INTEGRATION OF SIGNAL PROCESSING METHODS INTO EEG/ERP SYSTEM

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

Keywords: EEG/ERP signal processing, EEG/ERP experiment, Independent component analysis, Principal component analysis, FastICA, Matching pursuit, Plug-in, PluginEngine, JERPA, JUIGLE.

Abstract: This paper deals with the processing of signals from EEG/ERP experiments. Selected methods for EEG/ERP signal processing are shortly presented. Since these methods are usually implemented for various platforms and within various tools, a new designed and developed software system, which allows users running them uniformly, is presented. The methods are added and processed as plug-ins. It ensures a high modularity and flexibility of the system. The main system components (JERPA, JUIGLE) are introduced and shortly described.

1 INTRODUCTION

In our department (Department of Computer Science and Engineering, University of West Bohemia, Pilsen) we specialize in the research focused on attention, especially attention of drivers or seriously injured people. We use the methods of electroencephalography (EEG) and Event-Related potentials (ERP). We collaborate with many institutions e. g.: (Czech Technical University in Prague, University Hospital in Pilsen, Škoda Auto, Inc…).

Our research group performs experiments according to designed scenarios in a well equipped EEG/ERP laboratory. Data/metadata from experiments are stored into developed EEG/ERP portal (Ježek and Mouček, 2010). When data/metadata from experiments are obtained we process them using various signal processing methods. The existing methods are implemented by various vendors. These methods are usually platform dependent and use various user interfaces; they usually need to install additional frameworks or virtual machines for their running.

Since the work with these tools is unfriendly we have decided to implement own framework. This framework integrates implemented methods – they are added as plug-ins. Integration with the developed portal will be ensured.

2 METHODS FOR EEG/ERP SIGNAL PROCESSING

EEG/ERP signal obtained from a recording device is often distorted by artifacts originated from muscle activity, eyes movement, breathing, etc. Some methods used for signal separation, signal decomposition and removal of artifacts are described in this section.

2.1 Independent Component Analysis

The goal of independent component analysis (ICA) (Hyvärinen, Karhunen and Oja, 2001) is to find 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 the data in many applications, including feature extraction and signal separation.

2.1.1 FastICA

FastICA (Oja and Zhijian, 2006) is algorithm that solves some difficulties of ICA. The most important is very fast convergence, there is no need to set a step size. The algorithm finds independent components directly; they are calculated one after another.

Ježek P. and Mouček R..
INTEGRATION OF SIGNAL PROCESSING METHODS INTO EEG/ERP SYSTEM .
DOI: 10.5220/0003165305630566
In *Proceedings of the International Conference on Health Informatics* (HEALTHINF-2011), pages 563-566
ISBN: 978-989-8425-34-8

2.2 Principal Component Analysis

Principal component analysis (PCA) (Jolliffe, 2002) involves a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.

2.3 Matching Pursuit

Algorithm (Mallat and Zhang, 1993) 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.

3 SYSTEM SPECIFICATION

3.1 System Context

Because of uncomfortable work with existing methods for EEG/ERP signal processing we have decided to design and implement a custom software tool (called the system in the following text). The system should serve not only for our purposes but also for our collaborators or users interested in signal processing.

We believe that the developed system will help to remove difficulties with using signal processing methods and will increase efficiency of neuroscientific research.

3.2 System Requirements and Users

The system is primarily designed for our department and our collaborators, but we suppose that the other users interested in signal processing will welcome this tool.

Hence the system has to be suitable for users who have only basic computer knowledge. These users want to work with a simple interface with possibilities to install updates and new plug-ins (methods for signal processing). In addition, advanced users can implement their own signal processing methods and provide them within the system as plug-ins.

3.3 Project Scope

The system is developed as a standalone desktop product with data access into various data sources (e.g.: developed EEG/ERP portal).

The system and contained plug-ins are based on the Java platform and they are provided as an open source. It ensures a higher availability for potential users and faster development of new plug-ins. The main parts of the system are:

- GUI framework – support for creating graphical components presenting output of implemented methods.
- Plug-in engine – engine for installing new methods as plug-ins into the system
- Error report engine – framework for sending errors occurred in the system (e.g. email, log files, or SMS)
- Data store engine – connection of the system to existing EEG/ERP portal (or another databases).
- Localization support – user interface is translated into a user language

3.4 User Roles

Since the system processes personal data of tested subjects it is necessary to ensure the system security.

The system of user roles is developed. User has to create his/her user account, system administrator assigns user roles.

In the developed prototype we have one role "researcher". Researcher has administrator privileges, but the system is prepared to add new roles. With new roles we can restrict privileges; the user is prohibited from installing new plug-ins or processing non public data.

4 SYSTEM DESIGN AND REALIZATION

4.1 System Architecture

The system is a layered architecture. This architectural style is directly supported by used technologies and ensures high level of abstraction as well as a long term existence of the system as open source.

The next subsections describe the main modules of the system.

4.1.1 JERPA

JERPA (Java Event-Related Potential Analysis) is the main component of the system encapsulating other components (Figure 1).

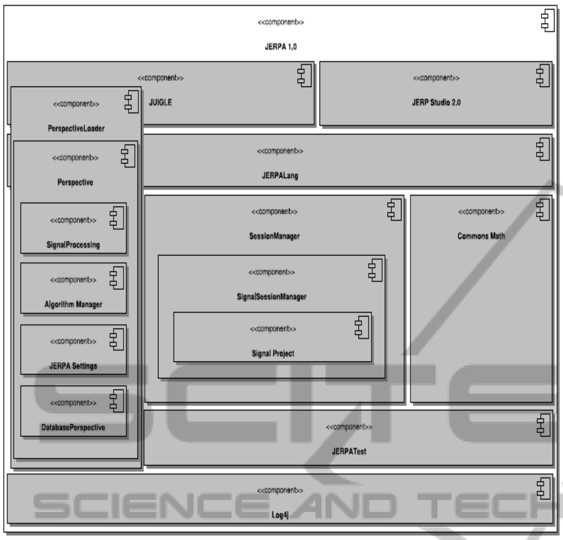

Figure 1: JERPA Component Model.

Data and application layers are based on jERP studio (Řondík, 2008). The data layer allows users to read raw EEG/ERP data and corresponding metadata in real time. The application layer provides an interface for creating EEG/ERP signal processing methods and corresponding plug-ins. It also runs a graphical user interface.

The presentation layer of the JERPA component is based on the JUIGLE library (see Section 4.1.2). It includes the following perspectives:

- Signal processing (EEG/ERP signal visualization and recording, epochs and artifacts marking, baseline correction, epochs averaging, results exports, and import from other projects)
- Algorithm Manager (list of available methods, classification and sorting of methods, selection of methods and their application to EEG/ERP signal)
- JERPA settings (language selection, activation/deactivation of plug-ins, etc.)

4.1.2 JUIGLE

JUIGLE (Java User Interface Graphic Library Environment) is a library initially providing the system user interface. It is based on the three layer architecture and allows drawing charts, which contain the results of implemented methods. JFreeChart library is used; JUIGLE provides an interface for a communication with JFreeChart. Then a user can put results into a selected chart by implementing the interface.

JUIGLE is also designed to simplify the system development. The library features include e.g. reduction of writing the same source code for GUI, a plug-in system, a database engine, an engine for reporting errors and exceptions, or a framework allowing component communication using the observer pattern. Figure 2 presents the JUIGLE component model.

JUIGLE contains a database connector for loading input data. The connector is an abstract interface, so data can be loaded from a relational database, XML or another data source. Then an implementation of the specific connector is necessary. JUIGLE also supports the project management. A user can save unfinished projects and continue in the work later.

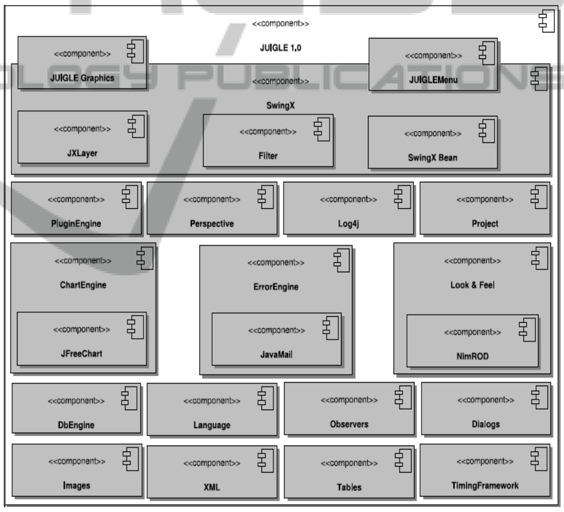

Figure 2: JUIGLE Component Model.

4.1.3 PluginEngine

Because we need the system to add new methods easily, we have also developed a plug-in engine. Plug-ins are java libraries which implement interface provided by PluginEngine. PluginEngine loads plug-ins dynamically according to XML configuration file. PluginEngine can actualize plug-ins, set properties (hidden, forbidden, active), install new plug-ins or sort them.

Plug-in developers define method input and apart from common table output they can also provide output in a chart. A developer only selects a chart type, defines axis values, and the framework ensures the rendering of selected chart.

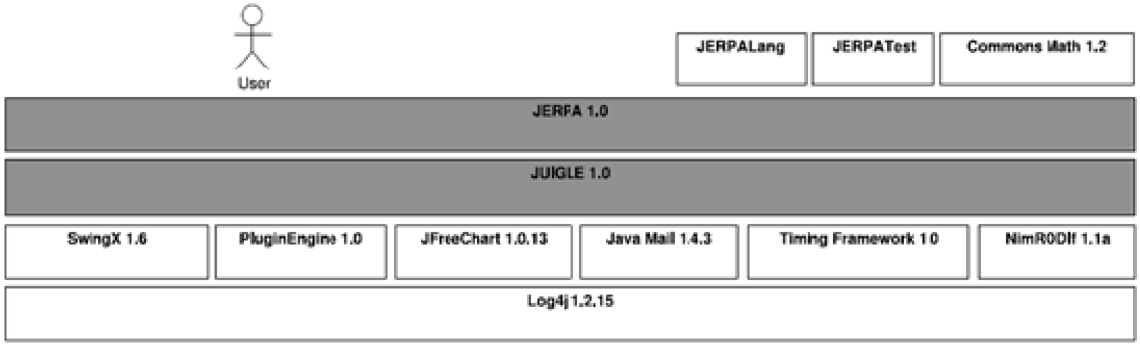

Figure 3: Component model of the system.

The methods shortly presented in Section 2 were implemented as the plug-ins.

4.1.4 Additional Modules

The system contains other modules providing additional functionality (e.g. JERPALang for JERPA localization, JERPATest for methods testing, Java Mail for email communication, NimROD for setting Look & Feel, or Log4j for logging information and error outputs. Figure 3 shows a simplified component model of the system.

5 CONCLUSIONS

In this paper we presented difficulties with the processing of data from EEG/ERP experiments. Because implementations of existing methods are often not suitable for their usage in real world applications (they are difficult to control, implemented for inconsistent target platforms, or have insufficient documentation) we designed and particularly implemented own system.

The presented system combines research in EEG/ERP with modern software engineering approaches. It increases research efficiency and makes its results widely available.

The system provides the implementation of introduced methods. Users can add their methods for EEG/ERP processing as plug-ins. Since the system is developed using common open-source technologies, an extension of the system using plug-ins is easy.

As the next step we are preparing an integration of the presented system with already developed EEG/ERP portal (Ježek and Mouček, 2010). We are going to ensure a complex way of EEG/ERP signal processing, which includes managing EEG/ERP experiments and data/metadata from them, EEG/ERP signal processing and distribution of results to collaborators or interested researchers.

ACKNOWLEDGEMENTS

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

REFERENCES

Ježek, P., Mouček, R., 2010. Database of EEG/ERP experiments.., 2010. In *Healthinf 2010 – Proceeding Third International Conference on Health Informatics.* Valencia Spain.

Hyvärinen, A., Karhunen, J., Oja, E., 2001. Independent Component Analysis. John Wiley & Sons. Hardcover.

Jolliffe, I. T. 2002. Principal Component Analysis, Springer.

Oja, E., Zhijian Y., 2006. *The FastICA Algorithm Revisited: Convergence Analysis,* IEEE transactions on neural networks.

Mallat, S. G., Zhang, Z. 1993. *Matching Pursuits with Time-Frequency Dictionaries*, IEEE Transactions on Signal Processing.

Řondík, T., 2008. Thesis. *Software for ERP processing*, Pilsen.