Towards Standardized User and Application Interfaces for the Brain Computer Interface

Paul McCullagh, Melanie Ware, Alex McRoberts,
Gaye Lightbody, Maurice Mulvenna, Gerry McAllister,
José Luis González, and Vicente Cruz Medina

School of Computing and Mathematics, University of Ulster, Northern Ireland, UK
Telefonica Investigacion y Desarrollo, Parque Tecnologico de Boecillo, 47151 Boecillo, Spain
{pj.mccullagh,mp. ware,a.mcroberts,g.lightbody,md.mulvenna,
hg.mccallister}@ulster.ac.uk

Abstract. In this paper, we consider two obstacles preventing widespread deployment of Brain Computer Interface (BCI) technology: the lack of standardization for the user interface and the applications interface. We suggest a structure for an intuitive graphical user interface (IGUI) and propose a methodology for usability testing. A universal application interface (UAI) based on Universal Plug and Play and deployed by Open Services Gateway initiative is proposed. This issues user commands and receives device status; and communicates to the IGUI using an eXtensible Mark-up Language file containing menu definitions. Using this approach we have achieved control of simple domestic devices, using 'plug and play' technology, interaction with a set top box and media player for entertainment applications.

Keywords: Brain computer interface, graphical user interface, accessibility, universal application interface, applications.

1 Introduction

For individuals with a neuro-muscular degenerative disorder, the need for a degree of control is paramount in their lives. Neumann and Kubler [1] commented on the distress of people who could no longer participate in a BCI research programme, therefore losing the potential to communicate and interact with their environment. Much of the BCI research to date has been for a communication channel for people with significant physical disabilities, where there are limited usable assistive technologies available. Here there is a significant dilemma, as in such circumstances BCI technology is relevant to only a very limited patient population. Schalk *et al* [2] commented on this aspect:

"*BCI research up to the present has consisted mainly of demonstrations that certain brain signals recorded and measured in a certain way, and translated into control commands by a certain algorithm, can control a certain device for one or a few users*."

Systems lack flexibility and hence have few areas of application. Can we envisage a scenario where BCI can become a 'plug and sense' technology linking with applications that can be deployed as 'plug and play'? What challenges face BCI technology

C. Stephanidis (Ed.): Universal Access in HCI, Part II, HCII 2011, LNCS 6766, pp. 573–582, 2011.

in becoming a more competitive choice of assistive technology? Table 1 illustrates current challenges in BCI, as suggested by Allison [3].

Table 1. Challenges to the deployment of BCI

BCI Challenges	**Discussion**
Recording: electrodes, use of an EEG cap, amplifier and computer	Setup requires about 20 minutes by an electrophysiological technician. The dependence on expert assistance could be reduced by more accessible recording technology.
Paradigm and parameter selection suited to the individual	Automatic tools are needed to identify the best BCI approach and parameters for each subject and develop a customized BCI accordingly.
Signal Processing	Tools are needed to identify and extract relevant features, and set operating parameters. This could further improve reliability and usability in noisy settings.
User Interface	BCIs use a simple, conventional interface that is identical for all users. An intuitive user interface is required that is much easier to learn and use, account for assistive design principles, and be customized for user.
Application Interface	BCIs are developed around only one application. Communication and entertainment tools will further enhance flexibility and usability by offering access to a much wider variety of applications
Cost	BCI hardware currently costs over $15,000 which inhibits wide scale deployment.

Within BRAIN [4], research is ongoing to address these scientific and technical challenges, which could potentially lower cost, enhance operation and promote wider uptake. In this paper, we restrict our discussion to challenges which aim to provide some degree of standardization for the User Interface and Application Interface.

In section 2, we outline the architecture of the Intuitive Graphical User Interface (IGUI) developed for BCI, and provide a human computer interaction (HCI) accessibility testing methodology. In Section 3, we provide a description of the technical details for implementing an application interface based on current standards, referred to as the Universal Application Interface (UAI). Figure 1 gives a visual overview of the interaction between the BCI, IGUI, UAI and applications. It illustrates a modular architecture in which BCI2000 provides the BCI interface supplying Universal Data Packets (UDP), the content of which can be specified by the signal processing being performed. The architecture given can support different BCI technologies and is depicted in operation with Steady State Visually Evoked Potentials (SSVEP) [5]. Light emitting diodes placed around the screen provide the visual stimulus for the protocol. Section 4 summarizes the communication between IGUI, UAI and the devices and illustrates some initial applications. In Section 5 we draw some tentative conclusions on wider application.

2 IGUI Interface Design

The intuitive graphical user interface (IGUI) offers a device control mechanism to the user that can support multiple BCI paradigms, namely SSVEP and the intended movement paradigm. It is designed to provide a framework, similar to Mason *et al* [6]. The interface acts upon a menu definition contained in an XML file which lists available domotic devices and their associated menu structures. This interface operates in conjunction with BCI actuated peripherals (see Figure 1). It is applicable to all devices in the device controller module, called the Universal Application Interface (UAI). The IGUI is also capable of handling modifications in display or operation according to user defined preferences. The UAI acts as wrapper for multiple device interaction protocols; it provides a single control interface to the IGUI, hiding the complexity of interaction.

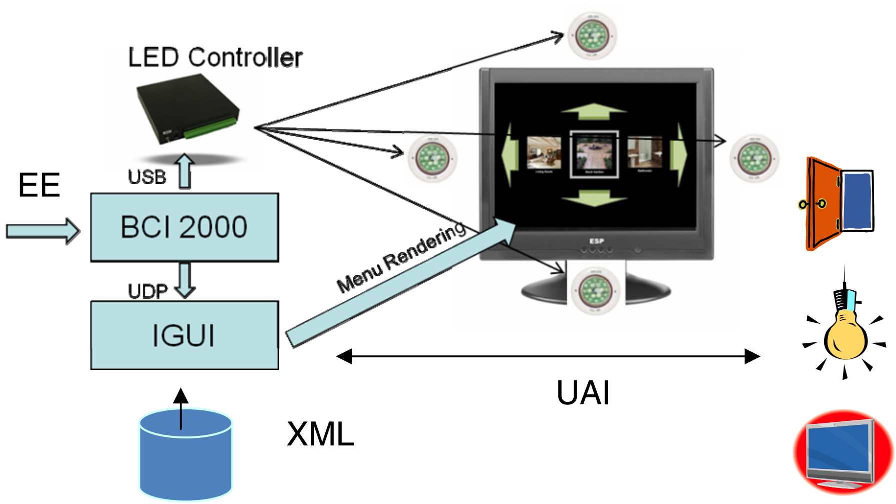

Fig. 1. System for SSVEP mediated BCI control showing Intelligent Graphical User Interface and Universal Application Interface components

BCI suffers from very low data communication rates. This presents a problem with regard to usability and hence acceptability. In Figure 1 a 4 way command choice (up, down, right, left) is illustrated but the user interface can support a 2 or 3 way command structure if this is more suitable to the chosen BCI paradigm and the user.

Navigation of a menu structure using a 4 way command is potentially faster than for 3 or 2 commands due to the reduction in overall steps to perform a task. However a paradox presents; errors may increase with the number of command choices provided by the interface if it becomes more difficult to differentiate the user's intention via the BCI. Furthermore, the cognitive load to the user must also be considered and for some users a more simplistic yet less powerful interface may provide them with a more robust and usable system overall. Finding the correct balance of interface complexity and system robustness is a multidimensional and multidisciplinary problem.

An accessibility test was designed to determine the level of tolerated accuracy in the command interface, for a user when using a four-way command interface. A ‘Wizard of Oz’ observation technique [7] was used. The testing determines: a) minimum desirable level of accuracy in a command mechanism; b) acceptable level of accuracy in a command interface when used by a motivated and informed user. The BCI accessibility assessment uses a mouse-controlled interface to provide input to the IGUI. The user is asked to navigate, for example, ‘*Go to the living room and turn on the light*’. The tool issues incorrect user commands to the IGUI on an increasing scale, resulting in incorrect navigation and feedback. Initially the user is unaware that inaccuracies have been generated. The test measures ability to tolerate inaccuracies prior to disengagement. The user is asked to repeat the test on a number of days, tolerating and accommodating the inaccurate interface until they judge that any meaningful navigation becomes impossible. Ware *et al* [8] have tested the approach with three healthy volunteer users. They found for these users that desirable accuracy and acceptable accuracy were greater than or equal to 77%. Users were subsequently asked to use the interface (Figure 1 with SSVEP BCI interaction), and were able to achieve accuracy in excess of 80%. This small initial test indicates that a figure of merit for the IGUI can be obtained, and reassuringly shows that the interface designed can accommodate that degree of user interaction. As there are many parameters that may be varied, it is useful to have a user acceptance figure. For example, IGUI was operated using the SSVEP paradigm with four selected frequencies. For a set menu navigation task lasting 4 minutes a single healthy user demonstrated accuracies in excess of 90%, with an optimal choice of mid range stimulation (26, 28, 30, 32Hz). This can be envisaged as a method used to assess appropriate operating parameters for a user.

3 UAI Interface Design

The main objective of the Universal Application Interface is to make various applications available to the user that can be controlled through the IGUI. The UAI is also responsible for the execution of the commands. It has two key principles:

1. All applications must implement the same Java interface, a single method.
2. All devices are controlled in a uniform way by using the standard UPnP protocol [9]. Non UPnP devices are controlled via a UPnP wrapper; a ‘proxy’ component that provides a UPnP interface while encapsulating the device’s native Application Programmer Interface (API).

The UAI has been implemented using the Open Services Gateway initiative (OSGi) framework [10], which conditions the interaction of the UAI with other parts of the system, and overcomes the limitations of the Java language facilitating a component oriented system. A component is a piece of code that provides some functionality through a well defined interface. The coupling between components must be minimal to allow for easy installation, update and removal. Applications and device drivers are easily managed, implemented as OSGi bundles that can be dynamically installed, started and stopped without the need of restarting the whole UAI. Component management and monitoring can be done remotely, which is very convenient when dealing with community based installations. OSGi provides standard mechanisms for the discovery and control of UPnP devices. Development of UPnP based applications are facilitated by available libraries. OSGi is designed for use in embedded environments. This means that the UAI could be installed, for example, in a residential gateway, away from the rest of modules of the components (BCI, IGUI). Communication between the UAI and the rest of system is achieved through the local home network. This segregation provides the advantage of removing the computing load of the UAI from the CPU dedicated to the BCI and user interface, thereby enhancing portability and battery life of the mobile system. Furthermore, it represents the system independence of the BCI from the application technology. De-coupling the IGUI from the UAI means that the each element can be tested separately, before integration testing. To support this, the UAI – IGUI interface is based on Web Services (WS). UPnP technology allows for a further degree of physical system distribution as devices and control points can communicate through the local network, Figure 2.

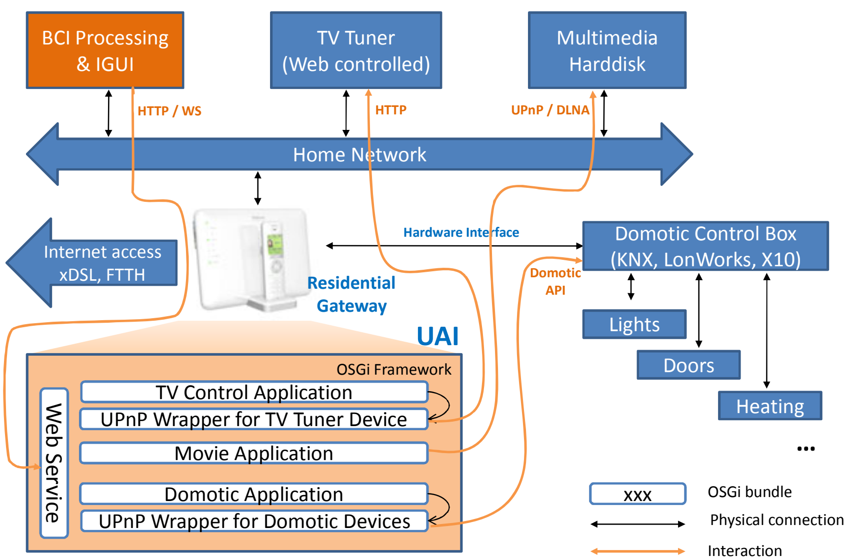

Fig 2. Architecture illustrating interaction of the IGUI with the Universal Application Interface

Three applications have been installed:

1. The domotic control application implements commands needed to operate elements such as lights, doors, windows, and heating. Since the domotic system is not based on UPnP, a wrapper is included in the UAI. The wrapper interacts with the domotic control box through its proprietary API and with the application through UPnP.
2. The TV control application allows the user to change TV channels and control the volume. Even though the TV tuner is IP-enabled, its control interface is not UPnP compliant. This means that a UPnP wrapper is again needed. In this case, no proprie-

tary API and hardware interface are needed, as the TV tuner can be accessed through the home network and controlled by HTTP.
3. The movie application allows the user to select a movie from his/her collection and play it on the TV. Since the multimedia hard disk is UPnP AV/DLNA (Digital Living Network Alliance standard [11] compliant, no wrapper is needed and the application can interact directly with the device.

The implementation of the OSGi framework that has been selected for the UAI is Equinox 3.4.3 [12]. The Eclipse Integrated Development Environment (IDE) is based on OSGi itself, which makes it the ideal tool for developing and testing OSGi applications.

4 IGUI – UAI – Device Communication

UAI applications can be invoked from the IGUI in a uniform way through the provided Web Service named `UAIDispatcherService` and exports only one operation named `raiseCommand`, which accepts three input parameters (device identifier, command identifier, label associated to the IGUI menu) of type string and returns an integer (0: Success, -1: UAI server not found, -2: Unknown application, -3: Unknown command, -4: Unknown device, -5: Unavailable device, -6: Application error). When the web service receives the command invocation from the IGUI it needs to find the application which is able to execute it. For this purpose the UAI's Device Manager maintains a mapping between devices and controlling applications. The web service then dispatches the command to the corresponding application for execution.

The method `executeCommand` is used by the web service to dispatch the command to the application. UAI applications show the following sequence: *Validate the command received from the IGUI, Access the relevant UPnP service on the device, Execute the command by invoking the corresponding UPnP action, Return the result code.* Some applications need to maintain status information between command executions. In this case the pattern is more complex than the one shown above.
Other applications need to subscribe to UPnP events from the controlled devices. OSGi provides the means to register the subscription. The application then needs to implement a listener interface by including a callback method that it is executed when an UPnP event from the device is received.

The OSGi specification includes support for UPnP in the form of a number of interfaces that represent the different elements of the UPnP specification: devices, services, actions, state variables and events. The OSGi framework automatically discovers all UPnP devices connected to the network. They are registered as regular OSGi services that applications can access in the usual way. When a new device implementation is registered, the OSGi automatically publishes its presence in the network. This makes it possible for the UAI applications to control both native UPnP devices and wrappers in a uniform way. If the device becomes unavailable, the wrapper is unregistered.

To facilitate development UAI applications also include a device emulator, which from the application's perspective, is not distinguishable from the hardware device or the wrapper. This aids development and testing within the consortium.

Figure 3(a) shows an application for controlling a domestic light. Figure 4 shows the BCI system in Figure 1 used to control a simple ‘etch-a-sketch’ drawing application. It is based on the original game, where two thumbwheels are turned to guide the direction of the stylus on the screen, one wheel guiding the stylus left and right, while the other wheel guides the stylus up and down. The control is through SSVEP with 4 commands (left, right, up and down). Pausing the drawing is achieved when the user accurately classifies 3 commands of the form [Left, Right, Left], [Right, Left, Right], [Up, Down, Up] or [Down, Up, Down]. The application has been tuned to work with BCI2000 slowing down interaction to a practical drawing speed without which the interaction would be unusable. This illustrates usability factors that must be considered with BCI, and the need to embed some autonomy in the application.

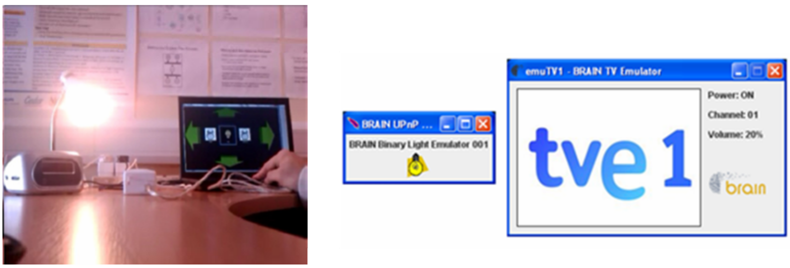

Fig. 3. (a) control of a domestic light, (b) light and TV emulator

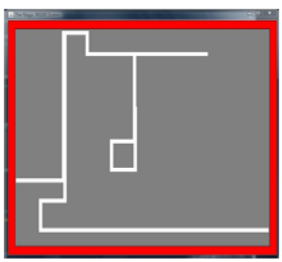

Fig. 4. etch-a-sketch drawing application

An important component of the UAI is the Device Manager (DM). It keeps the details of all devices under control of the BRAIN system and tracks their status. The Device Manager reads the static device information from an XML configuration file. Each UPnP device, wrapper and emulator to be controlled by the UAI must be included in the configuration file.

In the sample device configuration file we can see that two devices have been registered into the UAI: an X10 controlled light and a Dreambox TV tuner [13]. This is the information contained:

- `deviceId`: identifies the device within the BRAIN system. These identifiers are shared between the IGUI and the UAI.
- `upnpstring`: UPnP identifier used by the Device Manager to recognize the device from all discovered UPnP devices. It can be either the unique device identi-

fier or the service identifier. In the case of the Dreambox, a wrapper instance is created for every tuner installed at home; therefore the unique device identifier is used to recognize it. However, in the case of the X10 light, the selected X10 controller uses a random unique device identifier so it is not known in advance. The service identifier is used instead.
- `appId`: qualified name of the class implementing the controlling application. The Device Manager builds the device – application map from this information.
- `locationId`: identifies where the device is physically located. Not used at the moment.
- `ownerId`: identifies the user who is the owner of the device. Not used at the moment but can be used in the future for granting device control privileges to guest users.
- `properties`: contains device specific configuration parameters. These parameters can be used either by the applications or the wrappers.

When the DM is started, it reads the device configuration file and waits for UPnP events. When a device is discovered, the DM tries to recognize it by matching the configured UPnP string with the identifiers retrieved from the device. If a match is found, the device is marked as active and its status begins to be tracked by the DM. If no match is found, the device is ignored. When a tracked device becomes unavailable, the DM marks it as inactive. The DM provides a number of methods to allow other UAI components to access both the static and dynamic device information:

- `getDeviceByDeviceId`: returns all device information for a particular deviceId.
- `getDevicesByApp`: returns the list of devices under control of a particular application.

5 Discussion and Conclusions

So how far can BCIs develop and what will be the time frame? In the BRAIN project, with reflection to Table 1, we have made advances in the following areas:

Recording has been improved with the introduction of a small portable amplifier (TMSi Porti), which can use fibre optic cable or Bluetooth to transmit EEG signals to the computer for digitization. This should increase the acceptability of the BCI equipment. Further miniaturization is possible and in progress. Significant progress has also been made with regard to the testing of water based electrodes, which removes the need for conductive gel [14].

The SSVEP paradigm has been extended to higher frequencies, above 30Hz stimulation rates [5]. These rates are more comfortable to the user, but are associated with a lower signal to noise ratio. Hence new signal processing routines have been used to extract the features associated with High Frequency-SSVEP (HF-SSVEP). Acceptable Receiver Operating Characteristic (ROC) rates of 80% have been achieved. The imagined movement paradigm (ERD/ERS) [15] is currently under investigation. There is a requirement for training and concentration and attention has to be maintained over a longer period of time. Progress has been made with personalizing the

BCI, by the use of automated calibration using a 'wizard'. This attempts to determine the best stimulation frequencies and best EEG locations for determining the four distinctive commands. Similar techniques are required to produce ERD/ERS command classification – in this case only 3 (left hand, right hand, feet) with a suitably modified menu. However the stability and repeatability of the calibration is still under investigation. This may mean that a calibration session is required before each use to account for factors such as arousal, fatigue, time of day, and environmental conditions. Unless operational parameters can be stabilized the length of recording sessions are increased considerably and this mitigates against acceptance.

However there is still a clear gap between demonstrating a software package working in one laboratory for trial subjects and transferring that knowledge to other laboratories, given the plethora of user and software parameters, differences in equipment, operating environment. Significant planning is required to allow testing, debugging and roll-out. For example, achieving a reliable and robust 4-way decision in the SSVEP is proving elusive, particularly at higher frequencies (HF-SSVEP), where the visual response has a lower signal to noise ratio. This is the case in the laboratory setting with enthusiastic motivated subjects. This information allows us to consider the option to use of the GUI with a 2 or 3 way decision process.

In terms of the user and application interface a modular architecture has been developed which will support a range of BCI paradigms and characteristics which allows for user customization. For any assistive interface much effort is required by health care professionals and the HCI developers to produce a system that is targeted to the individual needs and cognitive ability of the user. The IGUI has been developed to support this customization, enabling the visual representation on the screen to be altered or images/ photos of locations and devices to be updated to the most suitable for that person. The separation of the BCI system from the devices through control of the IGUI and UAI enables the development of a BCI system that is independent physically to the applications. This creates portability and the IGUI operation has been shown to support the openBCI platform [16]. The importance of such portability has been demonstrated by OpenViBE [17]. The UAI and supporting infrastructure through Web Services sets the foundation for multi-application support.

Millán *et al* [18] provides a state of the art review of BCI technology and applications. Few applications are publically available outside the laboratory. An exception to this is g.tec's Intendix [19], which has been released as bringing "BCI technology into patients' everyday life'. It is a P300 based personal EEG system, allowing a user to spell out text. This widespread deployment of a BCI technology is certainly a worthy aim, however, as this paper presents, this comes with many challenges ranging from disciplines in science, engineering, computing and healthcare. However, the framework within BRAIN is such that it aims to support each new development as it is achieved.

Acknowledgements. The BRAIN consortium gratefully acknowledge the support of the European Commission's ICT for Inclusion Unit, under grant agreement No. 224156.

References

1. Neumann, N., Kubler, A.: Training Locked in Patients:A Challenge for the use of Brain Computer Interfaces. IEEE Trans. Neural Systems & Rehab. Eng. 11(2), 169–172 (2003)
2. Schalk, G., McFarland, D., Hinterberger, T., Birbaumerr, N., Wolpaw, J.R.: BCI200: A General purpose Brain Computer Interface (BCI) System. IEEE Trans. on Biomed. Eng. 51(6), 1034–1043 (2004)
3. Allison, B.: The I of bCIs: Next generation interfaces for brain–computer interface systems that adapt to individual users. In: Jacko, J.A. (ed.) HCI International 2009. LNCS, vol. 5611, pp. 558–568. Springer, Heidelberg (2009)
4. BRAIN, https://www.brain-project.org/ (accessed February 2011)
5. Garcia-Molina, G., Mihajlovic, V.: Spatial filters to detect Steady State Visual Evoked Potentials elicited by high frequency stimulation: BCI application. Journal of Biomedizinische Technik/Biomedical Engineering 3(55), 173–182 (2010)
6. Mason, G., Moore Jackson, M.M., Birch, G.E.: A General Framework for Characterizing Studies of Brain Interface Technology. Ann. Biomedical Engineering 33(11), 1653–1670 (2005)
7. Wizard of Oz, http://www.musicman.net/oz.html
8. Ware, M.P., McCullagh, P.J., McRoberts, A., Lightbody, G., McAllister, H.G., Nugent, C.D., Mulvenna, M.D., Thomson, E., Martin, S.: Contrasting Levels of Accuracy in Command Interaction Sequences for a Domestic Brain-Computer Interface Using SSVEP. In: Proc 5th Cairo International Conference on Biomedical Engineering, Cairo, December 6-18 (2010)
9. UPnP protocol, http://www.upnp.org/ (accessed February 2011)
10. Open Services Gateway initiative (OSGi) (http://www.osgi.org/) (accessed February 2011)
11. Digital Living Network Alliance standard, http://www.dlna.org/ (accessed February 2011)
12. Equinox 3.4.3 R34, http://www.eclipse.org/equinox/ (accessed February 2011)
13. Dreambox TV tuner, https://www.dream-multimedia-tv.de/en/dm-7025 (accessed February 2011)
14. Volosyak, I., Valbuena, D., Malechka, T., Peuscher, J., Gräser, A.: Brain-Computer Interface using Water-based Electrodes. J. Neural Eng (2010)
15. Graimann, B., Huggins, J.E., Levine, S.P., Pfurtscheller, G.: Visualization of significant ERD/ERS patterns in multichannel EEG and ECoG data. Clin. Neurophysiol. 113(1), 43–47 (2002)
16. OpenBCI, http://bci.fuw.edu.pl/wiki/OpenBCI_system (accessed February 2011)
17. OpenViBE, http://openvibe.inria.fr/ (accessed February 2011)
18. Millán, J.R., Rupp, R., Muller-Putz, G.R., Murray-Smith, R., Giugliemma, C., Tangermann, M., Vidaurre, C., Cincotti, F., Kubler, A., Leeb, R., Neuper, C., Muller, K.R., Mattia, D.: Combining Brain-Computer Interfaces and Assistive Technologies: State-of-the-Art and Challenges. Frontiers in Neuroscience 4, 161 (2010)
19. g.tec's Intendix, http://www.intendix.com/ (accessed February 2011)