



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

# **Analytic Methods and Workflows for EEG/ERP Domain**

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

Jan Štěbeták

Technical Report Number: DCSE/TR-2013-02

June 2013

Distribution: public

## Abstract

This work first summarizes the state of the art of analytic methods and workflows for the EEG/ERP domain. Since developing analytic methods is linked to collecting and sharing experimental data, electrophysiological experiments are also presented.

Then the work deals with modelling of workflows in general and in the domain of electrophysiological experiments. Then it describes and evaluates engines used for modelling of scientific workflows. It also deals with sharing workflows and adding new analytic methods into a workflow model.

Because workflow models in the domain of electrophysiological experiments are not satisfactorily solved, there are opportunities for innovation. Defining metadata (semantics) for analytic methods and designing a processing model of these methods in this domain will be the subject of Ph.D. thesis.

## Content

1	Introduction .....	4
2	EEG/ERP Research .....	5
2.1	Electroencephalography (EEG) .....	5
2.1.1	Artifacts .....	5
2.2	Event-Related Potential (ERP) .....	6
3	Neuroinformatics .....	7
3.1	Neuroinformatics Infrastructures .....	7
3.1.1	EEG/ERP Portal .....	7
3.1.2	EEG Data Processor .....	8
3.1.3	The CARMEN Portal .....	9
3.1.4	G-Node Portal .....	9
4	Analytic Methods and Algorithms .....	10
4.1	Signal Preprocessing .....	10
4.1.1	Detection of Epochs .....	10
4.1.2	Averaging .....	10
4.2	Signal Processing .....	11
4.2.1	Fourier Transform .....	11
4.2.2	Discrete and Fast Fourier Transform .....	11
4.2.3	Matching Pursuit .....	12
4.2.4	Wavelet Transform .....	13
4.2.5	Independent Component Analysis .....	14
4.2.6	Hilbert-Huang Transform .....	15
5	Experiments .....	16
5.1	OQ Experiment .....	16
5.2	Number Identification .....	16
5.3	LED Based Odd-ball Experiment .....	16
5.4	Drivers' Attention .....	17
6	Workflows .....	19
6.1	Example of Workflows .....	19
6.2	Workflow Management System .....	19
6.3	Workflow Diagram .....	19
6.4	Types of Workflows .....	22
6.5	Open Source Workflows Engines .....	23
6.5.1	Sarasvati .....	23
6.5.2	jBPM .....	23
6.5.3	Werkflow .....	23

6.5.4	Evaluation of Engines .....	23
6.6	Workflows in Domain of Electrophysiology Experiments .....	24
6.7	Scientific Workflow Engines.....	24
6.7.1	CARMEN.....	24
6.7.2	Taverna.....	24
6.7.3	E-Science Central .....	25
6.7.4	OpenViBE .....	27
6.7.5	Evaluation.....	27
6.8	Adding new Analytic Methods into Workflow .....	28
6.9	Sharing of Workflows .....	29
6.9.1	Existing Approaches .....	30
6.9.2	Remote Procedure Call.....	30
7	Scope of Ph.D. Thesis .....	33
8	Conclusion.....	35
8.1	Aims of Ph.D. Thesis.....	35
	References .....	36
	List of Figures .....	39
	List of Abbreviations.....	41
	Appendix A – CARMEN’s Analytic Tools .....	42

# 1 Introduction

Electroencephalography (EEG) has become a very popular method in brain activity research. Since EEG is a non-invasive and relatively cheap method, it is possible to perform experiments not only in hospitals but also at universities or in small labs. Event Related Potential (ERP) is a response to a single external stimulus (sensory, cognitive, or motor event).

Neuroinformatics entails development of standards and infrastructure for data acquisition, storage, provenance, sharing, publishing, analysis, visualization, modeling and simulation of neuroscience data [4]. The main goal of International Neuroinformatics Coordinating Facility (INCF) is to develop and maintain database and computational infrastructure for neuroscientists. Currently, it also focuses on modelling workflows for complex analysis of experimental data.

For complex analysis, scientists often must combine multiple processing steps into larger “analysis pipelines” that can involve a number of custom algorithms, specialized tools, local and remote databases, and web services. These “analysis pipelines” are known as workflows. Workflows are commonly used in many areas, e.g. business workflows control processes in companies. In neuroinformatics, a workflow includes a complex set of analytic methods that process experimental data sequentially or in parallel.

This thesis is focused on methods used for processing electrophysiological (EEG/ERP domain) data and using them in workflows. Since data processing is often composed of several steps, workflows allow scientists to define and execute complex analyses.

Since this thesis is focused on analytic methods for EEG/ERP domain, the first part of the thesis describes the necessary theoretical background of EEG and ERP techniques. The second part of this work deals with neuroinformatics and its international organization (INCF and its National Nodes). Chapter 4 describes analytic methods suitable for the domain and chapter 5 deals with electrophysiological experiments performed at the University of West Bohemia. Chapter 6 describes workflows, their modelling, and usage generally and in electrophysiological domain. It also deals with sharing workflows and adding new analytic methods into a workflow model. In chapter 7, the scope of Ph.D. thesis is given.

## 2 EEG/ERP Research

### 2.1 Electroencephalography (EEG)

Electroencephalography is an electrophysiological method for monitoring brain activity. The human brain produces an electrical potential. This potential is measured by electrodes attached to the subject scalp.

An electroencephalogram (Figure 1) is used for displaying an output of a measurement.



Figure 1: An example of electroencephalogram [1]

The main clinical use of EEG is in the diagnosis of epilepsy, coma or brain death. It is also used for biofeedback therapy for people (especially young ones) who suffer from learning disabilities, hyperactivity, or impaired concentration [29].

#### 2.1.1 Artifacts

Artifacts [16] are signals of non-cerebral origin which appear in the EEG signal. Artifacts are divided into two following categories:

- **Artifacts of biological origin:** All muscles in the human body are controlled by electrical impulses. These impulses spread in CNS from the brain to muscles and produce much more electrical activity than the brain activity monitored during EEG measurement. Biological artifacts are e.g. head movements, swallow, eye movements, perspiration, etc. [29]
- **Artifacts from the surrounding electromagnetic field,** e.g. 50 Hz from mains.

An example of a few artifacts is shown in Figure 2.

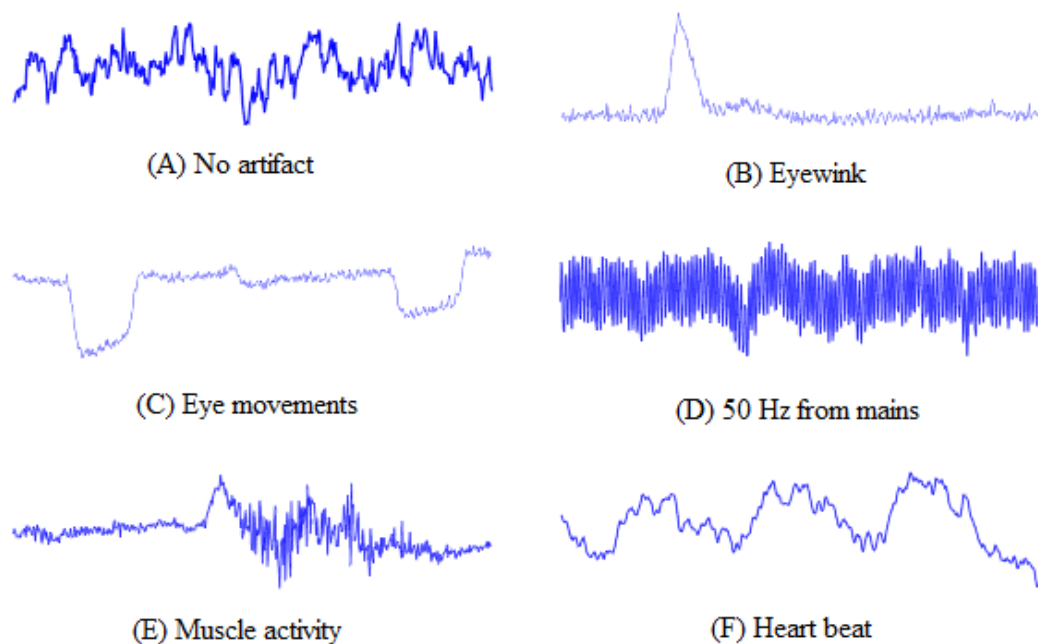


Figure 2: Examples of artifacts [2]

## 2.2 Event-Related Potential (ERP)

Event Related Potential (ERP) is a response to a single external stimulus (sensory, cognitive, or motor event). An ERP result is extracted from EEG. When extracted, ERP waveforms (components) appear. There are three properties that describe an ERP component [16].

- Latency
- Frequency
- Amplitude

The components are marked by letters - (**P** - positive, **N** - negative, **C** - the polarity is not specified or is not stable). The character is followed by a number. This number specifies latency between a stimulus and occurred component. N1 means that this component has negative amplitude and its latency is approximately 100 ms. P3 (P300) component has positive amplitude and 300 ms latency. A list of existing components can be found in [3].

In Figure 3, a few well-known ERP waveforms are shown.

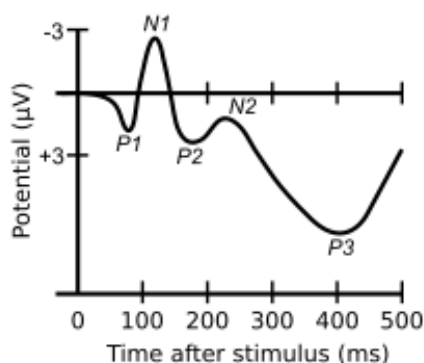


Figure 3: Well known ERP waveforms [3]

### 3 Neuroinformatics

Neuroinformatics focuses on collecting neuroscience data, storing, and sharing them. It also focuses on application of computational methods, analytic tools, and workflows. One of the definitions of neuroinformatics is specified by International Neuroinformatics Coordinating Facility [4].

Its purpose is to coordinate neuroinformatics research groups. INCF develops and maintains database and computational infrastructure for neuroscientists. Software tools and standards for the international neuroinformatics community are being developed through the INCF Programs, which address infrastructural issues of high importance to the neuroscience community [4].

INCF structure is divided into several independent Neuroinformatics National Nodes. Since research groups participate in developing, storing, and sharing analytic methods, solutions of these groups are described in the following part.

#### 3.1 Neuroinformatics Infrastructures

The following portals and databases have been developed in electrophysiology domain. They provide possibility to upload data and use analytic methods. Currently, these infrastructures are also being prepared for workflows.

##### 3.1.1 EEG/ERP Portal

The purpose of the EEG/ERP Portal (Figure 4) is to serve as a managing tool for EEG/ERP experiments that enables to share and interchange stored experiments (data, metadata, experimental scenarios, etc.) among interested laboratories [23].



Figure 4: Frontend of EEG/ERP Portal

The data are protected by the system of user accounts with defined user roles. Individual users are grouped into self-managed groups. On the basis of activities that the user can perform four user roles are recognized (Reader, Experimenter, Group Administrator, and Supervisor). The



user who wants to upload or download experiments has to create an account and to become a member of at least one group [23].

The data layer of the EEG/ERP portal uses the Hibernate framework [38] for object-relational mapping and data access. The application and presentation layer are implemented using Spring technology [39].

### 3.1.2 EEG Data Processor

Experimental data and metadata produced in the laboratory are analyzed. It resulted to development of a custom solution – EEG Data Processor.

The purpose of the EEG Data Processor [5] (frontend of this application is shown in Figure 5) is to analyze electrophysiological data.

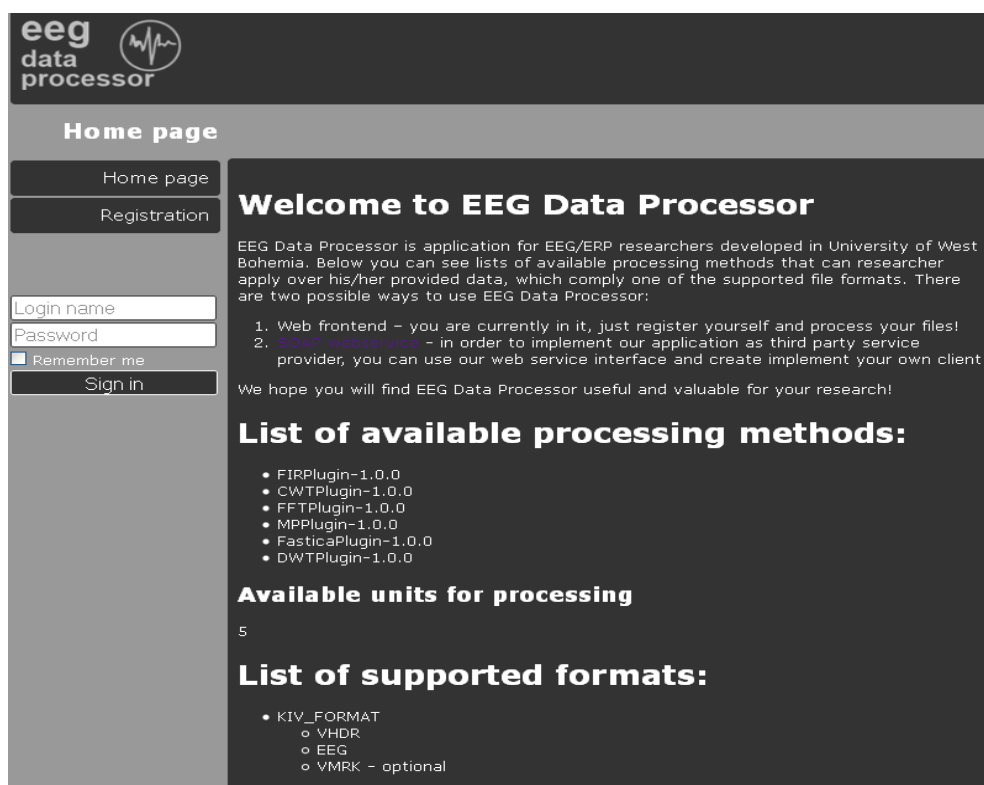


Figure 5: EEG Data Processor frontend

Registered users can use analytic methods provided by this application for analyzing their own experimental data. These methods (mentioned in chapter 4) are accessible via web browser. However, only the Brain Vision [37] data format is currently supported. The data are organized in these files:

- EEG - Includes raw EEG data in binary format
- VHDR - Header file that includes description of data format
- VMRK - Marker file that describes stimuli position

This application is written in Java language. The Spring Framework and AJAX technology for processing HTTP requests is used. For connection to the database, the Hibernate framework is used. The authentication process is ensured by the Spring Security framework.

### 3.1.3 The CARMEN Portal

The CARMEN Portal [8, 22] (*Code Analysis, Repository & Modelling for e-Neuroscience*) developed by British National Node allows neuroscientists to save and share experimental data and use services. The frontend of this portal is given in figure 6. This Figure also shows a set of features provided by CARMEN.

**CARMEN**

Welcome to the portal for the CARMEN Virtual Laboratory (VL) for neurophysiology - please view using the latest version of Internet Explorer, Firefox, Chrome, Safari or Opera.

More information can be found on [www.carmen.org.uk](http://www.carmen.org.uk). The CARMEN VL allows neuroscientists to share data and services with other users. It provides a secure online environment for:

- Storage of experimental or simulation data.
- Storage and execution of analysis code.
- Storage of the metadata associated with data and code using standard descriptions.
- Storage of experimental protocols.
- Analysis of data.
- Controlled and secure sharing of data, analysis code, analysis results and experimental protocols, etc. with other chosen users e.g. sharing other members of a research group, sharing between different research groups, etc.
- Use of standard data exchange formats.
- Curation of data, analysis code and experimental protocols.
- Making new contacts in similar research areas.

The system provides access to data and services some of which are publicly available to view and download. If you are not "logged in" you will not be able to see any data, service or workflow assets. In order to gain full access to the system you will need to register using the "register" panel to the right of the screen. Once registered and logged in you will be able to upload your own data and analysis code assets, run analyses, share your assets, and use other users assets (with permission).

First time users of the system can find a more detailed introduction on how to use the system here.

Users interested in the Neuroinformatics Data Format (NDF) Toolbox can download it here.

Only academic or related users will be permitted to use the system.

Please report any problems to the support team.

The CARMEN VL is funded by the UK EPSRC on contract number EP/E002331/1 and by the UK BBSRC on contract BB/1000984/1.

Please read the [Legal Statements](#).

Would you like to go back to the previous portal version? [Take me to the old look portal.](#)

Username Password  
Login Login >  
Problems logging in?

Sign up and get started with CARMEN >

Take the CARMEN tour

Figure 6: CARMEN System [22]

CARMEN provides storage of services. Collaborators can upload their analytic tools as web services and share them. These tools can be run on the stored data. The computing is executed on CARMEN's machines. Examples of methods for analysis of electrophysiological data are given in the attachment A.

### 3.1.4 G-Node Portal

G-Node Portal [25] provides the data management and data sharing platform for the German National Node. This service is provided for neuroscientists to facilitate data access, storage, analysis, and sharing. They may store and organize their data, exchange data with collaborators, search for published data or scientists with similar interests.

## 4 Analytic Methods and Algorithms

The following sections briefly describe the methods suitable for EEG/ERP signal analysis. These methods are used for detection ERP waveform (e.g. P3 component) or artifact removal.

### 4.1 Signal Preprocessing

A pure EEG signal contains lot of artifacts; ERP waveforms are hidden. Therefore, signal preprocessing methods are used for suppressing artifacts and obtaining ERP waveforms. A preprocessed signal is suitable for analyzing by signal processing methods.

#### 4.1.1 Detection of Epochs

An EEG signal is divided into epochs. Each epoch starts at the time when a stimulus appeared and its length depends on the latency and length of ERP waveforms.

In Figure 7, the first minute of stimulation is shown. The epochs are highlighted.

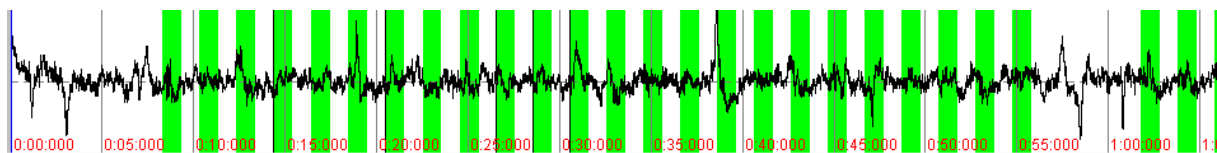


Figure 7: An example of an EEG signal with detected epochs [16]

In ERP experiments, several types of stimuli are used (auditory or visual stimuli, etc.)

#### 4.1.2 Averaging

Averaging [16] is a common method for highlighting ERP waveforms. During the averaging of the same kind of ERP waveforms, the noise is reduced and the waveform is highlighted (see Figure 8).

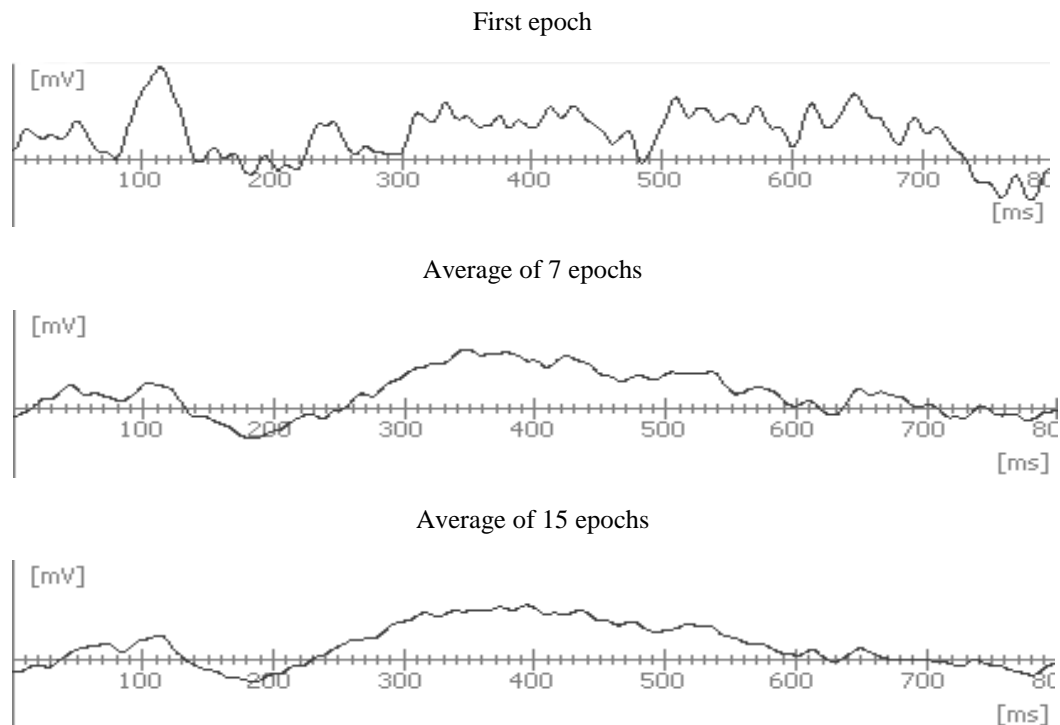


Figure 8: Averaging epochs [16]

Since the background EEG has a higher amplitude than ERP waveforms, the averaging technique highlights the waveforms and suppresses the background EEG [36].

A set of epochs is the input of the averaging method. The output of this method is an averaged signal belonging to a specific stimulus. This technique is shown in Figure 9.

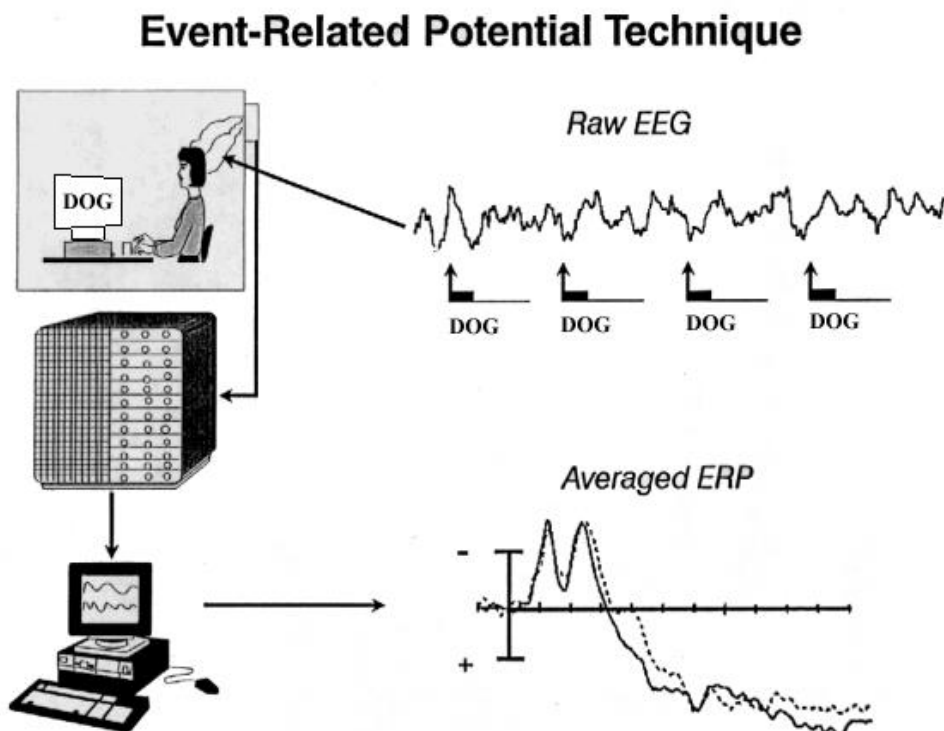


Figure 9: Averaging technique [9]

## 4.2 Signal Processing

### 4.2.1 Fourier Transform

Fourier transforms (FT) [26] take a signal and express it in terms of the frequencies of the waves that the signal includes. Figure 10 shows some examples of FT.

### 4.2.2 Discrete and Fast Fourier Transform

Experimental data usually consists of discrete data points rather than a continuous function. The Discrete Fourier-Transform (DFT) [26] is an algorithm for doing the transform with discrete data. The DFT is an order  $N^2$  calculation, meaning that the number of multiplications is equal to the square of the number of data points. This algorithm has been supplanted by Fast Fourier-Transform (FFT) algorithms, which reduce redundancies and take much less computer time. The order of this calculation is  $N \log N$ .

For example, calculated directly, a DFT on 1,024 (i.e.,  $2^{10}$ ) data points would require

$$n^2 = 1,024 \times 1,024 = 2^{20} = 1,048,576$$

multiplications. The FFT algorithm reduces this to about

$$(n/2) \log_2 (n) = 512 \times 10 = 5,120$$

multiplications, for a factor-of-200 improvement.

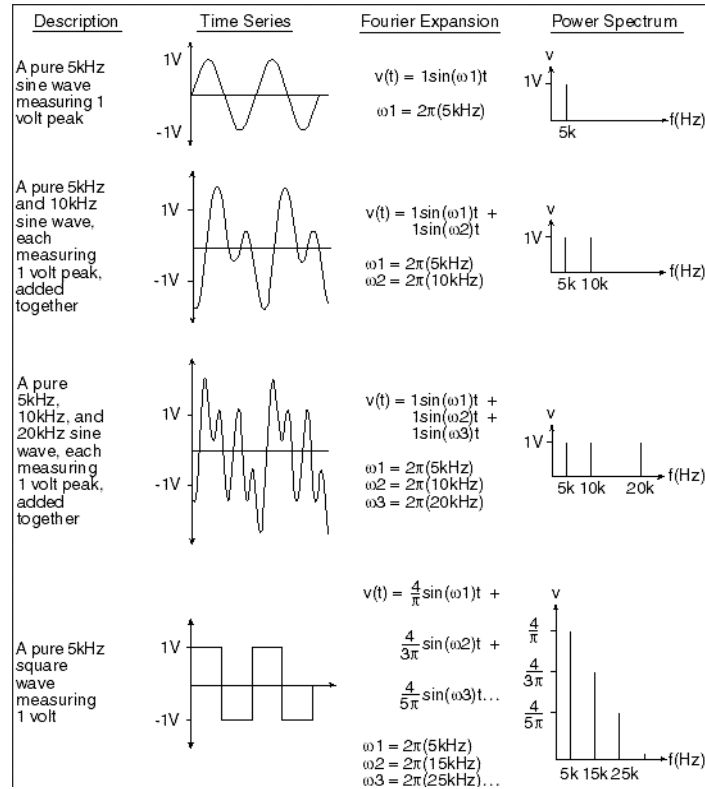


Figure 10: An example of Fourier transform [26]

The FFT function automatically places some restrictions on the time series to be evaluated in order to generate a meaningful, accurate frequency response. Because the FFT function uses a base 2 logarithm by definition, it requires that the range or length of the time series to be evaluated contains a total number of data points precisely equal to a 2-to-the-nth-power number (e.g., 512, 1024, 2048, etc.). For example, if your time series contains 1096 data points, you would only be able to evaluate 1024 of them at a time using an FFT since 1024 is the highest 2-to-the-nth-power that is less than 1096 [26].

In EEG/ERP domain, the transformation of a signal into a frequency domain is commonly used for artifacts detection.

### 4.2.3 Matching Pursuit

The matching pursuit (MP) algorithm is frequently used for continuous EEG processing. It decomposes any signal into a linear expansion of functions called atoms. It means that the input signal  $x$  can be expressed by atoms  $g_n$  and suitable constants  $a_n$  as follows [13, 16]:

$$x \approx \sum_{n=0}^{N-1} a_n g_n \quad (1)$$

At each iteration, a waveform is chosen in order to best match the significant structures of the signal. Typically, this part is approximated by a Gabor atom, which has the highest scalar product with the original signal, and then it is subtracted from the signal. This process is repeated until the whole signal is approximated by Gabor atoms with an acceptable error [11].

The MP algorithm is most often associated with Gabor atoms dictionary. Gabor atoms are defined as the Gaussian window:

$$g(t) = e^{-\pi t} \quad (2)$$

For each iteration, the input signal which enters the iteration, chosen atom, and the difference between input signal and chosen atom are shown. At the bottom, the input signal reconstruction made of chosen atoms is shown [16].

For displaying results is commonly used the time-frequency transformation known as Wigner-Ville transformation [12]. The input of this transformation is a set of chosen atoms.

During recordings of the brain activity ERPs appear just like the signal trends, which are disturbed by the EEG signal. After several iterations the input signal is approximated by the atoms in such a way that the signal trend is highlighted [46]. An example of detecting P3 component is shown in Figures 11 and 12.

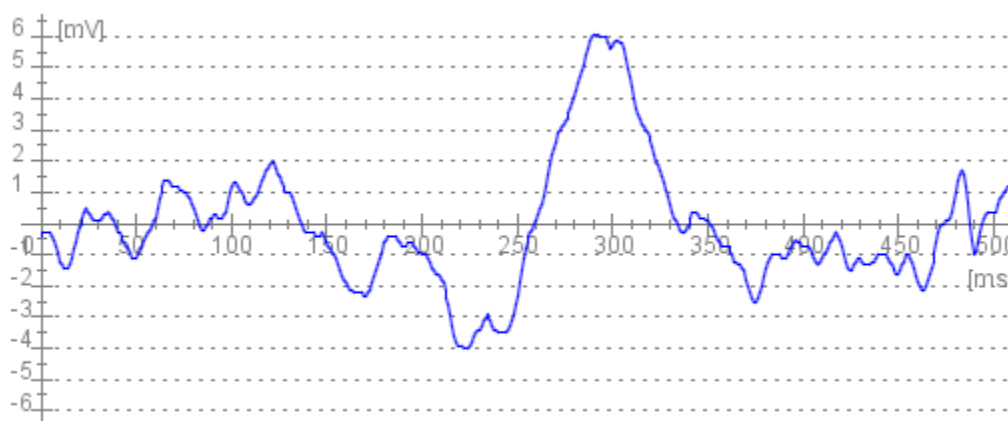


Figure 11: ERP signal containing P3 [46]

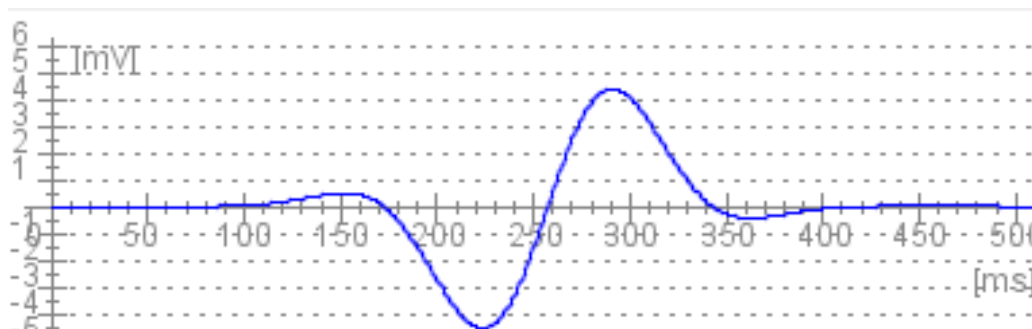


Figure 12: Gabor atom which best approximates P3 components [46]

#### 4.2.4 Wavelet Transform

Wavelet Transform (WT) [13, 16] is a suitable method for analyzing and processing non-stationary signals such as EEG. For EEG signal processing it is possible to use continuous wavelet transform (CWT) or discrete wavelet transform (DWT). Both CWT and DWT were tested during our research focused on automatic ERP detection. WT is a suitable method for ERP detection because it has a good time and frequency localization (see Figure 13 for an example of some well-known wavelet functions).

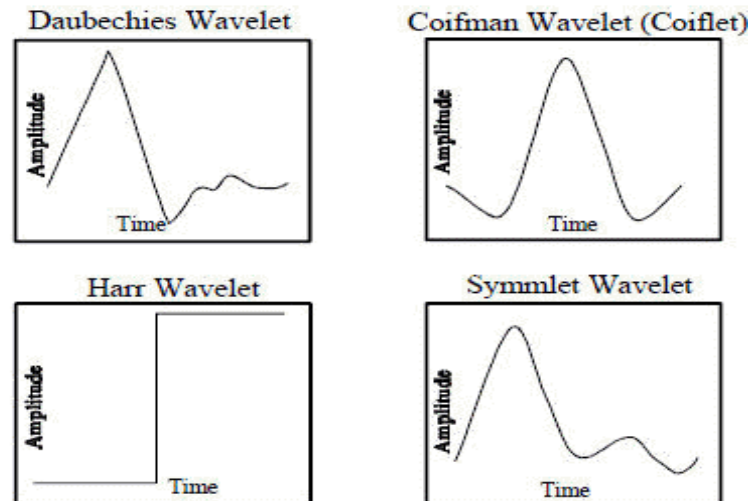


Figure 13: Some well-known wavelets [14]

- **Discrete Wavelet Transform**

DWT is common in computer science because of high performance caused by its algorithmic complexity. There are implemented a lot of wavelets which can be used in DWT. In automatic ERPs detection it is necessary to have a wavelet which corresponds to a detected ERP component as much as possible.

- **Continuous Wavelet Transform**

CWT is often replaced in computer science by its discrete form because of its algorithmic complexity. The result of the wavelet transform is visualized in a scalogram (Figure 14), where each coefficient represents a degree of correlation between the transformed wavelet and the signal. The scalogram is gray scaled and the highest values are white.

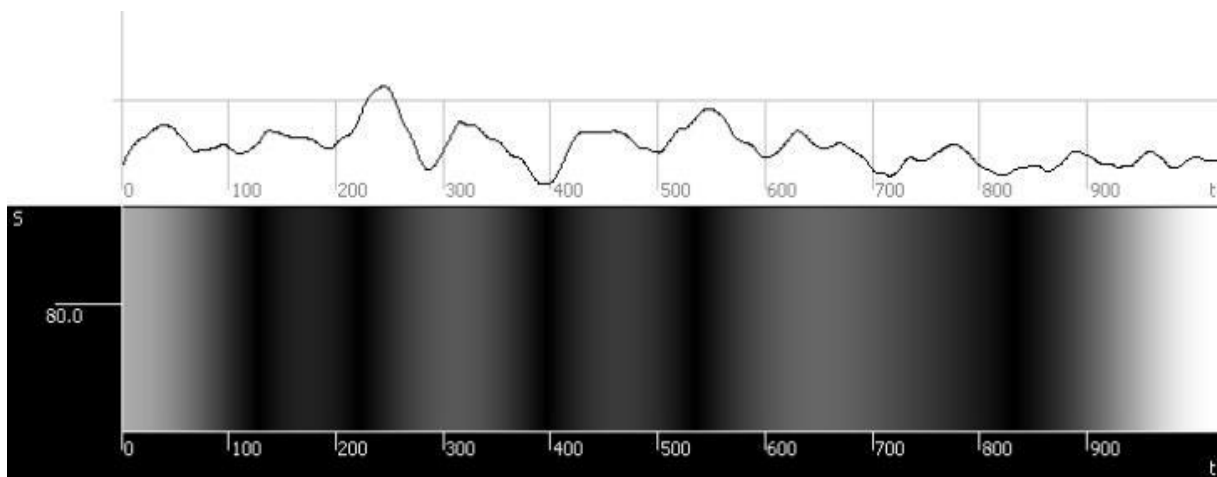


Figure 14: Input signal and its scalogram. [16]

#### 4.2.5 Independent Component Analysis

Independent Component Analysis (ICA) [15] is a method for blind signal separation and signal deconvolution. In the EEG/ERP domain, ICA can be used for artifact removal, ERPs detection, and – generally speaking – for detection and separation of every signal which is independent on EEG activity.

ICA is a quite powerful technique and is able (in principle) to separate independent sources linearly mixed in several sensors. For instance, when recording (EEG) on the scalp, ICA can separate artifacts embedded in the data (since they are usually independent of each other).

As mentioned above, ICA is a technique to separate linearly mixed sources. For instance, let's try to mix and then separate two sources (Figure 15). Let's define the time courses of two independent sources A (top) and B (bottom)

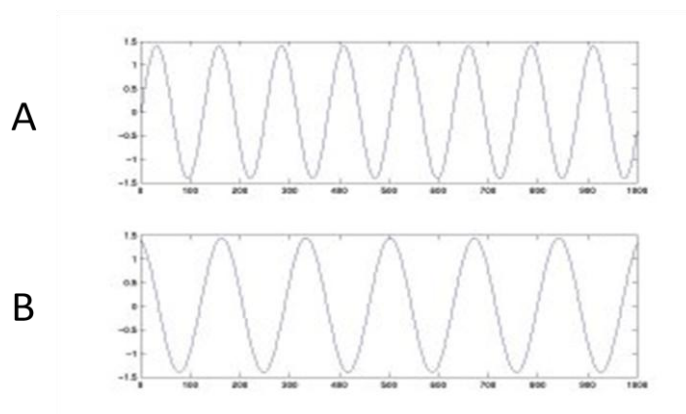


Figure 15: Independent sources A and B

We then mix linearly these two sources. The top curve is equal to  $A - 2B$  and the bottom is equal to the linear combination  $1.73A + 3.41B$  (Figure 16).

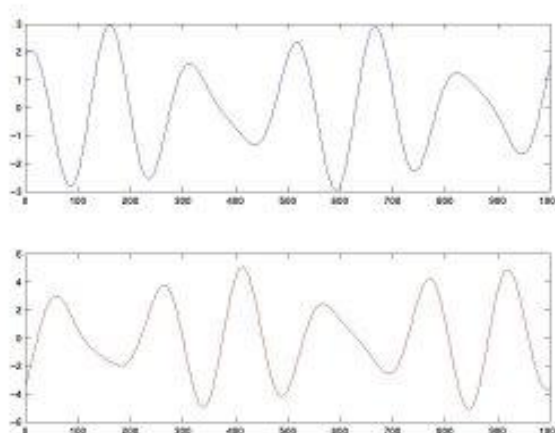


Figure 16: Mixture of sources ( $A - 2B$ )

Putting these two signals into the ICA algorithm (in this case, fastICA) this is able to uncover the original activation of A and B.

ICA algorithm is usually used for artifacts removal. This algorithm decomposes an EEG signal to a linear combination of functions. Since artifacts are usually independent of each other, a function that represents the artifact is removed from the signal reconstruction.

#### 4.2.6 Hilbert-Huang Transform

The Hilbert-Huang transform (HHT) was designed to analyze nonlinear and non-stationary signal. It includes detection of ERP waveforms. More information about HHT can be found in [47].



## 5 Experiments

Designed and developed analytic algorithms are tested on experimental data from EEG/ERP experiments. For performing these experiments, a laboratory has been continuously built at the University of West Bohemia. It is equipped with a soundproof cabin, a car simulator, 32 channels EEG recorder BrainAmp, BrainVision recording software, PresTi and Presentation software for presenting experimental protocols, computer for storing EEG data, and EEG caps (active and passive).

Scientific experiments as well as experiments producing data for testing analytic methods are performed. This section briefly describes these experiments.

### 5.1 OQ Experiment

This Classical P300 odd-ball experiment [29] focuses on visual ERPs. There are O and Q letters displayed to a subject. The O letter appears more frequently than the Q letter. Figure 17 shows an example of visual stimulus.

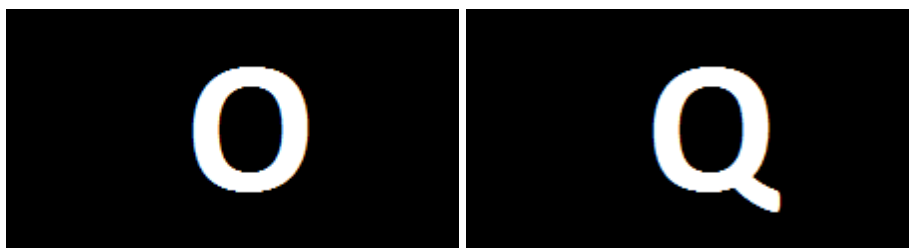


Figure 17: O and Q stimulus [29]

There are two types of stimuli:

- Target stimulus – In this case, Q letter is a target stimulus. It is expected that a brain response to a target stimulus contains the P3 waveform.
- Non-target stimulus – In this case, O letter is a non-target stimulus.

An experimenter can see an ERP signal as a response to stimuli. He/she decides, whether the P3 waveform occurs. If not, the data are discarded. Otherwise, the data are used for testing ERP waveform detecting methods or training classifiers.

### 5.2 Number Identification

The next experiment deals with *number identification*. A subject is thinking on a number (0 – 9) during the experiment. The numbers are randomly flashing on the screen with the same frequency.

After the experiment, epochs that belong to specific numbers are averaged. The experimenter obtains ten ERP waveforms for numbers 0 – 9. In ideal case, the P3 waveform is detected and belongs to the number to which the subject was thinking.

This type of experiment is used for testing classifiers based on recognition mechanism such as artificial neural networks.

### 5.3 LED Based Odd-ball Experiment

The standard odd-ball task [19] was used for subjects' stimulation. In this task, red and green LEDs representing non-target and target stimuli were randomly switched on and off for

a period of 0.5s. The probability of the target stimuli (i.e. the green LED was switched on) was set up to 0.2.

The arrangement of the typical experiment and the connection of the developed stimulator to the ERP recording system are presented in Figure 18.

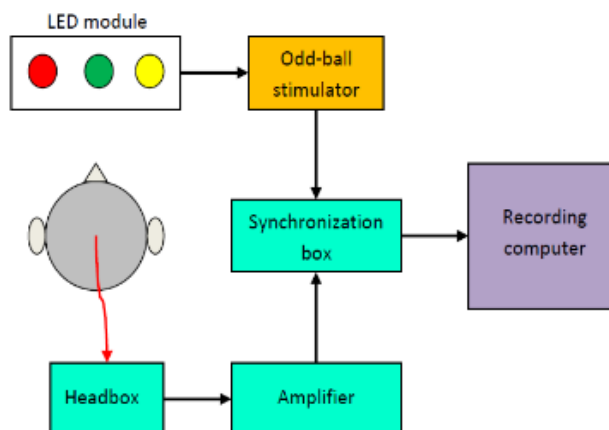


Figure 18: Arrangement of experiment and connection of stimulator to ERP recording system [19]

This stimulation is performed also for investigation of drivers' attention.

#### 5.4 Drivers' Attention

This type of experiments consists in stimulation of a subject while he/she is driving a car simulator. The subject is stimulated by obstacles on a track or by stimulating protocol.



Figure 19: The car simulator

The following experiments are performed using a car simulator (Figure 19):

- Monotonous driving - a subject is driving a car for a long time (40 - 60 minutes). The purpose of this experiment is to schedule the latency of the P3 component. This latency reflects a level of subjects' attention.

- Driving with obstacles - this experiment focuses on scheduling a reaction time of the subject under various conditions.
- A subject is driving a car; he/she is disturbed by events such as blinking diode in the simulator. The level of attention on driving the simulator is scheduled.

In all these experiments, the P3 waveform is detected. The amplitude and latency of the P3 component is crucial for looking for abnormality in behavior of the subject.

## 6 Workflows

Workflows are the combination of pipelines (i.e. modules representing individual programs with connecting pipes representing data transfer from one module to another) and data control systems that coordinate data processing on local or distributed computer architectures [35].

The term *workflow* is more commonly used in particular industries, where it may have particular specialized meanings.

- Processes
- Planning and scheduling
- Flow control

### 6.1 Example of Workflows

Workflows technique is commonly used in many areas (research or industry). The following examples show the variety of workflows.

- Business workflows represent processes in a company.
- Insurance claims processing is an example of workflows.
- In Scientific experiments, the overall process (tasks and data flow) can be described as a Directed Acyclic Graph, which is referred to as a workflow [40].
- In healthcare data analysis, a workflow can be used to represent a sequence of steps which compose a complex data analysis (data search and data manipulation steps) [41].
- In Service-oriented architectures an application can be represented through an executable workflow, where different, possibly geographically distributed, service components interact to provide the corresponding functionality, under the control of a Workflow Management System [42].

### 6.2 Workflow Management System

A workflow management system is a computer system that manages and defines a series of tasks within an organization to produce a final outcome or outcomes. Workflow management systems allow the user to define different workflows for various types of jobs or processes. For example, in a manufacturing setting, a design document might be automatically routed from designer to a technical director and to the production engineer. At each stage in the workflow, one individual or group is responsible for a specific task. Once the task is complete, the workflow software ensures that the individuals responsible for the next task are notified and receive the data they need to execute their stage of the process [18].

Workflow systems are used for transparent planning and control of every part of an enterprise – especially where employees work together and share information. Email, Excel, meetings and other costly manual coordination are minimized and general process efforts reduced. In a production environment, the system will increase productivity, improve quality and allows visibility anytime [18].

### 6.3 Workflow Diagram

A Workflow Diagram is a simple form of a flowchart depicting the flow of tasks or actions from one person or group to another. It typically consists of a set of symbols representing actions or individuals connected by arrows indicating the flow from one to another. Different

symbols represent different aspects of the workflow. Drawing workflow diagrams is similar to drawing UML diagrams.

- Action/Task - a workflow unit that is drawn using a single task in a workflow diagram.
- Embedded process – is drawn by embedded workflow diagram.

A sequence of simple tasks or actions in the workflow diagram is defined by control flow. In Figure 20, there is shown a simple workflow containing two tasks. The *Task 2* follows after the *Task 1* and is executed when the *Task 1* finishes.

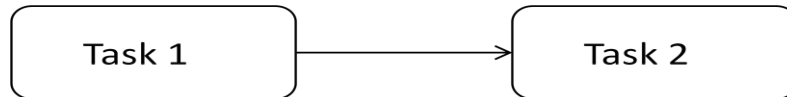


Figure 20: Simple workflow

A workflow diagram may include conditions as well. In Figure 21, the *Task 2* is executed when the *Task 1* finishes and the defined *Condition* is met.



Figure 21: Workflow with a condition

In Figure 22, a workflow diagram with more conditions is shown. The diamond shape symbol means *decision*. The *Task 2* is executed when the *Condition 1* is met; the *Task 3* is executed when the *Condition 2* is met.

It is generally used for *yes/no decision*. It usually contains one condition and one task is executed when the condition is met, the other task is executed when the condition is not met.

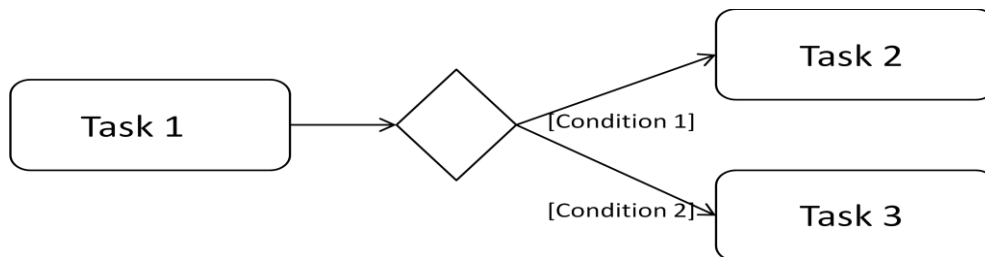


Figure 22: Workflow with more conditions

Connection of more tasks to one task (Figure 23) is also used for designing workflows. In this case, the *Task 3* is executed only when the *Task 1* and the *Task 2* are finished.

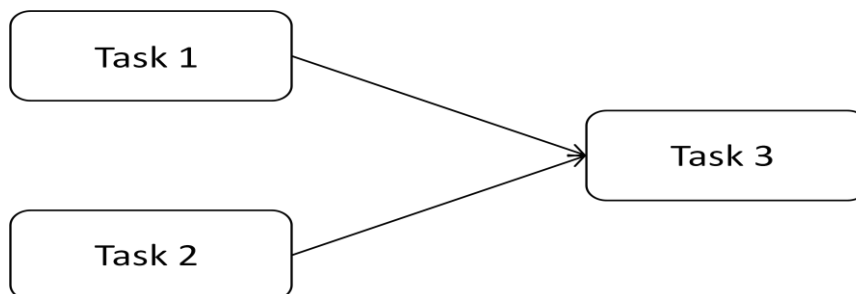


Figure 23: Connection in workflow

In Figure 24, more tasks are merged to one task. In this case, the *Task 3* will be executed when the *Task 1* or the *Task 2* is finished.

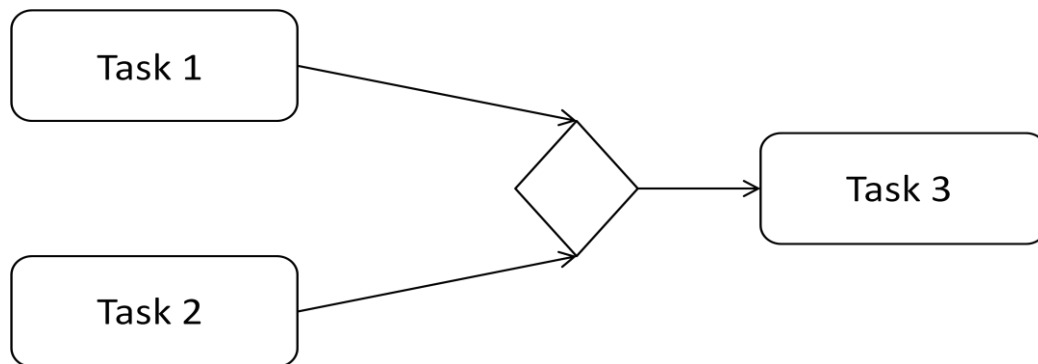


Figure 24: Merge in workflow

An example of a workflow diagram given in Figure 25 describes procedures and their order when a company recruits new employees.

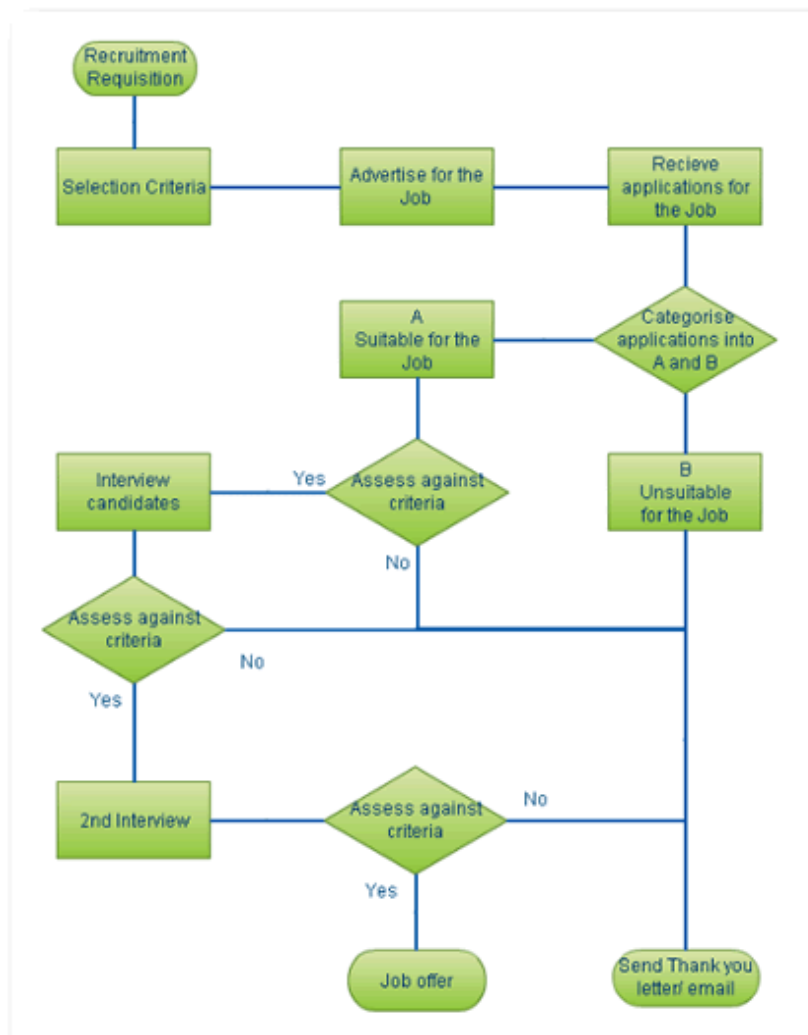


Figure 25: An example of workflows [6]

## 6.4 Types of Workflows

There are three types of workflows: serial, parallel, and series-parallel [31].

Technically, serial workflow means to put analytic methods into a pipe, where an output from the previous method becomes an input to the next method. The model of serial workflows is given in Figure 26.

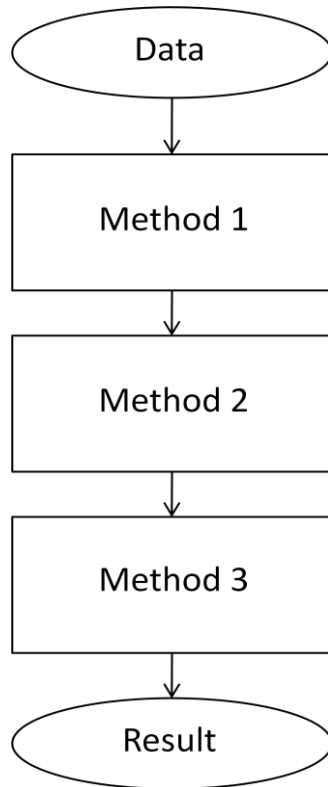


Figure 26: Serial workflow

The model of parallel workflow is given in Figure 27.

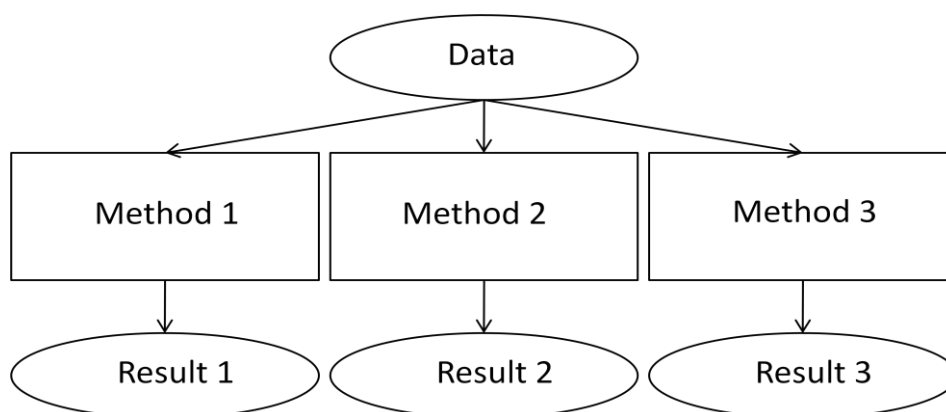


Figure 27: Parallel workflow

Series-Parallel workflow is a composition of serial and parallel workflows. It will also allow using more workflows in parallel. The model of serial-parallel workflows is shown in Figure 28.

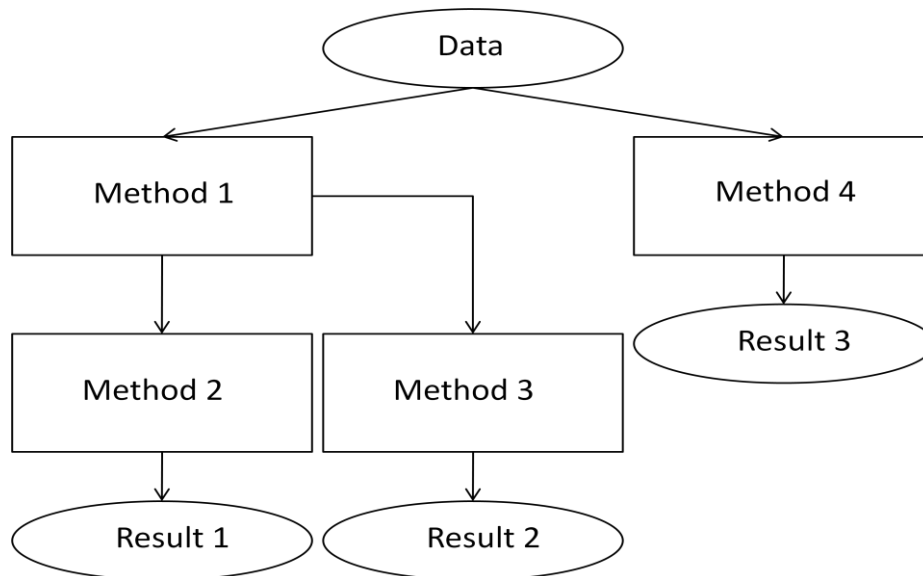


Figure 28: Series-Parallel workflow

## 6.5 Open Source Workflows Engines

This section briefly describes several open source workflows engines and evaluates their suitability.

### 6.5.1 Sarasvati

Sarasvati [20] is a capable, embeddable workflow/BPM engine for Java. For developers, it focuses on expressive modelling and ease of use features like embeddable sub-processes and backtracking. For users, it offers transparency via visualizations and human readable guards.

### 6.5.2 jBPM

jBPM is a flexible Business Process Management (BPM) Suite. It makes the bridge between business analysts and developers. Traditional BPM engines have a focus that is limited to non-technical people only. jBPM has a dual focus: it offers process management features in a way that both business users and developers like [28].

### 6.5.3 Workflow

Workflow [21] is a flexible, extensible process- and state-based workflow engine. It aims to satisfy a myriad of possible workflow scenarios, from enterprise-scale business processes to small-scale user-interaction processes. Using a pluggable and layered architecture, workflows with varying semantics can easily be accommodated. Processes can revolve around documents, objects or any other entity. The core workflow engine can be accessed through a Java API, EJB, JMS, SOAP and other conduits.

### 6.5.4 Evaluation of Engines

All these engines are suitable for Java language. However, they are primarily designed for business processes modelling. It is not suitable to use full engine for modelling workflows in EEG/ERP subdomain. Methods for EEG/ERP signal processing are very different from business processes in case of their input/output parameters. A set of syntactic and semantic rules has to be defined.



Nevertheless, an architecture (jBPM architecture is shown in Figure 29) or its modification can be used for modelling workflows in EEG/ERP subdomain as well.

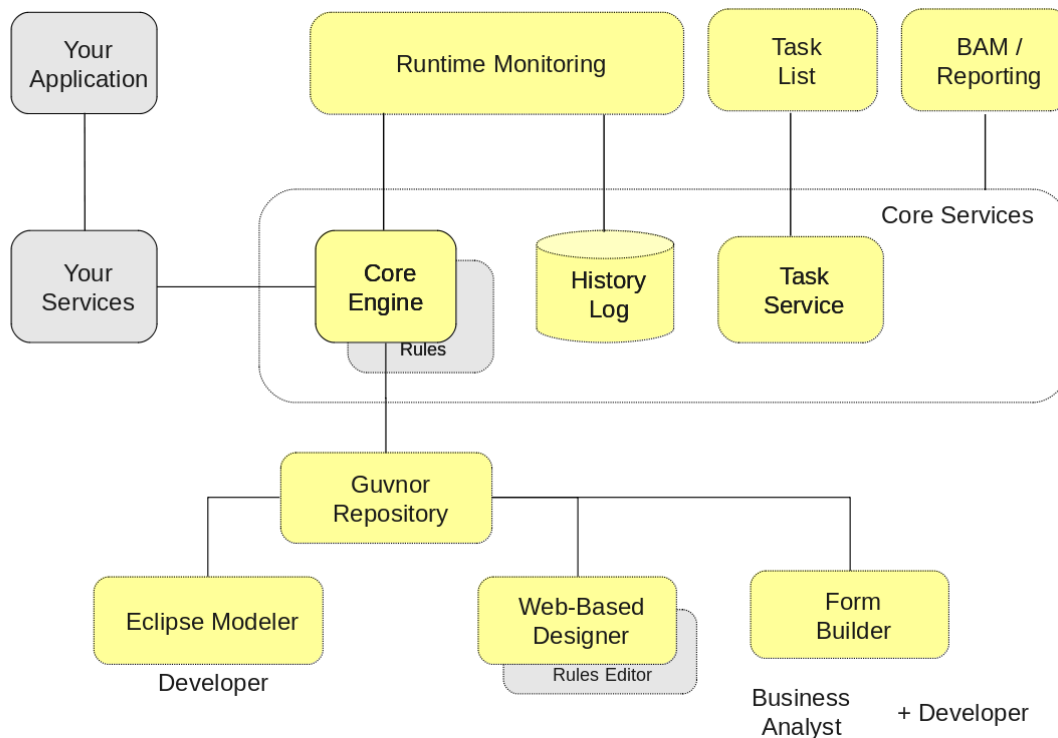


Figure 29: jBPM architecture [28]

## 6.6 Workflows in Domain of Electrophysiology Experiments

Since data processing often requires usage of more methods sequentially or in parallel, development of specific workflows is required. Workflows simplify the work with data and methods and offer more comfort to users.

It is commonly used, when original data such as raw EEG signal has to be preprocessed first using methods e.g. averaging or filtering. After this, preprocessed data are processed using methods presented in 4.2.

## 6.7 Scientific Workflow Engines

### 6.7.1 CARMEN

The CARMEN project [32] has now addressed requirements of scientists and developed a workflow generation and execution system within the platform. The CARMEN Workflow Tool is Java-based and designed to make use of CARMEN Services. The workflow tool supports both data and control flow, and allows parallel execution of services. The complete workflow tool consists of a graphical design tool, a workflow engine, and access to a library of CARMEN services and common workflow tasks.

However, this system of workflows is under development and is not provided to users of the CARMEN system.

### 6.7.2 Taverna

Taverna [33] is an open source and domain-independent Workflow Management System – a suite of tools used to design and execute scientific workflows.

The Taverna suite is written in Java and includes the Taverna Engine (used for enacting workflows) that powers both the Taverna Workbench (the desktop client application) and the Taverna Server (which allows remote execution of workflows). Taverna is also available as a Command Line Tool for a quick execution of workflows from a terminal [33].

An example of modelling workflows in Taverna is shown in Figure 30.

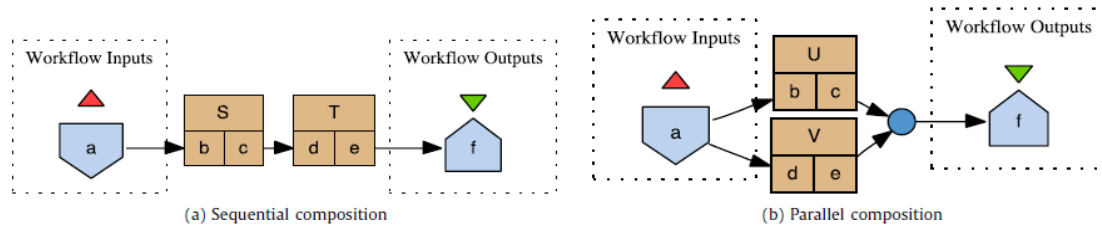


Figure 30: Workflow modelling in Taverna [31]

A Taverna workflow graph is a directed acyclic graph where nodes, called *processors*, represent software components, for instance Web services or local scripts, with an interface that consists of input and output *ports*. Arcs in the graph connect pairs of ports, and specify a data dependency from the output port of one processor to the input port of another. Additionally, a control link between a source and a sink processor can be used to specify that the sink processor cannot execute until the source processor has produced all its output [31].

Although the workflow is an acyclic graph, and thus does not permit users to explicitly describe loops over groups of processors, the Taverna model allows repeatedly execution of processes by iterating over its input ports list values [31].

### 6.7.3 E-Science Central

e-Science Central (Figure 31) is a Cloud based Platform for Data Analysis. It supports secure storage and versioning of data, audit and provenance logs and processing of data using workflows. Workflows are composed of blocks which can be written in Java, R, Octave or Javascript [34].

e-Science Central is portable and can be run on Amazon AWS, Windows Azure or your own hardware. The e-Science Central platform along with basic analysis blocks are provided under an Open Source License and are available from SourceForge. e-Science Central is developed by a team based at Newcastle University, UK.

e-Science Central allows users to analyze data, rather than just to share it. Its in-browser workflow editor allows users to build a workflow by dragging services from the left hand side of the screen onto the canvas on the right, and then connect them with arcs. A core set of services are provided for data manipulation, statistic analysis and visualization. However, the e-Science Central “Science Platform as a Service” allows developers can upload their own services into the system and share them in a controlled way, as for data [34].

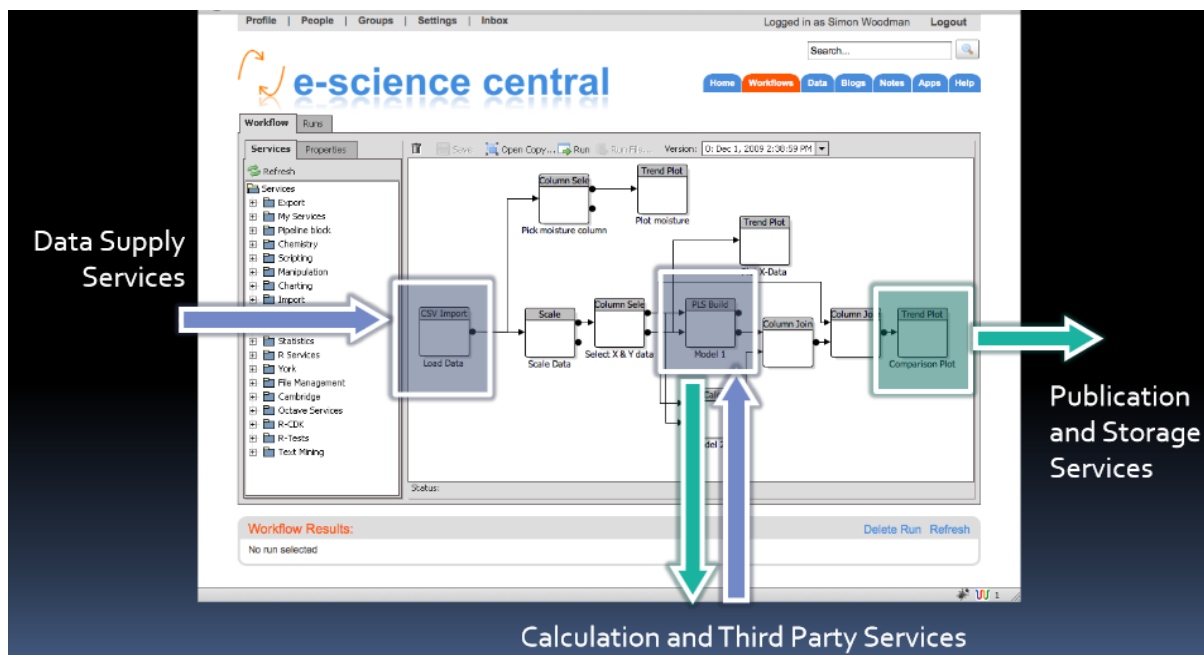


Figure 31: E-Science Central [17]

Figure 32 shows the architecture of the workflow system. Scientists are able to design workflows using the drag-and-drop online workflow designer by selecting blocks (services). Generic blocks exist to provide file management, data manipulation, mathematical modelling and visualization. Blocks of specific domain are also provided - currently for neuroscience and chemical informatics. The input and output of each block is typed to prevent incompatible blocks being connected to each other [7].

Users can write their own blocks. Supported languages are Java and R.

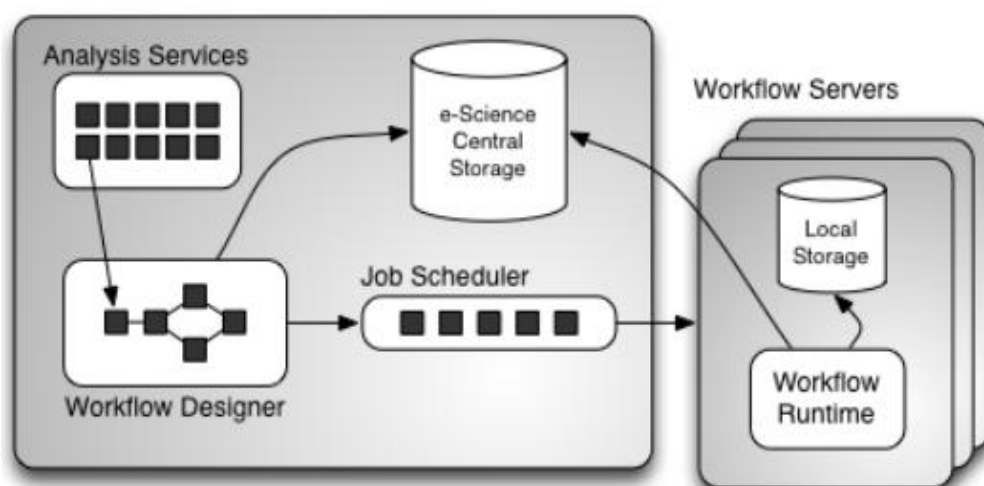


Figure 32: Workflow architecture [7]

### 6.7.4 OpenViBE

OpenViBE is a free and open-source software platform for the design, test and use of Brain-Computer Interfaces. The platform consists of a set of software modules that can be easily and efficiently integrated to design BCI for both real and VR applications [45].

OpenViBE has been designed for four types of users. On the first hand, the developer and the application developer are both programmers; on the other hand the author and the operator do not need any programming skills [45].

This software provides a designer that allows building complete scenarios based on existing software modules using a dedicated graphical language and a simple Graphical User Interface (GUI) as shown in Figure 33 [45].

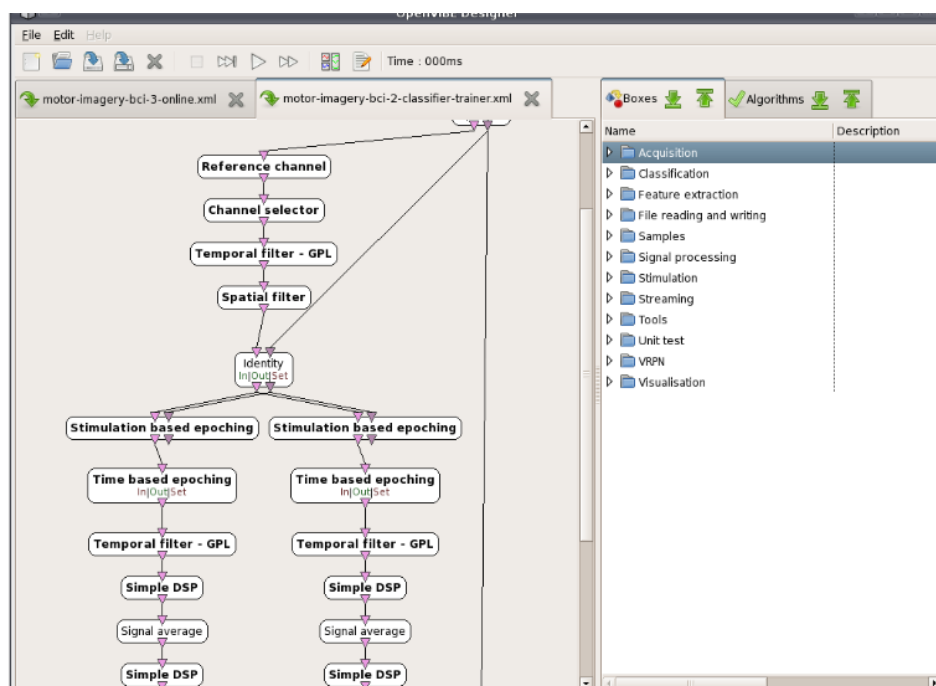


Figure 33: OpenViBE designer for scenarios [45]

This platform includes three different families of plug-in [45]:

- The driver plug-in allows adding acquisition devices to the acquisition server.
- The algorithm plug-in is a generic abstraction for any extension that could be added to the platform (e.g., add new feature extraction or signal processing methods).
- The box plug-in is the software components each box relies on. Boxes are the author's atomic objects. The developer describes them in a simple structure that notably contains the box prototype (its name, input/output connectors and settings). The box is responsible for the actual processing, i.e., it reads from inputs, computes data to produce a result and writes to outputs.

These boxes can be executed sequentially or in parallel. Each box and connected line is defined by input/output format. It ensures a format compatibility of input/output.

### 6.7.5 Evaluation

The engines described above are designed for scientific purposes. They provide modelling of workflows in many scientific areas including neuroinformatics and in the domain of electrophysiological experiments.

All of the mentioned engines use the parameter type control during data processing. It ensures that only compatible blocks can be connected. However, methods used in the electrophysiology domain are specific in case of syntax and semantics for various inputs/outputs. For example, only a subset of the result of a previous method can be used as an input to a next method. This case is not solved by these engines.

For well-designed workflows, ensuring syntactical compatibility is not satisfactory. Used methods have to be connected correctly in terms of their semantics. Currently, only Taverna deals with semantics. It is formal semantics, where restrictions for input/output parameters (e.g. available values) are defined. However, semantics of piped methods (if the connection makes sense or not) is not solved.

These tools are designed for adding new methods or boxes (including an analytic method). These methods and boxes are usually stored in a central storage of a system. e-Science Central is a cloud based engine. This engine has its own central storage, but it is also open to third-party services.

## 6.8 Adding new Analytic Methods into Workflow

Since new methods are developing, the well-designed model of workflows has to allow adding new methods.

Created workflow models are usually realized and controlled by workflow management systems. Methods, which workflows working with, are contained in these systems. Therefore, this chapter describes adding new methods into these systems.

- **Existing Approaches**

Workflows are often designed for methods, which the databases and portals provide including the possibility to add new methods. However, there are also methods designed and provided by third party as services. These methods are usually integrated using remote procedure call. The workflow model has to work not only with methods within a portal but with third party services as well.

This section describes approaches that allow developing of new methods and adding them into mentioned portals.

- **External Method Invoker**

EEG Data Processor uses External Method Invoker for running methods. This plug-in allows adding new analytic methods without changing the application itself.

External Method Invoker is responsible for execution of a requested method. It parses the method parameters and takes the method result. This library (Figure 34) has to meet following criteria.

- The initialization method must be of *void* type
- The method must have at least two parameters (an array of double values that represents a processed signal and a string parameter for the XML output file. Other parameters are optional.
- The output is described by XML. Because output of individual methods can be different, the XML format ensures its easier representation.

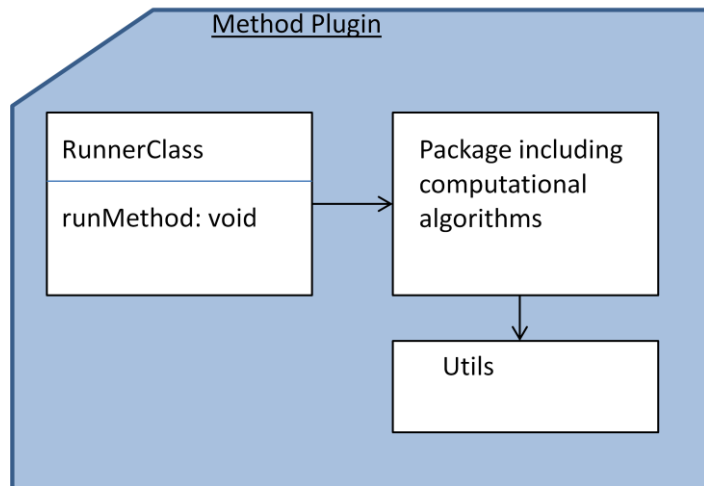


Figure 34: Structure of plug-in library

- **CARMEN Service Wrappers**

Wrapping Carmen Services [24] is an automated process that is performed using the CARMEN service builder tool. The user describes their application (inputs, outputs, descriptions, etc), adds their application in the form of a binary or script, and the builder tool creates the service in the form of JAR and XML files. These files are then be deployed on the portal so that they can become live.

In order to wrap legacy applications it is required that they are written as a command-line application. Therefore, the general rule set is [24]:

- The application is running as a service on a remote Carmen server. Therefore, the user will have no interactivity with the application. The application must be non-interactive; i.e. no key presses, mouse clicks, GUI's, etc.;
- Input parameters must be passed into the application as string arguments.
- The required output of the service must be printed to the screen/*stdout* surrounded by `<output></output>` tags.
- File handling must be performed within the application.
- When working with files, no paths must be used.
- The application writer should preferably write a wrapper around their algorithm.

The CARMEN currently can wrap Matlab, Python, R, and any language that compiles to an executable (i.e. C/C++).

## 6.9 Sharing of Workflows

Sharing is very helpful for scientific community. However, the present organization of science does not support it entirely. Scientists usually want to be the first to publish their ideas. Finally, after successful publication, the full access to data and analytic methods is possible.

The possibility to share analytic methods and workflows is important as well as possibility to share experimental data. Sharing of analytic methods and workflows improves the efficiency of scientific work.

### 6.9.1 Existing Approaches

Neuroinformatics databases and portals allowing access to them are usually designed for sharing experimental data and analytic methods. Registered users have according to their user role access to shared data and methods.

The CARMEN Portal allows sharing data and analytic methods. There are number of public services, which provide access to common neurophysiological processing functions such as data filters, neural spike detection and spike sorting. All analytic tools provided by CARMEN are accessible for registered users.

EEG Data Processor also provides executing analytic methods for registered users. In addition, this system is designed for sharing analytic methods by third-party applications. This approach is realized by remote procedure call. The EEG Data Processor provides a following set of features.

- Number of available threads
- List of available methods
- List of parameters of chosen method
- Running the selected method

A workflow can be interpreted as a set of methods; it has input(s), output(s), and computational processes inside. Therefore, presented approaches for sharing analytic methods can be used for sharing workflows.

### 6.9.2 Remote Procedure Call

Remote Procedure Call (RPC) is widely used for constructing distributed, client-server based applications. A client application calls a remote procedure (method), transfers data to a server application, and waits for a result. For web applications Web Services technology is used.

- **SOAP Based Web Services**

Web Services [27] use XML messages, HTTP protocol, and XML Namespaces for objects identification. Web Services include three cores:

- SOAP (Simple Objects Access Protocol)
- WSDL (Web Services Definition Language)
- UDDI (Universal Description Discovery and Integration)

The Simple Object Access Protocol (SOAP) is a lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment. SOAP-based requests and responses are combined with a transport protocol, such as HTTP [10].

SOAP has the following features [10]:

- Protocol independence
- Language independence
- Platform and operating system independence

- **REST Based Web Services**

A RESTful (Representational State Transfer) web API (also called a RESTful web service) is a web API implemented using HTTP and the principles of REST.

RESTful web API HTTP methods are following:

- GET - List the URIs and perhaps other details of the collection's members.
- PUT - Replace the entire collection with another collection. Replace the addressed member of the collection, or if it doesn't exist, create it.
- POST - Create a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.
- DELETE - Delete the entire collection.

REST defines a set of architectural principles by which you can design Web services that focus on a system's resources, including how resource states are addressed and transferred over HTTP by a wide range of clients written in different languages [43].

Both architectures are shown in Figure 35.

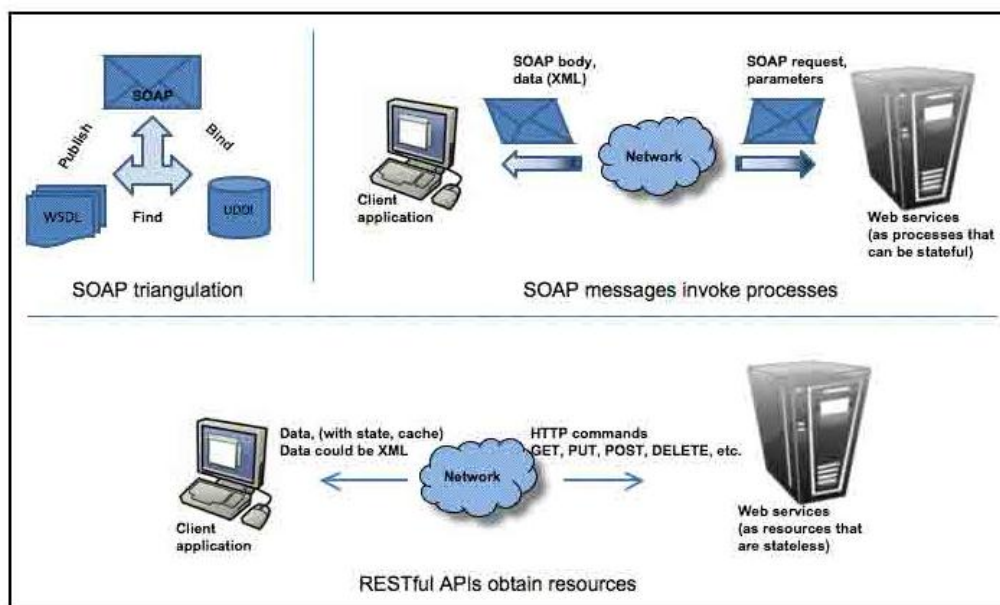


Figure 35: SOAP and RESTful architecture [44]

- **Cloud Computing**

Computational clouds are configured to provide service to end-users (clients) via high speed internet connections. Cloud components and basic cloud computing models are illustrated in Figure 36 [30].



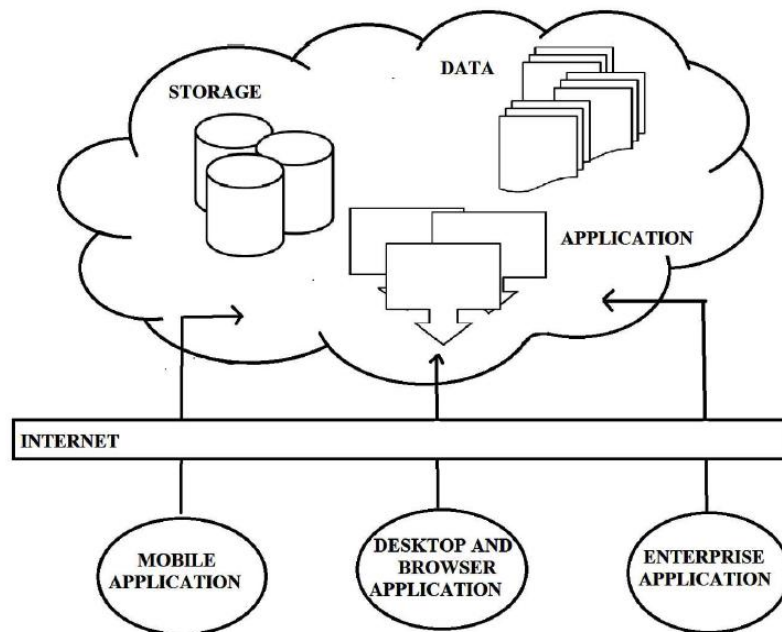


Figure 36: The basic model of cloud computing [30]

According to availability to the clients, clouds are divided into three types [30].

- **Public clouds** - These clouds are owned by third party. They usually provide services at lower cost using a pay-as-you-go manner. All of the services are maintained by the cloud providers. Amazon's Elastic Compute Cloud, IBM's Blue Cloud, and Google App Engine are some examples of public clouds
- **Private clouds** - These are clouds that are owned and operated by an enterprise solely for its own use. Data security and control are generally stronger than in public clouds. NASA's Nebula and Amazon's virtual private cloud (VPN) are private clouds.
- **Hybrid clouds** - These are combinations of both public and private clouds. The private cloud providers can use a third-party provider, either in partial or full manner, and provide the service to its enterprise.

Several scientific workflows engines such as e-Science Central are based on cloud. Apart from Web Service technology, clouds are also suitable for sharing data, analytic methods, and workflows.

## 7 Scope of Ph.D. Thesis

Since data processing often requires usage of more methods sequentially or in parallel, development of specific workflows is required. In this thesis, designing workflows in the domain of electrophysiological experiments is described. The described analytic methods are primarily used for analyzing EEG/ERP epochs and components. When modelling serial or series-parallel workflows, input and output parameters of methods given into a workflow have to be considered. Table 1 describes parameters of methods used in the domain of electrophysiological experiments.

Table 1: Parameters of method description

Method name	Input parameter(s) description	Output parameter(s) description
Detection of epochs	An EEG signal An array of values	A List of detected epochs (EEG signals) A two-dimensional array
Averaging	An EEG signal or a list of detected epochs An array/matrix of values	An ERP signal (averaged EEG) An array of values
Fourier transform	An EEG or ERP signal An array of values	Values of detected frequencies in signal
Wavelet transform	An ERP signal An array of values	Computed coefficients of translation, dilatation, and scale A two-dimensional array
Matching pursuit	An ERP signal An array of values	A list of selected atoms and their parameters A two-dimensional array
ICA	An EEG or ERP signal, correlation of signals An array/matrix of values	A list of decomposed components A two-dimensional array
Hilbert-Huang transform	An EEG or ERP signal An array of values	A set of decomposed Intrinsic Mode Function A two-dimensional array

According to Table 1, various input/output formats lead to syntactic and semantic incompatibility of the methods given into a pipe. In Figures 37 and 38, the preprocessing and processing methods suitable for giving into a pipe are shown.

The Ph.D. thesis will provide models of workflows for the domain of electrophysiological experiments. Because of incompatibility of method's input/output formats, the syntactic rules and metadata (semantics) for analytic methods have to be defined. The well-defined description of methods including their parameters description will also ensure the semantic compatibility.

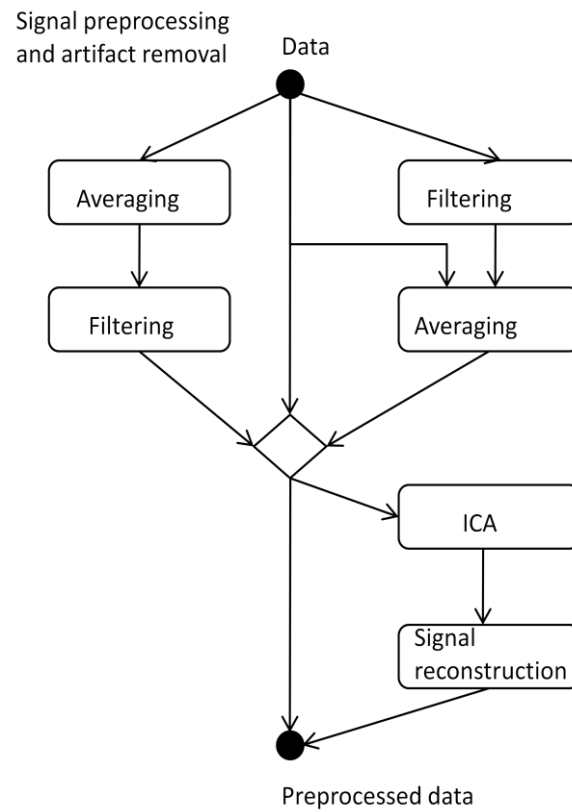


Figure 37: Signal preprocessing and artifact removal

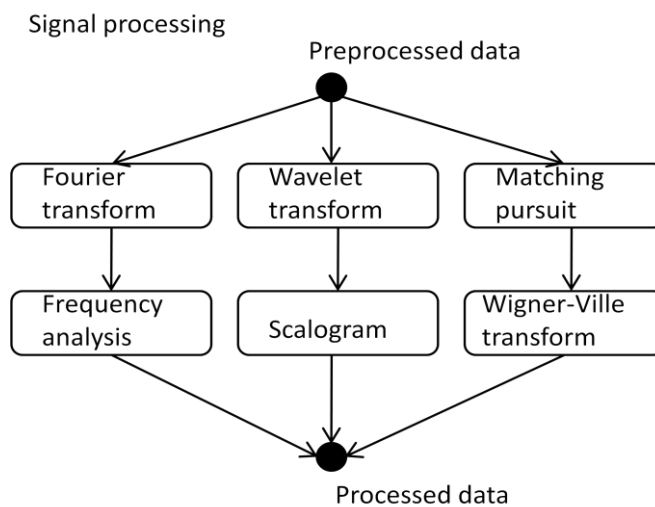


Figure 38: Signal processing

Since new and modified methods are developed, the model also allows adding new methods. The proposed model will be designed in a general way that the semantics of the model description will enable to add new methods easily. Defining a suitable structure of methods used in the workflow will be also the scope of Ph.D. thesis.

## 8 Conclusion

Several neuroinformatics infrastructures are being developed in parallel. Except from providing data storages and sharing data and metadata, they also provide opportunities to store and share analytic methods and their results. However, the models for workflows in neuroinformatics are still not satisfactorily designed and implemented, and there are still opportunities for innovation and improvement.

This work brought a summary of the current state of analytic methods in the domain of electrophysiological experiments. It also describes workflows, their modelling, and sharing. It evaluates existing engines for scientific workflows modelling.

In my Ph.D. thesis, I will focus on design of a processing model of analytic methods used in electrophysiology. These methods will be used in workflows. It includes defining metadata for analytic methods and design of a structure of workflow model.

The proposed solution will allow creating workflows for the domain of electrophysiological experiments. The solution also will allow adding new methods into workflows and sharing them. It will also solve the mentioned problem with incompatibility of parameters.

### 8.1 Aims of Ph.D. Thesis

- Design a processing model of analytic methods used in electrophysiology.
- Define a metadata for description of analytic methods and their organization.
- Identify/design workflows suitable for the domain of electrophysiological experiments.
- Verify the proposed workflows by implementation of a prototype and by integration of the prototype in suitable portal solutions.
- Test the solution on real electrophysiological data.

## References

- [1] N. Schaul, D. Kolesnik, D. Labar, P. Sethi, "Laptop artifact during electroencephalography" *The Internet Journal of Neuromonitoring* vol. 5, no 2, 2007.
- [2] J. N. Knight. Signal Fraction Analysis and Artifact Removal in EEG. Ph.D. Thesis, Colorado State University, Fort Collins, USA, 2003.
- [3] Luck, J. Steven, 2005, *An Introduction to the Event-Related Potential Technique*. Cambridge, Mass.: The MIT Press.
- [4] International Neuroinformatics Coordinating Facility [online]. <http://www.incf.org/about> [cit. 2013-03-15]
- [5] Jezek, P., Moucek, R. "Electroencephalography Data Processor" *HEALTHINF 2013 - International Conference on Health Informatics*, Barcelona, pages 357-361, 2013.
- [6] Workflow Diagrams [online] <http://www.kaskus.co.id/post/0000000000000000690620521> [cit. 2013-04-20]
- [7] P. Watson, H. Hiden, S. Woodman. "e-Science Central for CARMEN: Science as a Service." *Concurrency and Computation: Practice and Experience*, Volume 22, Issue 17, pages 2369-2380, 10 December, 2010.
- [8] Watson, P., Jackson, T., Pitsilis, G., Gibson, F., Austin, J., Fletcher, M., Liang, B., and Lord, P. "The CARMEN Neuroscience Server" *Proceedings of the UK e-Science All hands Meeting*, pages 135-141, 2007.
- [9] Event-Related Potential Technique [online]. <http://hiverndiscs.blogspot.com/2010/06/event-related-potential.html> [cit. 2013-03-13]
- [10] SOAP Web Services [online] [http://docs.oracle.com/cd/A97335\\_02/integrate.102/a90297/overview.htm#1010764](http://docs.oracle.com/cd/A97335_02/integrate.102/a90297/overview.htm#1010764) [cit. 2013-05-10]
- [11] Vareka, L. *Matching Pursuit for P300-based Brain Computer Interfaces*, Prague, 2012.
- [12] Quian, S. *Introduction to time-frequency and wavelet transforms* Paris, 2012.
- [13] Ciniburk, J., Mautner, P., Moucek, R., Rondik T. *ERP components detection using wavelet transform and matching pursuit algorithm* DCII, Prague, 2010
- [14] Transforms/Wavelets [online] <http://www.rfcafe.com/references/electrical/ew-radar-handbook/transforms-wavelets.htm> [cit. 2013-03-20]
- [15] Hyvärinen, A., Karhunen, J., and Oja, E. "Independent Component Analysis" *Adaptive and Learning Systems for Signal Processing, Communications and Control*. J. Wiley, 2001.
- [16] Rondik, T. "Methods for Detection of ERP Waveforms in BCI Systems" State of the Art and Concept of Ph.D. Thesis, Pilsen 2012.

- [17] S. Woodman, H. Hiden, P. Watson, J. Cała, “Workflows and Applications in e-Science Central”. *All Hands Meeting*, December 2009, Oxford, UK.
- [18] Workflow management system [online]  
<http://www.ceiton.com/CMS/EN/CEITON-CTWS-media-flyer-01.pdf>  
 [cit. 2013-04-11]
- [19] Karel Dudacek, Pavel Mautner, Roman Moucek, Jiri Novotny, “Odd-Ball Protocol Stimulator for Neuroinformatics Research” *Applied Electronics (AE)*, Pilsen, 2011.
- [20] Sarasvati [online]  
<https://code.google.com/p/sarasvati/> [cit. 2013-04-15]
- [21] Werkflow [online]  
<http://werkflow.codehaus.org/> [cit. 2013-04-15]
- [22] CARMEN System Introduction [online]  
[https://portal.carmen.org.uk/Content/carmen\\_portalintro.html](https://portal.carmen.org.uk/Content/carmen_portalintro.html) [cit. 2013-03-22]
- [23] Jezek, P., Moucek, R., “System for storage and management of EEG/ERP experiments – generation on ontology” *Databases and Information System Integration* vol. 1, Madeira: SciTePress, ICEIS 2010.
- [24] Weeks, M. *Writing Code for Carmen Service* [online]  
<http://www.youshare.ac.uk/wp-content/uploads/2012/04/WritingCodeforServices1.pdf>  
 [cit. 2013-03-25]
- [25] Ralph Meier, Martin P. Nawrot, Willi Schiegel, Tiziano Zito Andreas, V. M. Herz, “G-Node: An integrated tool-sharing platform to support cellular and systems neurophysiology in the age of global neuroinformatics”, *Neural Networks*, vol. 21, no. 8, pp. 1070-1075, 2008.
- [26] Waveform Analysis Using The Fourier Transform [online]  
<http://www.dataq.com/applicat/articles/an11.htm> [cit. 2013-04-20]
- [27] Jie Liu, Er-peng Zhang, Jin-fen Xiong, and Zhi-yong Lv *Deployment of Web Services for Enterprise Application Integration (EAI) System*, Berlin, 2006.
- [28] jBPM [online]  
<http://www.jboss.org/jbpm/> [cit. 2013-04-10]
- [29] Rondik, T., “Methods of ERP Signal Processing” Diploma Thesis, Pilsen 2010.
- [30] Radhe Shyam Thakur, Rajib Bandopadhyay, Bratati Chaudhary, Sourav Chatterjee “Now and next-generation sequencing techniques: future of sequence analysis using cloud computing” *Frontiers in Genetics* vol. 3, 2012
- [31] Sroka, J., Hidders, J., Missier, p., Carole Goble “A formal semantics for the Taverna 2 workflow model” *Journal of Computer and System Sciences*, pages 490-508, 2010
- [32] Development of a workflow system for the CARMEN Neuroscience Portal [online]  
<http://neuroinformatics2012.org/abstracts/development-of-a-workflow-system-for-the-carmen-neuroscience-portal> [cit. 2013-04-01]

- [33] Taverna [online]  
<http://www.taverna.org.uk/> [cit. 2013-04-03]
- [34] eScience Central [online]  
<http://www.esciencecentral.co.uk/?p=151> [cit. 2013-04-03]
- [35] Cinly Ooi et al., “CamBAfx: workflow design, implementation and application for neuroimaging” *Frontiers in Neuroinformatics*, 2009.
- [36] J.J. Vidal. Real-time detection of brain events in EEG. *Proceedings of the IEEE*, Volume 65, Issue 5, May 1977, pp. 633 - 641.
- [37] Brain Vision [online]  
<http://www.brainvision.com/> [cit. 2013-05-10]
- [38] Hibernate framework [online]  
<http://www.hibernate.org/> [cit. 2013-04-15]
- [39] Walls, C., 2007. *Spring in Action*. Spring Dallas User Group, Dallas.
- [40] Pandey, S., Voorsluys, W., Rahman, M., Buyya, R., Dobson, J. E., Chiu K., “A grid workflow environment for brain imaging analysis on distributed systems” *Concurrency and Computation: Practice and Experience*, volume 21, issue 16, pages 2118-2139, November 2009.
- [41] Huser, V., Rasmussen, L. V., Oberg, R., Starren, J. B. (2011). "Implementation of workflow engine technology to deliver basic clinical decision support functionality". *BMC Medical Research Methodology*, 2011.
- [42] Zimmermann, O., Doubrovski, V., Grundler, J., Hogg, K. “Service-oriented architecture and business process choreography in an order management scenario: rationale, concepts, lessons learned” *20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, USA 2005.
- [43] RESTful Web Services [online]  
<http://www.ibm.com/developerworks/webservices/library/ws-restful/> [cit. 2013-05-20]
- [44] SOAP and RESTful architecture [online]  
<http://4.bp.blogspot.com/-jP5Sev6xcYs/T1anGmPNTzI/AAAAAAAAAYo/MATXOYcY8xA/s1600/rest-v-SOAP-720.jpg> [cit. 2013-05-20]
- [45] Y. Renard, F. Lotte, G. Gibert, M. Congedo, E. Maby, V. Delannoy, O. Bertrand, A. Lécuyer, “OpenViBE: An Open-Source Software Platform to Design, Test and Use Brain-Computer Interfaces in Real and Virtual Environments”, *Presence: teleoperators and virtual environments*, vol. 19, no 1, 2010 (in press).
- [46] Rondik, T., “Použití matching pursuit s vlastním slovníkem funkcí při detekci ERP v EEG signálu (Using matching pursuit algorithm with its own dictionary for ERP in EEG signal detection)”. *Proceedings of the 10th Conference Kognice a umělý život (Cognition and Artificial Life)*, Opava: Slezská univerzita, pp. 329-332, 2010.
- [47] Ciniburk, J., “Hilbert-Huang transform for ERP detection“, Ph.D. Thesis, University of West Bohemia, Pilsen, Czech Republic, 2011.

## List of Figures

Figure 1: An example of electroencephalogram [1] .....	5
Figure 2: Examples of artifacts [2] .....	6
Figure 3: Well known ERP waveforms [3] .....	6
Figure 4: EEG/ERP Portal .....	7
Figure 5: EEG Data Processor frontend .....	8
Figure 6: CARMEN System [22] .....	9
Figure 7: An example of an EEG signal with detected epochs [16] .....	10
Figure 8: Averaging epochs [16] .....	10
Figure 9: Averaging technique [9] .....	11
Figure 10: An example of Fourier transform [26] .....	12
Figure 11: ERP signal containing P3 [46] .....	13
Figure 12: Gabor atom which best approximates P3 components [46] .....	13
Figure 13: Some well-known wavelets [14] .....	14
Figure 14: Input signal and its scalogram [16].....	14
Figure 15: Independent sources A and B .....	15
Figure 16: Mixture of sources ( $A - 2B$ ) .....	15
Figure 17: O and Q stimulus [29].....	16
Figure 18: Arrangement of experiment and connection of stimulator to ERP recording system [19] .....	17
Figure 19: The car simulator .....	17
Figure 20: Simple workflow .....	20
Figure 21: Workflow with a condition .....	20
Figure 22: Workflow with more conditions .....	20
Figure 23: Connection in workflow .....	20
Figure 24: Merge in workflow .....	21
Figure 25: An example of workflow [6] .....	21
Figure 26: Serial workflow .....	22
Figure 27: Parallel workflow .....	22



Figure 28: Series-Parallel workflow .....	23
Figure 29: jBPM architecture [28] .....	24
Figure 30: Workflow modelling in Taverna [31] .....	25
Figure 31: E-Science Central [17] .....	26
Figure 32: Workflow architecture [7] .....	26
Figure 33: OpenViBE designer for scenarios [45] .....	27
Figure 34: Structure of plug-in library .....	29
Figure 35: SOAP and RESTful architecture [44] .....	31
Figure 36: The basic model of cloud computing [30] .....	32
Figure 37: Signal preprocessing and artifact removal .....	34
Figure 38: Signal processing .....	34

## List of Abbreviations

AJAX – Asynchronous Java Script and XML

CARMEN – Code Analysis, Repository & Modelling for e-Neuroscience

CWT – Continuous Wavelet Transform

DWT – Discrete Wavelet Transform

EEG – Electroencephalography

ERP – Event Related Potential

FFT – Fast Fourier Transform

FT – Fourier Transform

GUI – Graphic User Interface

HTTP – Hypertext Transfer Protocol

ICA – Independent Component Analysis

INCF – International Neuroinformatics Coordinating Facility

JSP – Java Server Page

JSTL – JSP Standard Tag Library

MP – Matching Pursuit

MVC – Model – View – Controller

OS – Operating System

RPC – Remote Procedure Call

SEI – Service Endpoint Interface

SMTP – Simple Mail Transfer Protocol

SOAP – Simple object access protocol

SQL – Structure Query Language

UDDI – Universal Description Discovery and Integration

URL – Uniform Resource Locator

WS – Web Services

WSDL – Web Services Definition Language

WT – Wavelet Transform

XML – eXtensible Markup Language

## Appendix A – CARMEN's Analytic Tools

### Time Resolved Spectral Analysis:

It plots the time-resolved power spectra and coherence of two waveforms. Each figure has three plots. The first plot shows the waveform power (X-axis = time relative to the marker across, Y-axis= Frequency in Hz, and the color shows the power amplitude). The second plot shows the time course of the power at two frequencies (30Hz in red and 10Hz in black). The third plot shows power against frequency for two points in time (-2.5s relative to marker in black, -1.5s relative to marker in red). The Figure A1 shows these plots for the power in FileA, FileB, and the coherence between the waveforms in FileA and FileB.

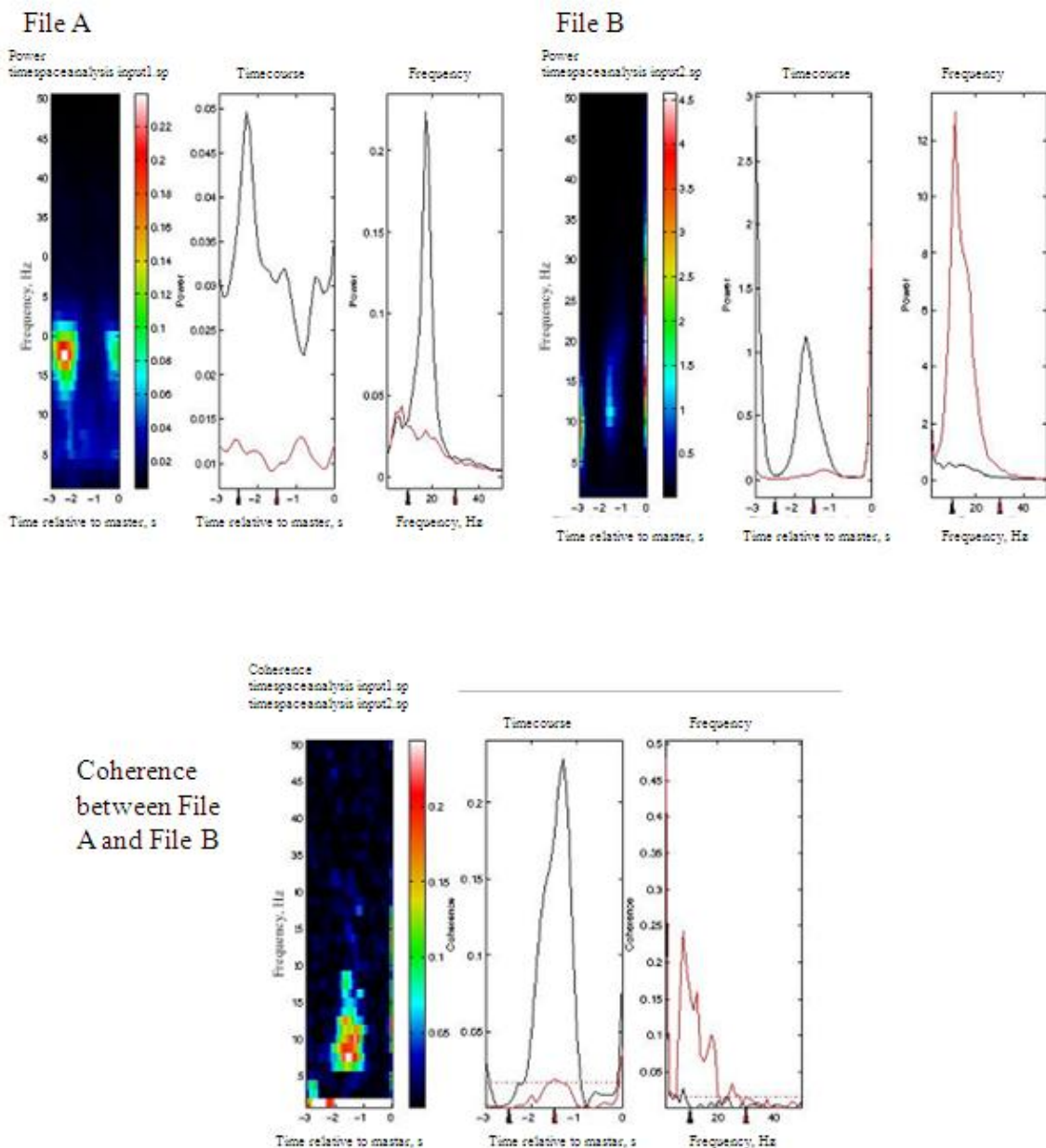


Figure A1: Time resolved spectral analysis

### Finding cells:

Finding cells: Finding extracellular single units in brief (secs) periods of recording. The algorithm detect the number of different single units (ncells) and how well separated they are (quality).

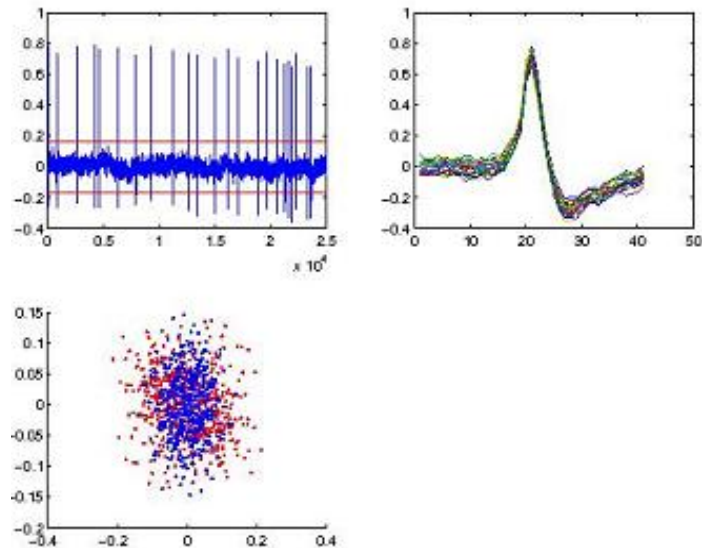


Figure A2: Finding cells

### Coherence Waveform Service:

Detects whether each file is a .sp or .spt, and calculates the power spectra and coherence plus phase appropriately for a waveform or event. Dashed line shows the significance level for coherence. The phase has been plotted three times ( $\pm 2\pi$ ) to make trends easily visible. Coherence should have confidence limits.

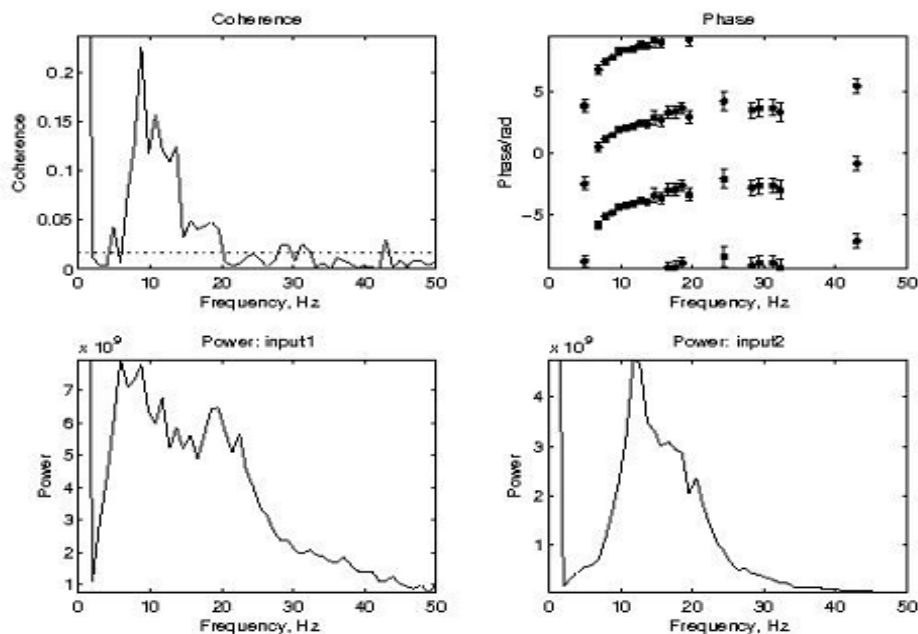


Figure A3: Coherence Waveform Service