/ERP data are provided by diverse groups of not only medical communities but scientists or universities as well. The system is, therefore, developed as an open source accepting INCF recommendations. It is offered as a free managing tool and a source of EEG/ERP data within a collection of other neuroinformatics data sources.

3.2.6 System security

The system database contains personal data, which are necessary either for an interpretation of EEG/ERP experiment or for a future contact with a testing subject. Only the Experimenter has access to personal data of the testing people taking part in his/her experiment. Personal data collection and storage are managed according to law. The database is installed on the proprietary server inaccessible to third parties.

3.2.7 System performance

Since the system is a server side application, performance of an application server has to enable a trouble-free application run for hundreds of users working at the same time. The system database is controlled by RDBMS and works with huge data sets. Performance tests are widely used during the system validation. Performance of a user computer is not critical; only quality of user Internet connection can be a restricting factor.

4. EEG/ERP Data Formats

A variety of data formats are used to store data obtained during EEG/ERP experiments. More spread formats include European Data Format (EDF or EDF+) [6], Attribute-Relation File Format (ARFF) [7], Vision Data Exchange Format (VDEF) [8] and KIV[3] [9] format. VDEF and KIV formats are widely used in our laboratory, and the last one has been introduced directly in our department.

Data and metadata in European Data format (EDF) are organized in one file. EDF data file consists of a header record followed by EEG data records. The header content has a variable length, identifies a testing subject and specifies technical characteristics of recorded signals. The data part contains consecutive fixed-duration epochs. EDF+ is an extension of EDF. In addition, it can contain interrupted recordings, or store any medical recording such as EMG, ECG or Evoked Potentials. Some omissions as a Y2K problem, little-endian integers, etc. were fixed in EDF+ format. Despite some disadvantages (especially storage of raw data and

[3]KIV - Czech abbreviation of our department

metadata in one file), it was probably the most promising attempt to standardize EEG data format.

Data and metadata in Attribute-Relation File Format (ARFF) are organized in ASCII text file. ARFF file contains two sections; a header section providing a limited set of metadata is followed by a data section. ARFF format is used in WEKA software [10]. This software includes a set of tools and algorithms for data mining and machine learning; some of them are used for analysis of experimental results in our laboratory.

Vision Data Exchange Format (VDEF) is used by the commercial technical equipment used in our laboratory. The EEG record consists of three files: a header file, a marker file and a raw data file. The header file (based on the Windows INI format) contains information about recorded data and a reduced set of metadata in the form of attribute-value pairs. The marker file is used for storage of information about markers (their type and timing) in the EEG record. The raw data file contains EEG record itself.

KIV data format is a modification of simple ASCII format. Metadata are stored in XML file and raw data are stored in binary files.

5. EEG/ERP Metadata

The data resulting from EEG/ERP experiments have to be completed by more detailed information about experimental scenarios, laboratory equipment, weather condition, etc. These metadata are necessary for both actual and future interpretation of raw data and data search and manipulation. We have designed several sets of metadata; each set represents a semantic group. Not all metadata values are obligatory to be filled in. In addition, a user with the role of Experimenter has the privilege to define his/her metadata.

The following semantic metadata groups were defined:

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

6. System Design and Implementation

6.1 Architecture

The system uses a three-layer architecture (persistent, application and presentation layer) and MVC pattern[4]. This architectonic style ensures a high level of abstraction. It is easy to manage and is also directly supported by the used programming

[4]MVC (Model View Controller) is software architecture that separates domain modeling from its presentation and from actions based on a user input.

tools and technologies (open source Java technologies, XML). The architecture and the used technologies ensure the sustainability of the system and the efficient future system maintenance.

6.1.1 Persistent layer

The persistent layer uses POJO[5] objects as an interface between the application layer and tables and relations in the relational database. The Hibernate framework is used for object-relational mapping. ERA model (available in Fig. 1) includes tables for storing user accounts, data and metadata relating to EEG/ERP experiments, experimental scenarios, and testing subjects. Most attributes are obligatory. The database scheme also contains additional tables for possible metadata extensions (these tables are omitted in Fig. 1 to keep the model understandable). This approach ensures a future extension of metadata sets without the necessity to change the basic database scheme.

6.1.2 Application layer

The application layer processes all requests in the application. This layer is implemented using the Spring framework (with other included modules, e.g. Spring MVC or Spring Security), Aspect Oriented Programming and Hibernate framework support. Spring MVC directly separates application into several modules.

The application layer contains a set of controllers, which process requests from the user and calls methods from data access objects (these objects synchronize the content of dynamic objects with the content of the relational database). The validation of user's input data is implemented by form validators.

Authentication and user roles management are ensured by the Spring Security module. This module provides tools for interaction between the user and the system including e-mail communication.

6.1.3 Presentation layer

The system user interface is web-based and is divided into several parts (Fig. 2). Each page consists of the header, the main menu, a content part and the footer. The system generates HTML pages using JSP technology (including JSTL); design is defined by CSS. Access privileges to individual pages (authorization) are secured by the Spring Security module as well. Users need only a supported web browser to work with the system successfully.

Logged-in users get basic information about their experiments, inserted data, scenarios, and group memberships in their homepage. They can get details of each category easily by clicking on the Detail link. The number and the form of main menu items available for the user depend on user's privileges.

[5] POJO means "Old Plain Java Object"; it has the same meaning as Java Bean or Transactional object.

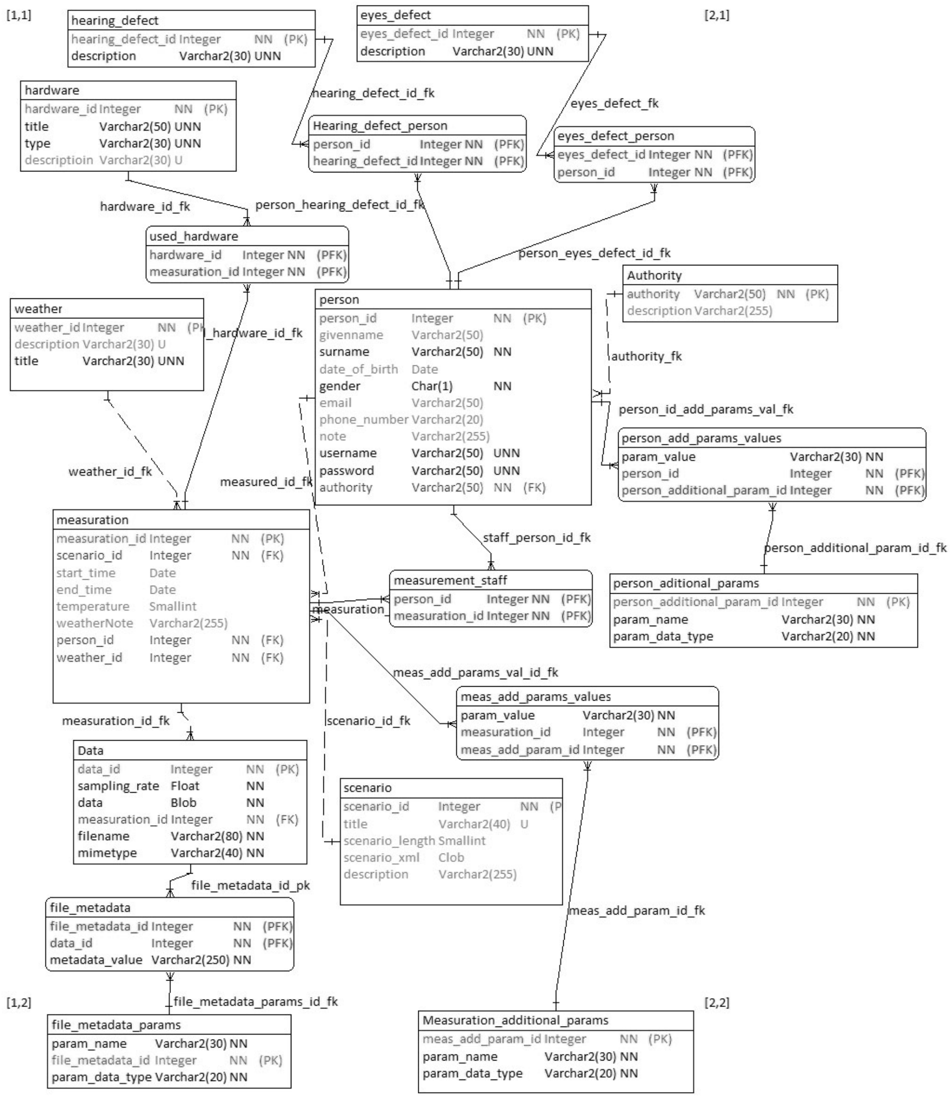


Fig. 1 *Database schema – ERA model.*

6.2 Application modules

During our latest work we have added other modules into the system. These modules facilitate data search, data download, and management of users and their accounts.

6.2.1 Search engine

Because of large records stored in the system database it is necessary to help a user find required data in an easy way. We suppose that system users are familiarized

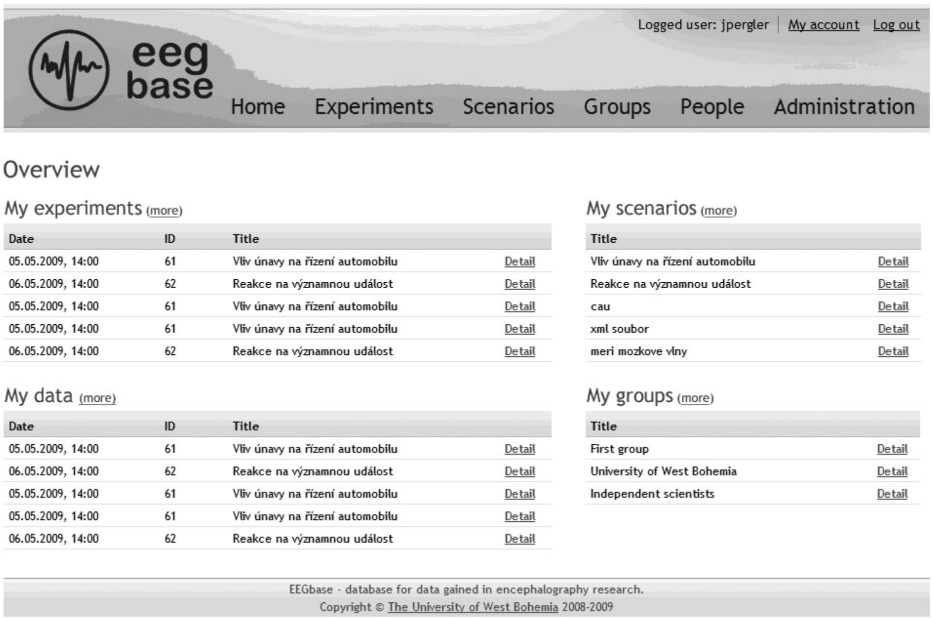


Fig. 2 *Application preview.*

with wide spread search engines (e.g. Google). It means that they type one or more keywords and expect that the system provides a set of relevant results.

We implemented this functionality by using the Hibernate Search Engine [11]. This framework uses Apache Lucene [12] internally. Apache Lucene is a full text search over text documents. Hibernate as an ORM framework provides searching not only over text documents but also over a domain model. Therefore, we are able to provide a full text search over data/metadata stored in the system relational database.

The next data search possibility is to use a specialized form in each web section (experiments, scenarios...). Then a user can restrict data/metadata search by filling restrictive conditions in the appropriate form. The results relevant to the selected section are returned.

The system also ensures a well-arranged output in each section; the user can sort the output by a selected property. Since a set of results can be extensive, the output pagination is implemented.

6.2.2 User accounts manager

To ensure data consistency and personal data privacy a module responsible for a management of user accounts, user groups and access privileges to individual web pages has to be designed and implemented. The Spring-Security module as a part of the Spring framework is used for this management. This framework is directly intended for web applications; a configurable XML file, in which the administrator can define access rules for each part of the web, is used.

First of all, the administrator defines a login page, logout page, home page, which is used if user is successfully logged in, and a page which is displayed in the case of unsuccessful authentication (the following example).

```
<form-login
        login-page=“/login.html”
        default-target-url=“/home.html”
        authentication-failure-url=“/login.html?login_error=1” />

<logout
        logout-url=“/logout”
        logout-success-url=“/home.html” />
```

In the next step the administrator defines an URL pattern and user roles with access to the pages corresponding to the URL pattern. There are two special roles: IS_AUTHENTICATED_ANONYMOUSLY and IS_AUTHENTICATED_FULLY. The first role is used for the user who is not logged in and the second role is used for a logged-in user. The following example represents definitions of URL patterns and special roles.

```
<intercept-url
        pattern=“/home.html”
        access=“IS_AUTHENTICATED_ANONYMOUSLY” />

<intercept-url
        pattern=“/my-account/*.html”
        access=“IS_AUTHENTICATED_FULLY” />
```

Naturally we need a more accurate role definitions. It is possible to define an arbitrary role as done in the following example.

```
<intercept-url
        pattern=“/administration/*.html”
        access=“ROLE_ADMIN” />
```

The set of user roles is defined in the database. An authentication provider defines SQL queries which obtain the user name, password, and user authority from the data source (available in the following example).

```
<authentication-provider>
        <jdbc-user-service
                data-source-ref=“dataSource”
                users-by-username-query=“select username, password, 1 AS
                enabled FROM person WHERE username=?”
                authorities-by-username-query=“select username, authority
                FROM person WHERE username=?” />
        <password-encoder hash=“md5” />
< /authentication-provider>
```

Restrictions to URLs are defined; it ensures that the user cannot access pages that are prohibited for his/her role. It also means that the user cannot access prohibited pages by entering URL into the address input field in the web browser.

However, we often need to hide only a part of the web page which is prohibited for enumerated user roles. It is useful e.g. in the case when we need to hide a link to another page which is prohibited for some user role (“security” element is used).

The following example illustrates the link to the experiments page. This page is visible to the standard user and administrator roles. The second link to the administration page is visible only for the administrator. We can see that security elements can be nested.

```
<security:authorizeifAnyGranted=“ROLE_USER,ROLE_ADMIN”>
        <a href=“<c:url value=’/experiments ’/>” experiments</a>
        <security:authorizeifAllGranted=“ROLE_ADMIN”>
             <a href=“<c:url value=’/administration’/>Administration</a>
        < /security:authorize>
< /security:authorize>
```

The security solution is more complicated in the system because users are separated into user groups and each group has its own administrator. Additionally, the role of supervisor over all groups is defined. If a user wants more privileges than he/she currently has they have to contact their group administrator who can change their privileges. The form enabling change of user access privileges is shown in Fig. 3.

The system stores passwords encoded by the SHA-1 algorithm with randomly generated salt. This indecipherable password provides a high level of security.

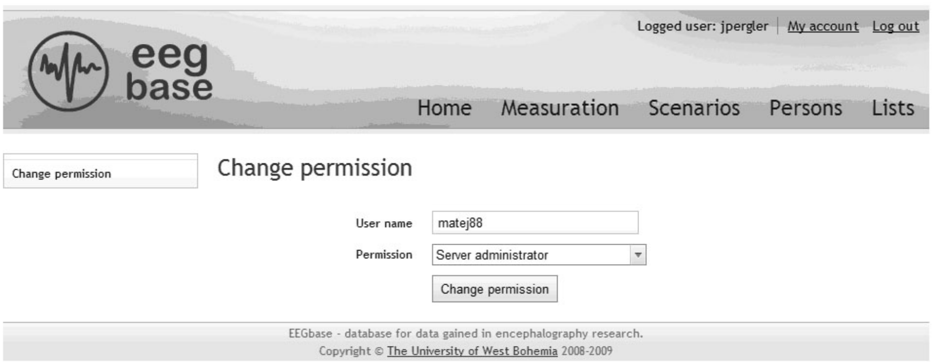


Fig. 3 *Change permission form.*

6.2.3 Data download manager

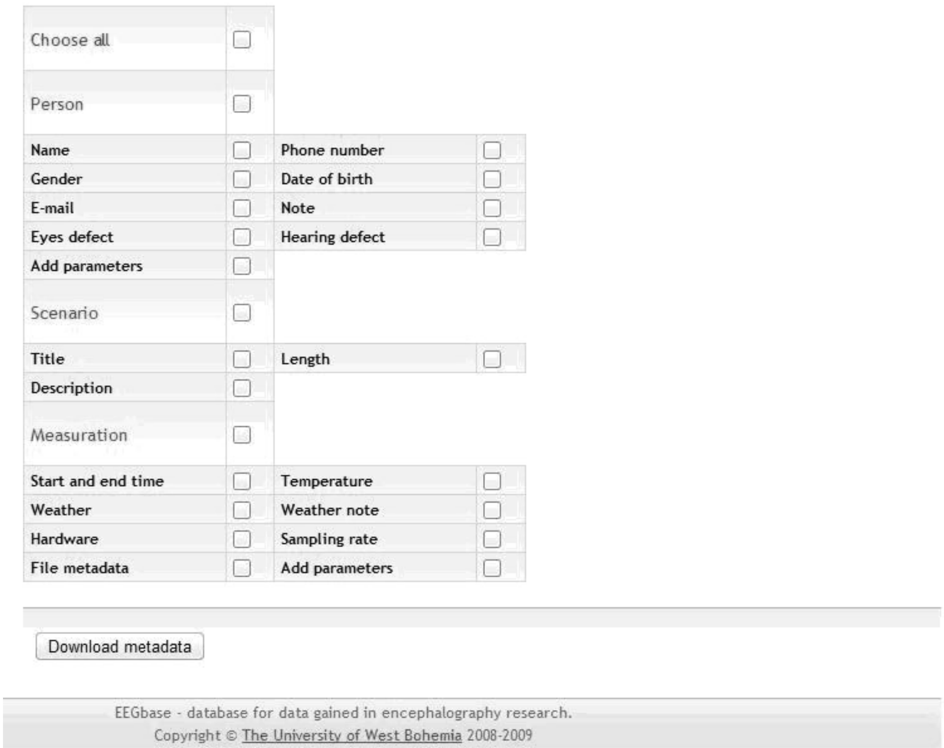
When the user searches for data/metadata stored in the system he/she usually wants to download them for a future work. The advanced download manager is implemented for downloading data/metadata in several forms.

- User readable format – HTML and XML format
- Format for machine processing – particularly XML format and RDF graph for semantic web representation

As mentioned above, the system provides information about experiments through the web interface. When the user wants to download an experiment in the XML format for a future processing, he/she can use a well-formed document structure for data and metadata selection.

The user chooses the experiment which he/she wants to download and clicks on "Download experiment" button. Fig. 3 represents the form where user can choose which metadata will be downloaded. Then the system builds the XML output. This XML output is packed together with binary data files into one output ZIP archive and is finally provided to the user.

Since sensitive personal data are stored in the system, users can download only personal data of the experiments they stored. In case of public experiments users are allowed to download data without personal metadata.



Choose all	☐		
Person	☐		
Name	☐	Phone number	☐
Gender	☐	Date of birth	☐
E-mail	☐	Note	☐
Eyes defect	☐	Hearing defect	☐
Add parameters	☐		
Scenario	☐		
Title	☐	Length	☐
Description	☐		
Measuration	☐		
Start and end time	☐	Temperature	☐
Weather	☐	Weather note	☐
Hardware	☐	Sampling rate	☐
File metadata	☐	Add parameters	☐

Download metadata

EEGbase - database for data gained in encephalography research.
Copyright © The University of West Bohemia 2008-2009

Fig. 4 *Metadata selection form.*

JAXB [13] technology was used to prepare this well-formed XML document. This technology generates the XML output directly from data objects and vice-versa. In addition, it is possible to generate data objects according to attached the

XSD scheme. It guarantees that the XML output is well-formed according to W3C standards.

6.3 Conversion between existing data formats

Mentioned above in Section 4 were some existing data formats. Converters between these data formats were designed and implemented. These converters can be downloaded and used locally to already downloaded data; no conversion is performed during data upload/download.

6.4 Semantic web technologies

Registration of the system as a recognized data source occasionally requires providing data and metadata structures in the form of ontology in accordance with ideas of semantic web. We also have worked on the representation of data and metadata structures using semantic web technologies. Currently, it is possible to generate and provide data and metadata structures using Ontology Web Language (OWL). The details will be presented in a separate paper.

7. Conclusion

We presented a system for long-term storage of EEG/ERP records including raw experimental data, corresponding metadata, experimental scenarios, testing people, etc. The system significantly promotes community work, it serves the community in electrophysiology to store, manage and share data and metadata from EEG/ERP experiments.

We designed and implemented the web interface to enable manage data and metadata. The system includes user accounts to ensure data availability in large scientific community respecting the privacy of tested subjects. We also designed and implemented a set of converters for various data formats and data search over the system.

The system is installed on the developmental server and testing process is running. The system also runs on the production server[6] and is available for our collaborators and people interested in neuroscience and neuroinformatics. Currently, more than 100 experiments are stored and more than 130 active users are registered. If these users accept our software solution we will provide the system to a wider research community. We also believe that the system will be useful also for researchers who cannot produce their own data but are interested e.g. in data processing or data mining.

Currently we work on the system which will be able to produce data and metadata in the semantic web form (RDF, OWL). We tested a set of frameworks generating ontology from the relational database and the object oriented model. The produced ontology enables us to register the system as a recognized source of EEG/ERP data and metadata within international institutions, e.g. NIF [14]. We are also going to prepare a set of advanced software tools including a set of methods for data processing and tools for design of EEG/ERP experiments.

[6] http://eegdatabase.kiv.zcu.cz/home.html

Acknowledgement

The work was supported by the Ministry of Education, Youth and Sport of the Czech Republic under the grant ME 949.

References

[1] Ježek P., Mouček R.: Database of EEG/ERP experiments. In: Proc. of International Conference on Health Informatics HEALTHINF 2010, INSTICC, Valencia, Spain, 2010, pp. 222-227.

[2] Faber J., Pěkný J., Piekník R., Tichý T., Bouchner P., et al.: Simultaneous recording of electric and metabolic brain activity. Neural Network World, 2010, **20**, 4, pp. 539-557. ISSN 1210-0552.

[3] Bouchner P., Novotný S., Jiřina M.: Identification of driver's drowsiness using driving information and EEG. Neural Network World. 2010, **20**, 6, pp. 773-791. ISSN 1210-0552.

[4] Luck, S. J.: An Introduction to the Event-Related Potential technique. Cambridge, MIT Press, 2005.

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

[6] European Data Format. [Online] 7, 2009. http://www.edfplus.info/.

[7] Attribute-Relation File Format. [Online] 7, 2009. http://www.cs.waikato.ac.nz/~ml/weka/arff.html.

[8] Vision Data Exchange Format. [Online] 7, 2009. http://biomag.uni-muenster.de/EEGlab/docs/VisionRecorderManual.doc.

[9] Kučera J.: Software for ERP processing, data reading and presentation (Software pro zpracování ERP, načítání a zobrazení dat). Pilsen, Thesis, University of West Bohemia, Pilsen, 2009.

[10] WEKA software. Weka 3 - Data Mining with Open Source Machine Learning Software in Java.[Online] 7, 2009. http://www.cs.waikato.ac.nz/~ml/weka/.

[11] Hibernate Search. [Online] 1, 2010. https://www.hibernate.org/410.html.

[12] Apache Lucene. [Online] 2010. http://lucene.apache.org/.

[13] Java Architecture for XML Binding (JAXB). [Online] 1, 2010. http://java.sun.com/developer/technicalArticles/WebServices/jaxb/.

[14] Neuroscience Information Framework (NIF). 1, 2010. http://www.neuinfo.org/.