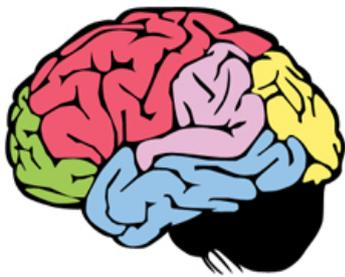




TECHNICAL DOCUMENTATION



IOnlyThink
A BRAIN-COMPUTER INTERFACE SYSTEM

September
2012

Technical documentation of I Only Think

I Only Think (IOT) is a direct neuronal interface which allows, with the help of an electro-encephalographic helmet (EEG headset), to realise some simple actions on a computer, using your thoughts. *IOT* is aimed to permit disabled people, or people which can't use the standard peripherals (mouse and keyboard), to use a computer and access to the numerical world.

TECHNICAL DOCUMENTATION OF I ONLY THINK

I – PROJECT PRESENTATION 2

 I.I – Definitions, context and history..... 2

 I.II – Specifications 2

 I.III – Prerequisite..... 3

 I.III a – Hardware configuration..... 3

 I.III b – Software configuration..... 3

 I.IV – Sources..... 4

 I.IV a – Sources download..... 4

 I.IV b – Sources documentation..... 4

II – GLOBAL DIAGRAM 5

III - DETAILS OF THE PROJECT 6

 III.I – The Graphical Environment Controller (*GEC* or *CEG*)..... 6

 III.II – The *Builder* 7

IV – COMMUNICATION DIAGRAM..... 8

V – CONCEPTION 9

 V.I – Technological choices 9

 V.II – UML / Object Diagram..... 9

FURTHER INFORMATION..... 10

 License 10

 Contact 10

 Authors 10

I – PROJECT PRESENTATION

I.1 – Definitions, context and history

I Only Think is a project developed since 2009 as an EIP (EPITECH Innovative Project).

An EIP is an internal project at the School of Computing EPITECH. This is a two-year project launched in the last two years of the course (fourth and fifth years). It is the realization of the work done during the five years of studies in this school.

An EIP is a concrete project, established on all sides by a single group of students and followed by a dedicated teaching team.

IOT is a neural interface for using an EEG headset to perform simple actions directly on computer by thought. It was first created by a group of students in 2009. It was at this time developed in partnership with the INRIA (National Institute for Research in Computer Science and Control, in France), the parent institute of the OpenViBE project.

OpenViBE is a software solution to design, test and use of brain-machine interfaces. It allows, as part of *IOT*, to interface different types of EEG headsets ensuring full interoperability with those compatible OpenViBE.

An EEG headset is an electrode system receiving certain signals from the brain activity. These analog signals are then amplified and converted into digital signals which can be interpreted by a computer using an acquisition device.

In 2011, the project was taken over by a second group of students, also as part of an EIP at EPITECH until mid-2013.

I.2 – Specifications

IOT uses an electroencephalographic helmet (EEG helmet) to analyse and interpret the thoughts of a user and a camera (or webcam) to emulate a cursor via a principle of eye-tracking (tracking eye movement). Both technologies can be used independently or simultaneously.

The project is based on OpenViBE for interpreting signals from the EEG helmet, allowing full compatibility with helmets supported by OpenViBE.

The project is divided in two parts:

- The *Builder*: A simple tool for performing ergonomic layers customized for any program. These layers allow you to choose selectable and clickable areas of each application.
- The *CEG* (or *GEC*, Graphical Environment Controller): This is the core of the project to navigate using the EEG helmet and / or eye-tracking. The CEG is able to load the layers generated using the Builder and then associate them with corresponding applications or programs. The CEG also has a standard layer used with any programs.

I.III – Prerequisite

I.III a – Hardware configuration

We recommend the following hardware configuration to run *IOT*:

- An EEG helmet compatible with OpenViBE (list of compatibility: <http://openvibe.inria.fr/supported-hardware/>)
- A video-camera or webcam high resolution (to use the eye-tracking solution)
- A computer with a dual-core processor is highly recommended (mandatory for the eye-tracking solution)

I.III b – Software configuration

We recommend the following software configuration to run *IOT*:

- The OpenViBE software (download at: <http://openvibe.inria.fr/downloads/>)
- A Microsoft Windows 7 (32 bits or 64 bits) or Linux (tested on Ubuntu and ArchLinux) operating system

I.IV – Sources

I.IV a – Sources download

The sources of IOT are available as an archive in various formats on our web page inside the "Download" part.

You can download them to compile, read, or modify to your liking.

The installation instructions for these sources and a list of external dependencies are subject to an external documentation available within the archive or on our web page, inside the "Documentation" part.

MD5 hash of the content control are generated each publication to ensure the users safety. We encourage you to check any files downloaded using the hash control.

I.IV b – Sources documentation

A Doxygen documentation is available to simplify the understanding of the sources.

You will find this documentation on our web page inside the "Documentation" part, then "Documentation of sources".

II – GLOBAL DIAGRAM



The electroencephalographic helmet (EEG headset), consisting of a helmet with electrodes and sensors, coupled to an acquisition device is missioned to convert and transmit brain waves into digital signals understood by the server acquisition.

The acquisition server is a software component that retrieves those signals, converts them into usable data and transmits then to the OpenViBE part.

The OpenViBE part is a software component which analyses and process information. The processed information is transformed as an event understood by the CEG and is passed to it.

The CEG is a software component installed on the machine to be controlled. It applies to the operating system the appropriate action.

Relating to this component, the Builder will permit you to personally select areas of the screen for an easy navigation on the software interface.

The eye-tracking part will fit with the CEG to make the navigation much more fluid and ergonomic for users wishing it.

III - DETAILS OF THE PROJECT

III.1 – The Graphical Environment Controller (GEC or CEG)

The CEG is the main part of the project; it allows you to control an operating system using EEG headset and / or a camera or webcam. Its operation is based on the graphic library Qt 4 (see the list of external dependencies for more information) and is developed in C++.

Its principle is very simple: superimpose a generic layer (configurable grid-size) to the screen to allow the user, using very simple actions (originally "Right", "Down" and "Action"), to navigate on a computer. It also has different menus for the implementation of some secondary actions (see the user documentation for more information about using the CEG).

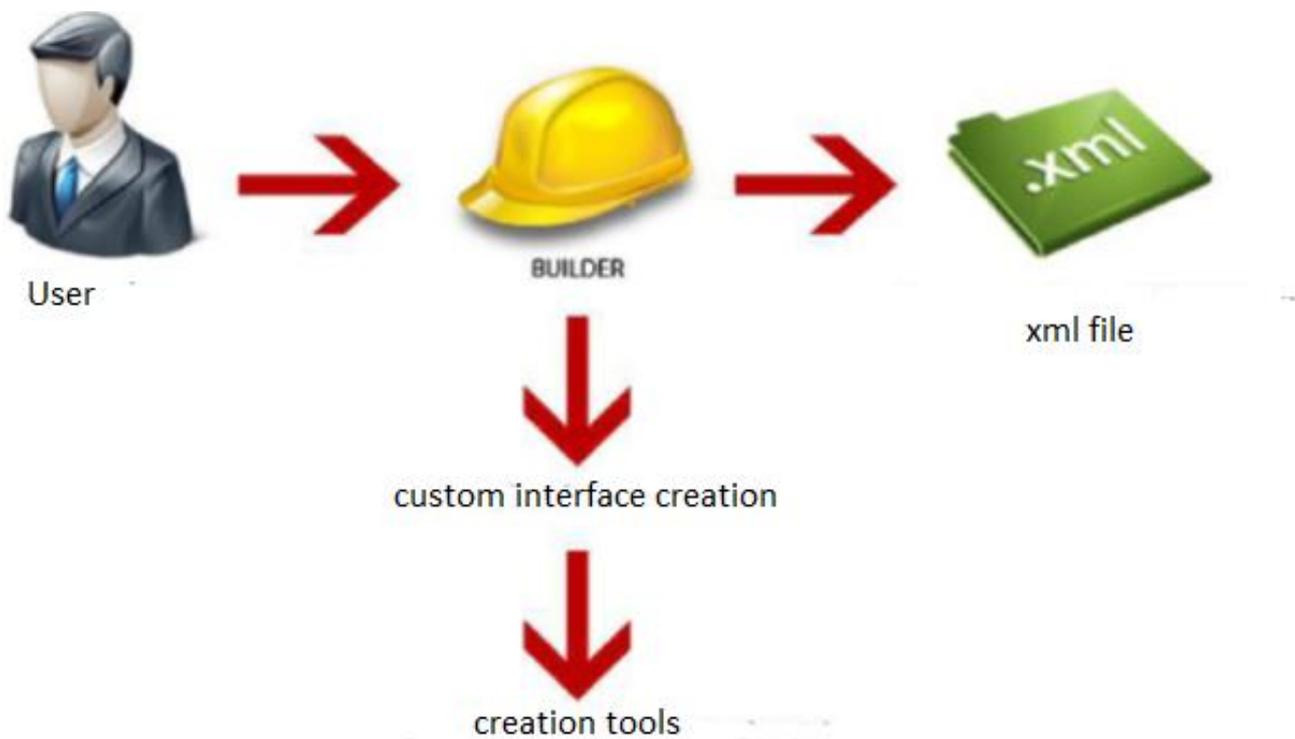
It recovers various actions sent by the EEG headset via the VRPN (Virtual Reality Peripheral Network) protocol managed by OpenViBE. This protocol can be used via the VRPN library (see the list of external dependencies for more information).

It allows you to make the connection between what you think (interpreted via the EEG headset) and what we see (layer superimposed on the screen) involving OpenViBE events emitted by the various actions described above.

Then the eye-tracking part provides more flexibility by allowing the user if they wish to replace one or more actions by eye-tracking (movement or blinking). The OpenCV library (see the list of external dependencies for more information) is used for the acquisition and interpretation of images transmitted by the camera or webcam.

Finally, through use of the Builder, the generic layer is replaced by a set of pre-defined areas in the *Builder* and stored as an XML file.

III.II – The *Builder*



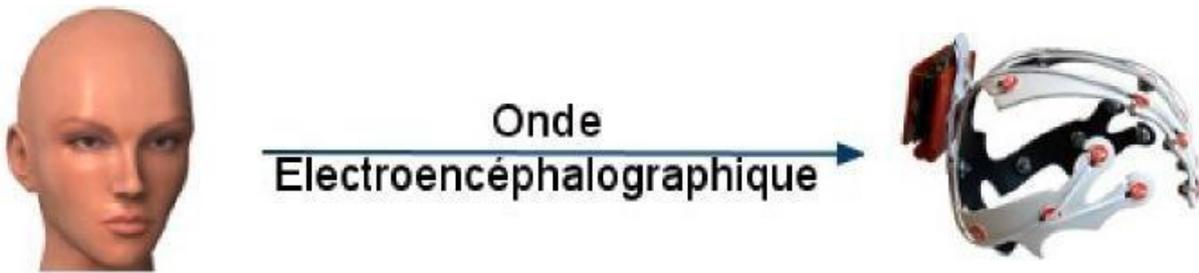
The *Builder*, meanwhile, consists of two parts: the selection part and the zone part. It allows you to load and save some sets of zones as XML files. For more information on how to use it, please refer to the user documentation.

The selection part allows, for a valid person, to resize, edit, copy, cut or paste clickable areas of the application layer to replace the generic form of grid.

The zone part allows you to edit the properties of an area and change related actions, such as clicking, zooming, or any other possible action on the *CEG*.

The *Builder* is a fairly simple program developed in Python and based on the graphic library PyQt (see the list of external dependencies for more information).

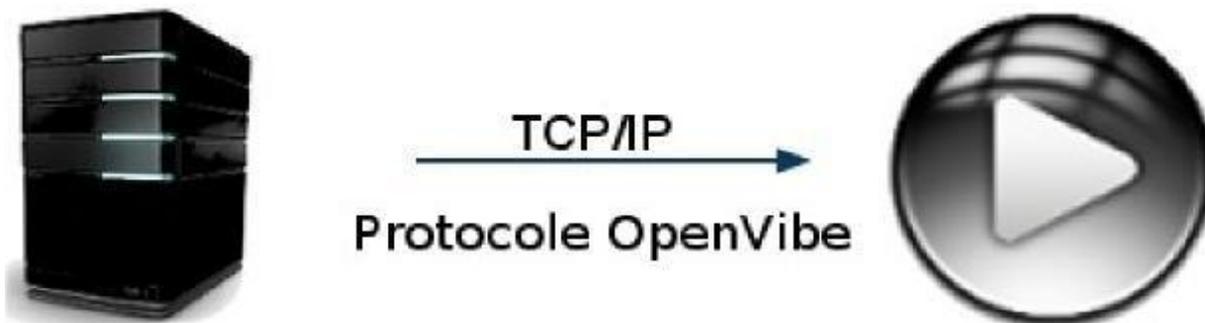
IV – COMMUNICATION DIAGRAM



The brain signals are recovered as electroencephalographic waves emitted by the brain and converted into digital signals.



The acquisition device amplifies and converts the digital signals captured by the EEG headset. Then, it transmits them to the acquisition server.



The acquisition server translates the received information for a further transmission over TCP/IP using the OpenViBE protocol to the OpenViBE part.

V – CONCEPTION

V.I – Technological choices

IOT is based first and foremost on the use of two development languages, C++ for the *CEG* and Python for the *Builder*.

The choice of using C++ for the *CEG* was originally made to facilitate interoperability with the acquisition software, OpenViBE. Since the integration of the VRPN protocol within OpenViBE, the development of *CEG* could have been made in any language, but the strength and stability of C++ convinces us to keep it. In addition, the external libraries used (Qt, OpenCV and VRPN) are compatible with many operating systems, as well as the code of *CEG* itself since its inception. There was therefore no interest to change the technologies used, even if it would be possible.

About external libraries of the *CEG*, each has its own interest. For the graphics, Qt is used because it is a portable library (compatible with most existing operating systems), easy to use, with a great community and perfectly suited to C++. The VRPN library is itself used because it is again a portable library, easy to use and allows some transmissions with the OpenViBE part. Finally, OpenCV is a library for the acquisition and processing of images and videos. This library is one of the most successful in the Computer Vision domain and has a very large community. In conclusion, all these libraries are portable, easy to use, supported on a long term and free for a non-commercial use.

The choice of using Python for the *Builder* is primarily done to ensure the development to be fast and easy. In fact, Python is a portable language, ergonomic, easy to use and perfectly suited to our application. Coupled to the external library PyQt (Qt adapted to Python), it is easy to retrieve complete information about the graphical user environment.

Finally, the OpenViBE software can be considered as one of the most important technological choice within our project. Indeed, it ensures the acquisition and transmission of signals picked up by the EEG headset and therefore the thoughts of the user. It allows full interoperability with all EEG headsets compatible with OpenViBE. It also allows the simulation of scenarios in order to simplify the testing and development without necessarily using an EEG headset. OpenViBE is developed and supported by the INRIA, has a very large community, a long-term support, is open-source and meets our expectations.

V.II – UML / Object Diagram

The UML conception (object diagram) is subject to an external documentation, downloadable from our Web page inside the "Documentation", then "UML – Object diagram".

FURTHER INFORMATION

License

IOT is an open-source project under the GNU/GPL license. The complete terms of use of the GNU/GPL license are available on the following address: <http://www.gnu.org/licenses/gpl.html>.

Contact

Please visit our Web page for any further information: <http://eip.epitech.eu/2013/iot/>.

You can contact us directly at any time via the contact form on our website or using the following mail address: iot_2013@labeip.epitech.eu.

Authors

Thibault BRONCHAIN
Nicolas NIRLO
Maxime NOUGAREDE
Antoine RAPPINET
Nicolas ROCHELLE
Julien SADDOUR
Jeremy VIENNE
Charles YVANES