Electroencephalography Data Processor
Framework for Running Signal Processing Methods

Petr Ježek[1] and Roman Mouček[2]
[1]*New Technologies for the Information Society, University of West Bohemia, Univerzitni 8, 306 14, Pilsen, Czech Republic*
[2]*Department of Computer Science and Engineering, University of West Bohemia, Univerzitni 8, 306 14, Pilsen, Czech Republic*

Keywords: EEG/ERP, Signal Processing, Data Processing, Plug-in, Wavelet Transform, Matching Pursuit, Fast Independent Component Analysis, Finite Input Response, Fast Fourier Transform, Workflow, Software as Service.

Abstract: This paper introduces difficulties related to running of signal processing methods. Although several systems that implement signal processing methods exist, their sharing and remote calling is not satisfactorily solved. Authors present a custom server-side approach that provides a powerful plug-in engine for integration of signal processing methods. The plug-in engine ensures high modularity and flexibility of the system. Since the implemented methods are accessible via the SOAP Web Service, integration with another system is available. There is also possible to use the system locally via a web browser. The set of basic methods is already implemented and presented. The architecture and the most important parts of the system are also presented.

1 INTRODUCTION

In our research group we specialize on research of brain activity. During our experiments we largely use the methods of Electroencephalography (EEG) with its subset Event-Related Potentials (ERP). Experiments are performed in a neurophysiological laboratory including recording devices, a car simulator or a soundproof booth. When experiments are performed experimental data/metadata are collected for future processing. Since neuroscience community is facing problems with the long-term storage of data/metadata, raw data analysis, data/metadata sharing or design of experimental protocols International Neuroinformatics Coordinating Facility (INCF)[1] released recommendations (Van Pelt and van Horn, 2007) for handling of the neurophysiological data.

The Czech INCF National Node cooperates on the definition of a standardized data/metadata format for electrophysiology research. As an initial step we developed a custom solution called the EEG/ERP Portal (Jezek and Moucek, 2010).

Data are stored within the EEG/ERP Portal and they are accessible through a web browser. In addition, we provide data in the Semantic Web form available for automatic readers. The Semantic Framework (Jezek and Moucek, 2012) that we developed transforms data from the EEG/ERP Portal into the Semantic Web languages. The EEG/ERP Portal is registered within the Neuroscience Information Framework (NIF) (Gardner et al., 2008); it ensures easier searchability of stored experiments. The EEG/ERP Portal is also integrated with other developed systems (e. g. JERPA (Jezek and Moucek, 2011)) via Web Service API.

Because data from stored experiments are usually further processed using various signal processing methods, we are presenting the EEG Data Processor that enables running of signal processing methods remotely.

[1]http://www.incf.org/

2 STATE OF THE ART

When data are collected they are supposed to be further processed. When third-party signal processing methods are to be used, data need to be downloaded and processed locally. The obtained results are not stored as well.

When we were looking for a suitable tool, we found that the set of available frameworks limited. The following existing tools are able to process EEG/ERP experimental data.

The Carmen Portal (Watson et al., 2007) provides a set of implemented analytical tools. A user can up-

Ježek P. and Mouček R..
Electroencephalography Data Processor - Framework for Running Signal Processing Methods.
DOI: 10.5220/0004246303570361
In *Proceedings of the International Conference on Health Informatics* (HEALTHINF-2013), pages 357-361
ISBN: 978-989-8565-37-2

load custom data into a local storage. Data can be analyzed and the results are putted into a virtual directory assigned to users' profiles. When the user wants to add a custom analytical tool he/she can upload it into the system. When this method is approved by Carmen administrators they integrate it into the system.

Carmen is only one well-known system that enables calling analytical methods remotely together with the possibility to store experimental results.

The next approaches work locally without any possibilities to save or share experimental results.

Modular Toolkit for Data Processing (MDP) (Zito et al., 2008) is a data processing framework written in Python. MDP is a modular framework that Python programmers can extend by additional modules. Common users can call implemented modules locally.

BioSig (Schlogl and Brunner, 2008) is a free open source software library of biomedical signal-processing tools implemented as a library for Matlab, Octave and C/C++. The last version of the tool called biosic4c++ can be also used in the Python. This system contains additional submodules e.g. for real-time data processing or a signal viewer.

Since data processing often requires usage of more methods sequentially, development of specific workflows is required.

Taverna (Hull et al., 2006) is an open source and domain-independent Workflow Management System developed by ^{my}*Grid*[2] team. It is a suite of tools used to design and execute scientific workflows and aid in silico experimentation.

E-Science Central (Watson et al., 2010) combines three emerging technologies Software as a Service, Social Networking and Cloud Computing.

The Carmen developers are also developing CARMEN Workflow Tool. It is a Java-based approach designed to support both data and control flow. It allows parallel execution of services using a Service Invocation API.

3 EEG DATA PROCESSOR

3.1 Existing Approaches Difficulties

Although described solutions work quite satisfactorily, several difficulties stay. The most of tested tools work as stand-alone systems that a user has to install on a custom computer and execute manually. The tools are implemented using various programming languages where each requires a specific runtime environment. It could be an obstacle for many users because it requires a certain level of user knowledge. The last disadvantages are inability to integrate such systems with users solutions and controlling them from a custom client program.

[2]http://www.mygrid.org.uk/

3.2 System Scope

Because of difficulties mentioned in Subsection 3.1 we have decided to implement a custom system that enables running of signal processing methods on data stored within the EEG/EEP Portal. The system is prepared to be integrated with the systems of other users or to be used independently.

The implemented methods are plug-ins based on the Java platform. The libraries and the whole system are provided as an open source. It ensures a higher availability for potential users and faster development of new plug-ins. Basically, the system is a lightweight wrapper for implemented plug-ins with a double type user interface: The first is a standardized Web Service API, the second one is an interactive web interface.

The focus of the EEG Data Processor is to be a software as a service (SaaS) delivery model. It offers reuse of software applications, ability to compare processing algorithms and access to considerable computing resource for high intensity computing tasks.

4 DESIGN AND REALIZATION

4.1 Overall Architecture

The system is a layered architecture. This architectural style is supported by used programming languages and technologies (Java, Maven, Spring, Hibernate, Apache CXF web services etc.)

The internal structure consists of several components. Figure 1 shows the main components of the system together with EEG base (Jezek and Moucek, 2010) integration:

- EEG Binary Loading - it loads data from binary files obtained from an analogue-digital converter. We currently support data obtained from the Brain Vision Recorder but the system is fully prepared for adding support of a new data format.
- Processing Resource Pool - Since performance capacity of the hosted server is limited, we have implemented a pool of available resources. The system can be configured to manage a number of requests simultaneously. When this limit is reached, other requirements are queued and gradually processed.

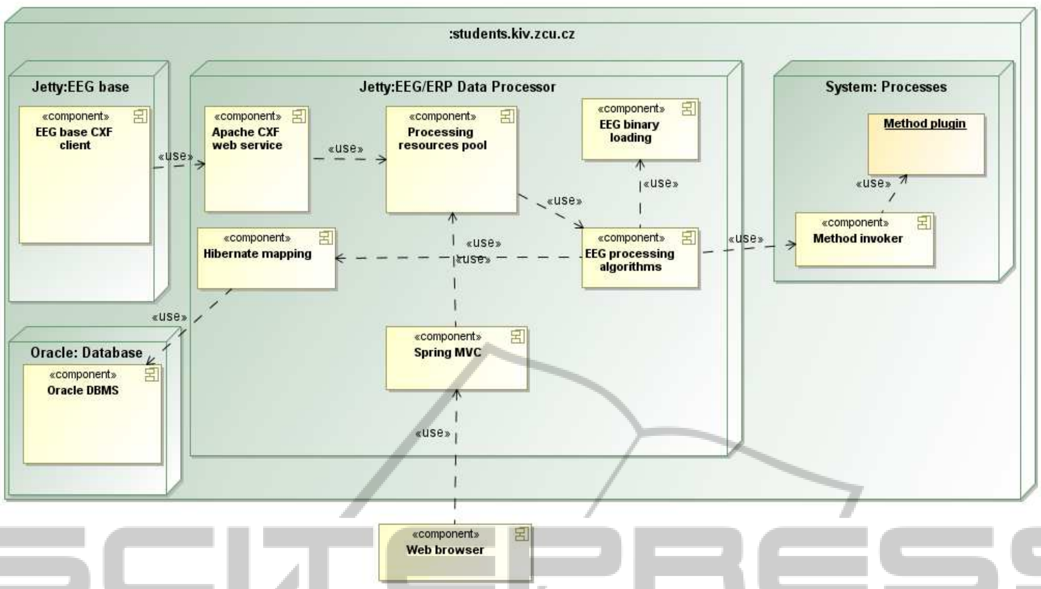

:students.kiv.zcu.cz
Jetty:EEG base
«component» EEG base CXF client
Jetty:EEG/ERP Data Processor
«component» Apache CXF web service
«component» Processing resources pool
«component» EEG binary loading
«component» Hibernate mapping
«component» EEG processing algorithms
«component» Spring MVC
System: Processes
Method plugin
«component» Method invoker
Oracle: Database
«component» Oracle DBMS
«component» Web browser
«use»

Figure 1: Component Model.

- EEG Processing Algorithms - This module manages a running of installed plug-ins. It has access to the list of installed plug-ins and calls a method invoker.
- External Method Invoker - Is responsible for execution of a requested method. It parses the method parameters and takes the method result.

4.2 User Interaction

4.2.1 Web Service Connector

Since the EEG Data Processor is supposed to be accessed by external clients' systems, the Web Service API best matches this requirement. The system provides a simple interface with several basic methods. The client can obtain a list of installed methods, lists of required parameters for the specific method, get available processing units and call the method that processes input data. Such interface provides a sufficient list of operations in order to be integrated with any client system. A usage of SOAP web services ensures easy integration by a well-defined protocol.

4.2.2 Web Interface

Communication with a client system is ensured using Web Service Connector. If the user has only data but not a custom system connected to the Internet, a simple web based interface is prepared. Figure 2 shows the interface overview. The user can list installed signal processing methods and available processing units. When he/she wants to upload custom data, a simple upload form is prepared. The uploaded data are stored within the user's profile and processed on the background. When the result is available, a link to download appears.

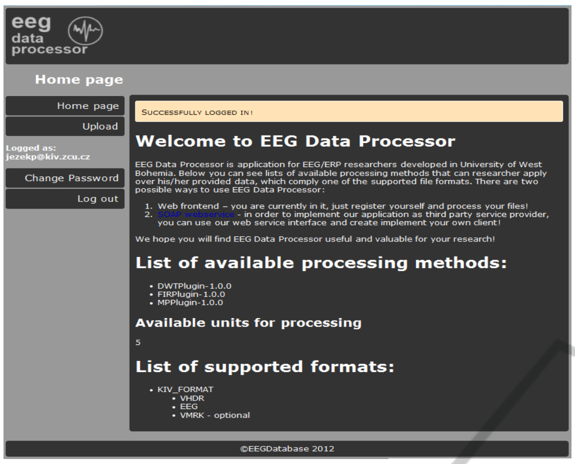
4.3 Customization Part

4.3.1 Plug-in Engine

A Plug-in engine is a central unit of the system. It contain an implementation of a Processor. The Processor is an abstraction of implemented EEG Binary Loading libraries that are intended to read input digitalized signal stored using various data formats. The Processor creates individual Processes. The Process is an encapsulation of a specific method together with the specific parameters needed to start the method.

4.3.2 Plug-in Requirements

When a new method is developing it has to satisfy several conditions in order to be integrated into the system. First it must contain one input method with specified input parameters of the method and a XML description of the output. Because of output of individual methods can be different (e.g waveforms, discrete points, etc.) the XML format ensures its easier future representation. Second, the method has to have a descriptive properties file that contains the name of the input method, package and class. When the method is invoked by the described External Method Invoker, it reads all parameters from the properties file and configures the created Process.



Logged as:
jezekp@kiv.zcu.cz

SUCCESSFULLY LOGGED IN!

Welcome to EEG Data Processor

EEG Data Processor is application for EEG/ERP researchers developed in University of West Bohemia. Below you can see lists of available processing methods that can researcher apply over his/her provided data, which comply one of the supported file formats. There are two possible ways to use EEG Data Processor:

1. Web frontend – you are currently in it, just register yourself and process your files!
2. SOAP webservice - in order to implement our application as third party service provider, you can use our web service interface and create implement your own client!

We hope you will find EEG Data Processor useful and valuable for your research!

List of available processing methods:

- DWTPlugin-1.0.0
- FIRPlugin-1.0.0
- MPPlugin-1.0.0

Available units for processing

5

List of supported formats:

- KIV_FORMAT
  - VHDR
  - EEG
  - VMRK - optional

©EEGDatabase 2012

Figure 2: Web Interface Overview.

4.4 System Security

Since the system is public available through the Internet, secured access has to be ensured. The system is protected by system accounts with two possible user roles. The *user* can upload custom data that can be processed while the *administrator* can manage registered users and configure the system. At the implementation level, the security is ensured by Spring Security framework with protected user passwords by a cryptographic hash function with randomly generated salt.

When the user accesses the system through the Web Services API, the SSL transfer is ensured and users credentials are required.

5 IMPLEMENTED PLUG-INS

We have already implemented methods that are most often used during experiments within our laboratory. Implemented methods are common Java libraries extended according to the description in Section 4.3.1. It ensures their implementation outside the system independently. We provide implemented plug-ins open source hosted on the public repositories (GitHub, SorceForge). The list of implemented methods is gradually expanding.

Currently, we have implemented the following methods:

- *Complete and Discrete Wavelet Transform* (Mallat, 1999) are a class of a functions used to localize a given function in both space and scaling; and has some desirable properties compared to the Fourier transform. The transform is based on a wavelet matrix, which can be computed more quickly than the analogous Fourier matrix.
- *Matching Pursuit* (Ferrando et al., 2002) decomposes any signal into a linear expansion of waveforms that are selected from a redundant dictionary of functions. These waveforms are chosen in order to best match the signal structure. The dictionary is often based on Gabor functions.
- *Fast Independent Component Analysis* (Hyvärinen et al., 2001) is used for finding a linear representation of nongaussian data so that the components are statistically independent, or as independent as possible. Such a representation seems to capture the essential structure of data in many applications, including feature extraction and signal separation.
- *Fast Fourier Transform* (Rao et al., 2010) is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse. the discrete Fourier transform (DFT) is a specific kind of discrete transform, used in Fourier analysis. It transforms one function into another, which is called the frequency domain representation.
- *Finite Input Response* is a digital filter that have an impulse response which reaches zero in a finite number of steps. It can be implemented non-recursively by convolving its impulse response.

6 CURRENT STATUS

The system is almost fully implemented and prepared for testing. We put it on the testing server where it is tested by the research group members. After that we provide the system for testing by the limited set of selected collaborative partners. When testing ends, we release the system on the production server.

7 FUTURE WORK

When the system is released, the next step will be an implementation of a client side within the EEG/ERP Portal. It allows computational portal users to work more comfortably because the load will be distributed between two servers. A faster response will be ensured.

Since a need to associate methods into workflows was mentioned in Section 2, we plan to implement a wokflow module into the system. This module will ensure a development of custom workflows and the possibility to store them within users' accounts. An interface with a drag and draw functionality ensures its easy usability.

When signal processing methods implemented in various programming languages (e. g. Python, C/C++, Pearl) exist, the next technological challenge leads in providing a wrapper for a set of most often used programming languages in signal processing.

Further, we plan to investigate possibilities in the area of Cloud Computing, because of a potential load of the server will increase together with the raising number of users. The suitable cloud should help us to improve management of system resources.

8 CONCLUSIONS

The difficulties related to the processing of data from EEG/ERP experiments are presented. Since implementation of present signal processing methods is usually intended for local usage we decided to propose and implement a custom system. The aim of the presented system is to serve a wide community of researchers to share experimental methods.

The system combines research in the EEG/ERP domain with modern software engineering approaches. It helps to enhance research efficiency and enables faster achievement of scientific results.

The most often used methods in our laboratory are briefly presented. The presented methods are already implemented and integrated within the system as plug-ins. Due to a powerful plug-in engine interested users are welcome to implement a custom method according to a described procedure. We are able to assume these plug-ins and incorporate them into the system.

ACKNOWLEDGEMENTS

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

REFERENCES

Ferrando, S. E., Kolasa, L. A., and Kovacevic, N. (2002). Algorithm 820: A flexible implementation of matching pursuit for gabor functions on the interval. *ACM Trans. Math. Softw.*, 28(3):337–353.

Gardner, D., Akil, H., Ascoli, G. A., Bowden, D. M., Bug, W. J., Donohue, D. E., Goldberg, D. H., Grafstein, B., Grethe, J. S., Gupta, A., Halavi, M., Kennedy, D. N., Marenco, L. N., Martone, M. E., Miller, P. L., Müller, H.-M., Robert, A., Shepherd, G. M., Sternberg, P. W., Essen, D. C. V., and Williams, R. W. (2008). The neuroscience information framework: A data and knowledge environment for neuroscience. *Neuroinformatics*, 6(3):149–160.

Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M. R., Li, P., and Oinn, T. (2006). Taverna: a tool for building and running workflows of services. *Nucleic Acids Research*, 34(suppl 2):W729–W732.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. Adaptive and Learning Systems for Signal Processing, Communications, and Control. J. Wiley.

Jezek, P. and Moucek, R. (2010). Database of eeg/erp experiments. In Fred, A. L. N., Filipe, J., and Gamboa, H., editors, *HEALTHINF 2010 - Proceedings of the Third International Conference on Health Informatics, Valencia, Spain, January 20-23, 2010*, pages 222–227. INSTICC Press.

Jezek, P. and Moucek, R. (2011). Integration of signal processing methods into eeg/erp system. In Traver, V., Fred, A. L. N., Filipe, J., and Gamboa, H., editors, *HEALTHINF 2011 - Proceedings of the International Conference on Health Informatics, Rome, Italy, 26-29 January, 2011*, pages 563–566. SciTePress.

Jezek, P. and Moucek, R. (2012). Semantic web in eeg/erp portal - extending of data layer using java annotations. In Conchon, E., Correia, C. M. B. A., Fred, A. L. N., and Gamboa, H., editors, *HEALTHINF 2012 - Proceedings of the International Conference on Health Informatics, Vilamoura, Algarve, Portugal, 1 - 4 February, 2012*, pages 350–353. SciTePress.

Mallat, S. (1999). *A wavelet tour of signal processing (2. ed.)*. Academic Press.

Rao, K., Kim, D., and Hwang, J.-J. (2010). *Fast Fourier Transform - Algorithms and Applications (Signals and Communication Technology)*. Springer.

Schlogl, A. and Brunner, C. (2008). Biosig: A free and open source software library for bci research. *Computer*, 41(10):44 –50.

Van Pelt, J. and van Horn, J. (2007). 1st incf workshop on sustainability of neuroscience databases. *Workshop report*. Stockholm.

Watson, P., Hiden, H., and Woodman, S. (2010). e-science central for carmen: science as a service. *Concurr. Comput. : Pract. Exper.*, 22(17):2369–2380.

Watson, P., Jackson, T., Pitsilis, G., Gibson, F., Austin, J., Fletcher, M., Liang, B., and Lord, P. (2007). The CARMEN Neuroscience Server. In *Proceedings of the UK e-Science All hands Meeting*, pages 135–141.

Zito, T., Wilbert, N., Wiskott, L., and Berkes, P. (2008). Modular Toolkit for Data Processing (MDP): A Python Data Processing Framework. *Frontiers in neuroinformatics*, 2.