USING MOBILE AGENTS IN EEG SIGNAL PROCESSING

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

Keywords: Electroencephalography (EEG), Event related potentials (ERP), EEG/ERP data, EEG/ERP processing methods, EEG data formats, Artefacts detection, Mobile agents, Java, System aglets.

Abstract: Our EEG/ERP repository contains large EEG/ERP data. The partner institutions would like to work with these data e.g. to verify their processing methods, but they cannot or they are not allowed to transfer large data collections over network. The possible solution is to use a mobile agent system. The paper briefly introduces the agent system Aglets and basic EEG processing methods implemented as mobile agents. The usability of this approach is tested on selected data files. The implementation of e-mail announcements within the mobile agent system is mentioned.

1 INTRODUCTION

Our research group at Department of Computer Science and Engineering, University of West Bohemia in cooperation with other partner institutions (Czech Technical University in Prague, University Hospital in Pilsen, etc.) specializes in the research of attention, especially attention of drivers and seriously injured people. With regard to our research we widely use the methods of electroencephalography (EEG) and event related potentials (ERP). Within our partner network we are responsible for technical and scientific issues, e.g. EEG/ERP laboratory operation, development of advanced software tools used in EEG/ERP research, or analysis and proposal of signal processing methods.

EEG/ERP experiments take usually long time and produce a lot of data (sizes of data files are usually from tens to hundreds of megabytes). These data, finally stored on a computer in EEG/ERP laboratory, are then analyzed, converted and processed using various software tools and processing methods. Recently, a common tree of directories and subdirectories served as an organization scheme for EEG/ERP data. Nowadays, the first prototype of EEG/ERP database has been developed. In parallel, we found out that transfer of huge EEG/ERP data over network could be still an important trouble for some partner institutions. Moreover, the suggested network load in our laboratory would be also enormous.

An easy access to large EEG/ERP data and a possibility to work with them using own processing methods are very important requirements of our partner institutions. The following process is expected. The partner institution wants to work with EEG/ERP data, e.g. to verify a processing method on a large collection of EEG/ERP data stored in our repository. They cannot or they are not allowed to transfer this large data collection over network. Therefore they use data from the repository (create their local copy) and transfer their software code to perform a requested operation (e.g. matching pursuit algorithm). When the operation is finished the result is announced to user by e-mail.

A possible solution how to ensure this kind of remote processing of EEG/ERP data is to use the system of mobile agents based on two ideas: data are local, and operating software code (agent) is transferred. Because an agent system serves for EEG/ERP research (construction of a specific agent system was not the task), there was important to look for a system, which is at least proven in some different domain and seems to be promising for our needs. Other necessary conditions include possibilities of easy deployment on various computer platforms and extensibility of the selected agent system (open source software).

The following parts of the paper briefly introduce the selected agent system and some basic EEG processing methods implemented as mobile agents. The most important part of the paper describes the testing of usability of the selected agent system for our research. The running time of

Mouček R. and Šolc P. (2010).
USING MOBILE AGENTS IN EEG SIGNAL PROCESSING.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Agents*, pages 155-158
DOI: 10.5220/0002706301550158

EEG signal processing methods implemented in one way as common Java applications and secondly within a corresponding mobile agent system is compared. The implementation of e-mail announcements within the mobile agent system is mentioned.

2 SYSTEM AGLETS

According to our needs and general recommendations (Šolc, 2009) we choose the system Aglets from the variety of existing mobile agent systems.

System Aglets is a Java mobile agent platform and library that eases the development of agent based applications. An *aglet* is a Java agent able to autonomously and spontaneously move from one host to another.

System Aglets was originally developed at the IBM Tokio Research Laboratory, the Aglets technology is now hosted at *sourceforge.net* as open source project, where it is distributed under the IBM Public License.

System Aglets includes both a complete Java mobile agent platform, with a stand-alone server called *Tahiti*, and a library that allows developer to build mobile agents and to embed the Aglets technology in their applications. The system is based on callback model (Lange, 1998; “Aglets”, 2004).

3 EEG SIGNAL PROCESSING METHODS

We selected a limited set of methods for experimental implementation of mobile agents:

- Methods for conversions between EEG/ERP data formats;
- Basic methods for detection of artefacts in EEG signal.

3.1 Data Formats

There exists a variety of data formats for storing EEG/ERP data. The more spread formats and formats used in our laboratory include e.g. European Data Format (EDF and EDF+) (“EDF”, n.d.), Vision Data Exchange Format (VDEF) (“VDEF”, n.d.), and KIV format (Kučera, 2008).

European Data Format (EDF) contains an uninterrupted digitized EEG record stored in one file (a header record is followed by data records). The header content has a variable length. It identifies a testing subject and specifies the technical characteristics of recorded EEG signal. The data part contains consecutive fixed-duration epochs of the record. Despite its drawback this data format has been probably the most hopeful attempt to standardize description of EEG data.

Vision Data Exchange Format (VDEF) is used by the specific technical equipment in our laboratory. EEG record is divided into three files: a header file, a marker file and a data file. The header file describes recorded data and provides a limited set of corresponding metadata. The marker file contains information about markers in EEG signal. The data file contains raw EEG data.

KIV data format is a modification of simple ASCII format of EEG signal, where metadata (file header in ASCII) are stored in XML file and data from electrodes are stored in separate binary files.

3.2 Artefacts Detection

Artefacts contaminate EEG signal; they include e.g. blinks, skin potentials, and muscle movements. Since artefacts may lead to misinterpretations during EEG data analysis there is a high effort to find and to eliminate them.

The basic methods for artefacts rejection use a gradient criterion (the absolute difference value for neighbouring EEG signal values is computed; values exceeding a threshold are rejected) and an amplitude criterion (EEG signal values exceeding a maximum value, or not exceeding a minimum value are rejected). Both methods are implemented as standalone Java applications and also as aglets (Section 4).

EEG signal containing artefacts is labelled using a selected method. Labelled parts are then stored in a special list for the next processing.

4 IMPLEMENTATION OF AGLETS

All methods from Section 3 were implemented as aglets.

4.1 Aglet for EDF- KIV Conversion

An aglet realizing the conversion between EDF data format and KIV data format is inherited from com.Ibm.aglet.Aglet; interface MobilityListener has to be implemented. When a new aglet instance is created on a server, the following event method is invoked (method body implements the conversion from EDF data format to KIV data format).

```
public void onArrival(MobilityEvent ev)
{
 buffCreator = new BufferCreator
 (new File("file.edf"));
 KivFormatWriter kivWriter =
 new KivFormatWriter();
 kivWriter.write(buffCreator.getHeader(
 ), buffCreator.getBuffer(), (new
 File()));
}
```

The other methods from Section 3 are implemented in a similar way.

4.2 Java Mail

There is usually useful to inform a user about results of performed operation. Therefore the possibility to announce the results of aglet method by e-mail was designed and implemented. Because system Aglets does not support this kind of communication naturally, JavaMail API as a client system was used. A user has the possibility to define e.g. used mail server, the message content, which is sent as the result of aglet operation, and a list of message recipients.

5 TESTING OF AGLETS SYSTEM

There is important if implemented aglets finish their operations in time comparable to the running time of methods implemented as common Java applications. The testing of both implementations (common Java classes and aglets) was performed on the computer with EEG/ERP repository. Tahiti server was installed on this computer and was run with extended memory (256 MB).

Each test including one of the methods described above was run ten times (finally the average running time was computed). The results (average running times) of each test performed subsequently on 7 MB and 22 MB files in EFD format are available in the following tables.

Table 1: Average running times (in seconds) for conversion from EDF file to EDF file.

code type/file size	7 MB	22 MB
Java class	24 s	146 s
Aglet	27 s	148 s

Table 2: Average running times (in seconds) for conversion from EDF file to KIV file.

code type/file size	7 MB	22 MB
Java class	76 s	249 s
Aglet	79 s	250 s

Table 3: Average running times (in seconds) for detection of artefacts – gradient criterion.

code type/file size	7 MB	22 MB
Java class	3,8 s	10,0 s
Aglet	3,9 s	10,9 s

Table 4: Average running times (in seconds) for detection of artefacts – amplitude criterion.

code type/file size	7 MB	22 MB
Java class	2,5 s	6,5 s
Aglet	2,5 s	6,7 s

The results show that running of Aglets system does not slow down the computational process considerably. The following tests of implemented aglets using local university network had again a marginal influence on the system performance.

6 CONCLUSIONS

The important aim of our research was to verify if mobile agent system can help our research partners to use our EEG/ERP repository in such a way that transfers of huge data over network can be eliminated. The open source system Aglets proved its usability and it is now installed and prepared in EEG/ERP laboratory for testing by our research partners.

The possible disadvantage for the users is the necessity to install not only Java Virtual Machine, but also Tahiti server.

ACKNOWLEDGEMENTS

This work was supported by Grant Agency of the Czech Republic under the grant GA 102/07/1191.

REFERENCES

Šolc, P., 2009. *Decentralized processing of EEG signal (Decentralizované zpracování signálu)*. Thesis (in Czech). University of West Bohemia. Pilsen.

Lange, D., Oshima, R., 1998. Programming and Deploying Java Mobile Agents with Aglets. Addison-Wesley Professional.

Java mobile agent platform Aglets (Aglets), 2004. Retrieved from http://aglets.sourceforge.net/

European Data Format (EDF). (n.d.). Retrieved from http://www.edfplus.info/

Vision Data Exchange Format (VDEF). (n.d.). In *Brain Vision Recorder Manual*. Retrieved from http:// biomag.uni-muenster.de/EEGlab/docs/VisionRecorderManual.doc

Kučera, J., 2008. *Software for ERP processing, data reading and presentation (Software pro zpracování ERP, načítání a zobrazení dat).* Thesis. (in Czech). University of West Bohemia. Pilsen.