

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



BCI Integration: Application Interfaces

Christoph Hintermüller, Christoph Kapeller,
Günter Edlinger and Christoph Guger

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/55806>

1. Introduction

Many disorders, like spinal cord injury, stroke or amyotrophic lateral sclerosis (ALS), can impair or even completely disable the usual communication channels a person needs to communicate and interact with his or her environment. In such severe cases, a brain-computer interface (BCI) might be the only remaining way to communicate [1]. In a BCI, the brain's electrical activity during predefined mental tasks is analyzed and translated into corresponding actions intended by the user. But even for less severe disabilities, a BCI can improve quality of life by allowing users to control a computer or specially prepared electronic devices, or to stay in contact with friends through social networks and games. P300 evoked potential [2, 3, 4] based BCIs can provide goal-oriented control as needed to operating spelling devices [5] or control computer games [6]. For navigating in space e.g. moving a computer mouse [7]), controlling the motion and movement of a robot or a wheelchair, steady state visual evoked potential (SSVEP) [8, 9, 10, 10] and motor imagination [12, 13] based BCI paradigms can be used.

All these applications require integrating the BCI with an external software application or device. This chapter will discuss different ways this integration can be achieved, such as transmitting the user's intention to the application for execution, updating the options available to the user, related actions and integrating visual BCI stimulation and feedback paradigms with the Graphical User Interface (GUI) of external applications. The current developments and efforts to standardize the application interfaces will be presented.

2. General structure of a BCI

A BCI system consists of several components (figure 1). The first is the biosignal acquisition system, which records the body's signals (like the EEG) used to extract the user's intentions

and responses to the presented stimuli and feedback. It consists of a set of electrodes and a dedicated biosignal amplifier, which typically directly digitizes the signals and transmits them to the feature extraction system. The feature extraction processes the signals and analyzes them with respect to specific signals like P300, SSVEP or error potentials [14, 15, 16, 17].

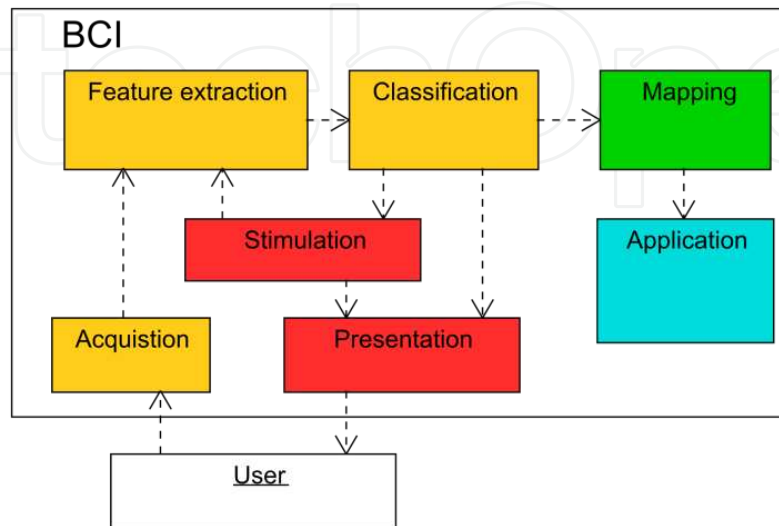


Figure 1. A BCI system consists of various components, which acquire, process and classify signals from the user's brain. Other components handle the presentation of dedicated stimuli and the feedback of the classification results. A dedicated mapping component converts them into commands and tasks to be executed by an attached or embedded application or service.

The classification determines the intention of the user based on the extracted features, which may reflect that the user does not intend to communicate at that time. The classification results are converted by a dedicated mapping component or appropriate methods into commands, actions and tasks to be executed by the attached applications such as a spelling device [2, 3, 4] or robot [10]. In order to enhance the user's response to the presented stimuli and to help assess the system's efficacy, a feedback related to the classification results is presented and the stimulation is adopted accordingly. Furthermore, the stimulation unit provides information about the presented stimuli using trigger signals and events, which are used to synchronously process the input signals and extract corresponding features.

3. Methods for integrating a BCI with an application

There exist three basic approaches to interconnect the BCI with a user application. The following sections, 3.1-3.3, discuss the different designs, their advantages and disadvantages. Each section will present some of the currently available BCI systems and frameworks that use that design. Section 3.4 compares the possibilities to establish standardized interfaces for interconnecting an application with the BCI system.

3.1. Direct integration

The most straight forward approach is to integrate the application within the BCI system. This approach, which is sketched in figure 1, allows developers to hardcode the symbols and feedback presented. In other words, the conversion of the classification results into application commands, tasks and actions, and the application itself, represent a static addendum to the BCI systems. This approach was used for the first proof of concept systems, and can still be found in simple spelling devices such as the P300 speller distributed by g.tec medical engineering along with the g.HIGHsys, a Highspeed Online Processing block set [18] for Matlab/Simulink™ (Mathworks, USA) and other BCI systems used for demonstration and educational purposes.

Modern BCI frameworks [19] such as OpenViBE [20], BCILAB [21] or xBCI [22] that are based on this design use a module based approach to allow application developers and designers to integrate their own application within the existing BCI framework.

The advantage of directly integrating the application within the BCI system is that it can be distributed and used as a compact, all-in-one component. Despite the need for appropriate acquisition and processing hardware, no additional interfaces or protocols are required.

The downside of this approach is that application developers and designers require some knowledge of the interpretation of the feature signals and classification results and how to convert them into appropriate commands, tasks and actions. Whenever an application must be added, updated, exchanged or removed, the presentation of stimuli and feedback has to be adjusted.

3.2. External executable component

The limitations of the tight integration approach can be reduced or eliminated by modeling the application as an individual executable that act as an external component of the BCI. As shown in figure 2, all other components like feature extraction, classification, stimulus and feedback presentation remain inside the core BCI system. This design is utilized by BCI systems which use the BCI2000 [23] or the TOBI platform [24], among other examples.

A well defined interface, such as the TiC output interface [25] used by the TOBI platform, allows applications to receive information based on the action that the user chooses to execute. If supported by the stimulus and presentation components, applications even may update and modify the choices available to the user through the BCI. Within TOBI, the TiD trigger interface [24] can be used by the application to initiate the processing of the change requests.

As a consequence, applications become independent from the BCI and may be connected and disconnected to and from the BCI any time. Both the BCI and the applications can, for example, be adapted to the user's needs independently without affecting each other. On the BCI side, this includes the selection of the acquired biosignals and features, improvements in the algorithms used to assess the user's intentions and the stimulation and feedback paradigms used. On the other hand, applications may dynamically adopt the available choices to some

internal state or provide new commands and actions to the user without the need to modify any of the core BCI components.

As shown in figure 2, converting the feature classification results has to be done by the application. This basically requires that developers and designers of BCI enabled applications have some knowledge of how to interpret these results with respect to the services, actions and tasks offered by their applications. Propagating changes from the internal state of the application to the user is only possible if the BCI is able to reflect these changes by adopting the options presented to the user accordingly.

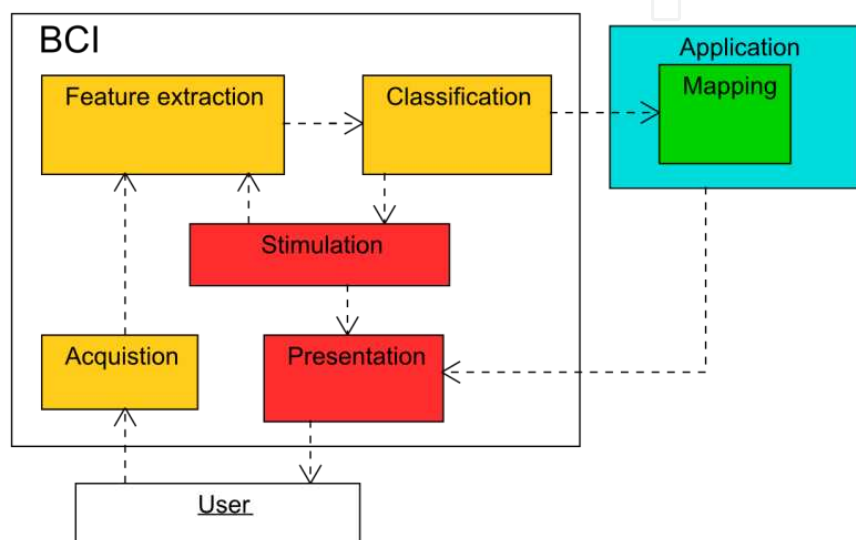


Figure 2. The application acts as additional, external executable component of the BCI system. It attaches to the different data and trigger interfaces to receive information about the user's intention and initiate the adoption of the stimulus and feedback presentation to its internal states.

3.3. Message based

The limitations of the approaches described in the previous sections, 3.1 and 3.2, can be overcome by integrating a dedicated interface module or component within the BCI system. The mapping component collects the classification results and converts them to corresponding application control messages and the connection interface component transmits each of them to the application, where they are interpreted and the requested services, actions or tasks are executed.

Depending on the capabilities of the core components of the BCI and the interface component, the application may request an update of the presented stimuli and feedback based on the application's internal state. Figure 3 shows the bidirectional case, where the application can acknowledge the reception of the transmitted messages and send update requests to the BCI whenever its internal state changes. The interface component of the connected BCI system decodes the corresponding messages and initiates the required changes and updates of the

stimulus and feedback presentation. Hence, the paradigms and input modalities the BCI uses do not matter.

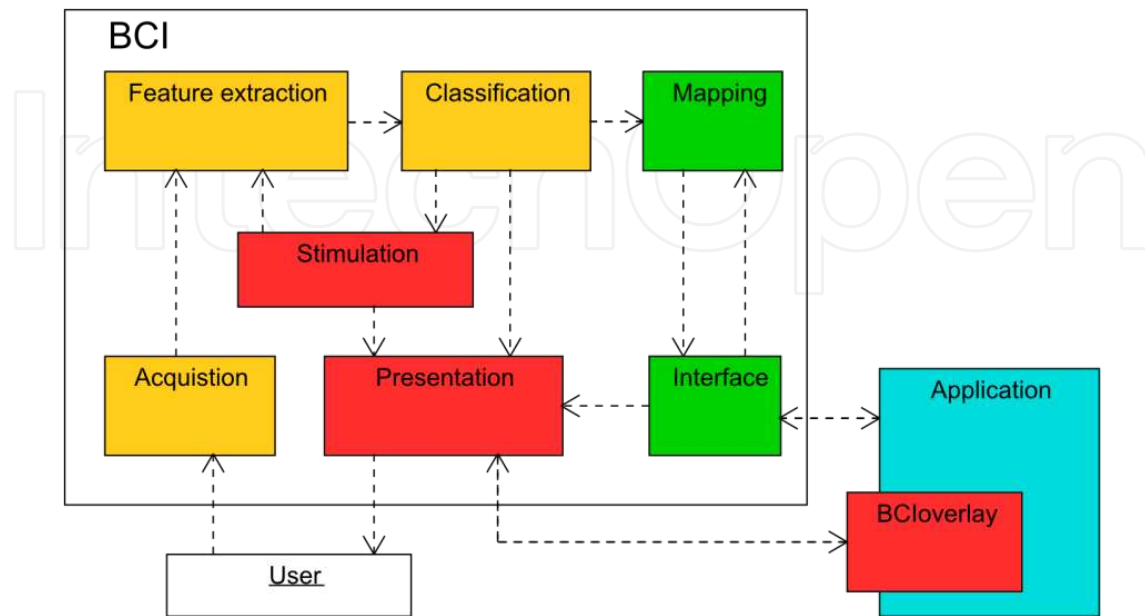


Figure 3. The BCI and the application are loosely coupled through dedicated interface components. An optional BCI overlay module allows embedding the presentation of the BCI stimulation and feedback remotely within the applications when requested by the application.

The message based approach provides a loose coupling between the BCI system and the attached applications. It is used, for example, by the *intendiX™* system (g.tec medical engineering GmbH, Austria), which is further described in section 4. Structural and technical changes applied to the BCI system, such as modifications of the signal acquisition, feature extraction, classification, stimulus or feedback presentation, have no impact on the attached application. Improvements and modifications of the services, actions and tasks offered by the applications do not require any technical or algorithmic changes to any part of the BCI system. The dedicated interface component, in combination with a well defined, message based and standardized protocol, enables applications to provide new features to the user any time, without modifying any of the BCI components.

In contrast to the approaches described in the previous sections, 3.1 and 3.2, a dedicated mapping component enclosed within the BCI (figure 3) interprets the classification results and converts them to their corresponding application control message. This encapsulation enables developers with little or no expertise in biosignal processing, analysis, classification and interpreting results to develop applications controllable by a BCI.

The resulting decoupling of the application from the BCI can be taken one step further by transparently embedding it within the user interface of any application. This is achieved by overlaying the BCI stimuli and feedback on top of the application's standard user interfaces. This may be achieved using an overlay module as described in section 4.3.

3.4. Connection design

The previous sections, 3.1-3.3, described the different ways to integrate the BCI within an application. Independent of these different approaches, the dedicated communication and control interface between the BCI and the application can be implemented using two distinct methods. Platforms like TOBI, BCI2000 or the unidirectional extendiX clients described in section 4.1 and in [26] provide dedicated application programming interface (API) libraries that handle all the connection and data transfer related issues. An application, as shown in figure 4a, calls the API functions to retrieve information about the user's intention and initiate the update of the presented stimuli and feedback. All the details about the underlying protocol and connection are hidden by the library. This allows administering modifications and extensions to the protocol any time, without affecting the application, unless the interfaces of the API functions need to be adopted to reflect the changes made.

A large variety of programming languages exist to implement applications and the API library. If the application is implemented using a different language than the API, a so called language wrapper library is necessary. Such a wrapper provides access to the BCI client API from within the language used to implement the application.

User data, selections and information about the internal state of the application are stored in dedicated data structures and fields. Hence, it is likely that these differ from ones used by the API library to receive instructions from and to collect update requests to be sent to the BCI. As a consequence, an application has to convert all data to and from its internal data structures before and after calling a BCI client API function. Depending on the intended usage and the related requirements concerning responsiveness, user experience and usage of computational resources, this may cause additional, probably undesired effects from the application.

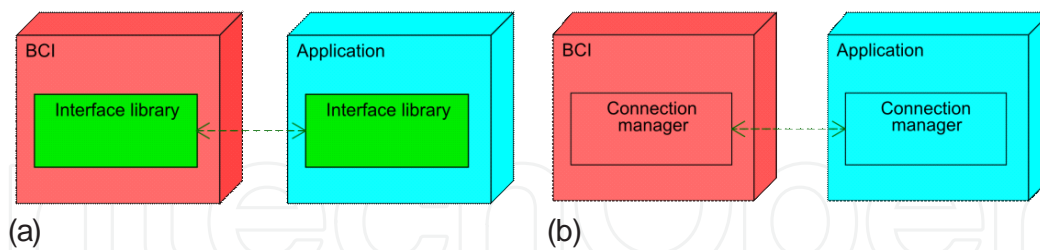


Figure 4. The interconnection of the BCI and the application can either be encapsulated within a dedicated library a) or be based on a dedicated protocol b).

Figure 4b shows a different way to establish a connection between the BCI system and an application. It is based on a well defined and standardized protocol only. Both the BCI and the application utilize their own, dedicated connection manager to handle the typically network based connection, process all received requests and generate all outgoing messages based on their current state. The connection manager interprets the incoming messages, extracts relevant data and information and converts them directly into the data structures and representations used by the application or the BCI respectively.

Changes to the application's internal state are directly converted into appropriate messages requesting changes and updates to the stimuli, the feedback presented to the user and the conversion of classification results to be returned to the application. Thus, it does not matter which programming language is used to implement the application. Instead of an API library wrapper, a dedicated connection manager has to be developed. This requires that the communication protocol uses a well defined and properly described handshake and communication procedures to avoid unnecessary workarounds, which might prevent the application from being used with BCI systems supporting a different version of the protocol.

4. Intendix™

The previous section discussed 3 different approaches to interconnect the BCI with user applications, services and devices. This section presents the intendiX™ system [26], which is used by the applications and projects presented in section 5.

The intendiX BCI system was designed to be operated by caregivers or the patient's family at home. It consists of active EEG electrodes to avoid skin abrasion, a portable biosignal amplifier and a laptop or netbook running the software under Windows (see figure 5a). The electrodes are integrated into the cap to allow a fast and easy montage of the equipment so it can be mounted quickly and easily. The software lets users view the raw EEG to inspect data quality, but automatically informs the inexperienced user if the data quality on a specific channel is good or bad.

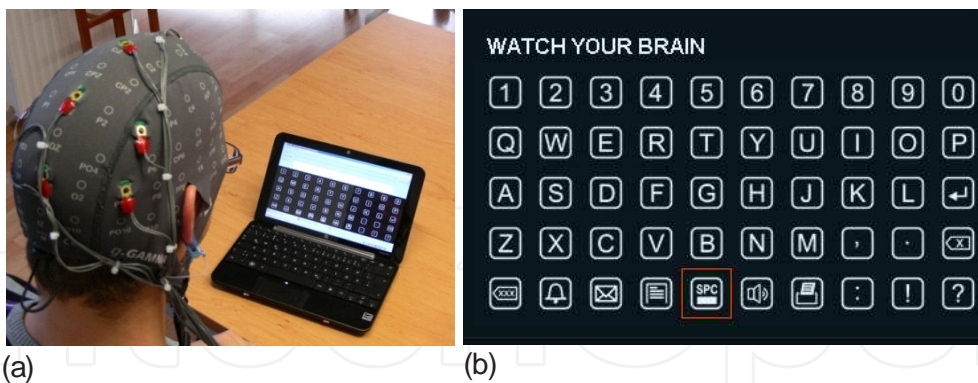


Figure 5. The user who is wearing the EEG cap equipped with active electrodes can run the intendiX system on a laptop a). By default, the intendiX presents a matrix of 50 characters using a layout comparable to a computer keyboard b).

This control can be realized by extracting the P300 evoked potential from the EEG data in real-time. The characters of the English alphabet, Arabic numbers and icons were arranged in a matrix on a computer screen (see figure 5b). Then the characters are highlighted in a random order while the user concentrates on the specific character he/she wants to spell. The BCI system is first trained on the P300 response of several characters with multiple flashes per character to adapt to the specific person.

During this training, 5-10 training characters are typically designated for the user to copy. The EEG data is used to calculate the user specific weight vector, which is stored for later usage. Then the software automatically switches into the spelling mode and the user can spell as many characters as desired. The system was tested with 100 subjects who had to spell the word LUCAS after 5 minutes of training. 72 % were able to spell it correctly without any mistake [4].

The speed and accuracy of the classification can be optimized by choosing the appropriate number of flashes manually. A statistical approach could also automatically determine the optimal number of flashes for a desired accuracy threshold. This latter approach could also determine whether the user is paying attention to the BCI system. The statistical approach could have a major advantage: no characters are selected if the user is not looking at the matrix or does not want to use the speller.

The intendiX™ user can perform different actions after spelling: (i) copy the spelled text into an editor; (ii) copy the text into an email; (iii) send the text via text-to-speech facilities to the loud speakers; (iv) print the text or (v) send the text via UDP to another computer by selecting a dedicated icon. The intendiX™ system offers two distinct ways to manage this interaction with external software and control various applications [26], described below.

4.1. extendiX

The first is the unidirectional extendiX protocol. It is accessible through the dedicated closed source extendiX library encapsulating the extendiX protocol.

For simple control applications, the extendiX batch file starter client allows users to start dedicated batch scripts whenever it receives information about the symbols selected by the user. This approach is suitable for all applications, services and devices that offer dedicated command-line interfaces for controlling their state and executing specific actions.

The intendiX™ Painting [26] program, a small painting program comparable to Microsoft Windows Paint, allows users to create paintings and other images, as well as store, load and print them.

4.2. intendiX ACTOR protocol

For complex control tasks, the intendiX™ Application ConTrol and Online Reconfiguration (ACTOR) Protocol is provided. Compared to extendiX, it offers a bidirectional User Datagram Protocol (UDP) message based connection between the BCI and the application. A short summary of the standardized intendiX ACTOR protocol can be found in [27] and its detailed description is available from [26]. No dedicated API library is available. All applications have to implement their own dedicated connection manager, which handles all intendiX ACTOR based communication.

The Interface Unit handles the UDP connections to all simultaneously attached applications, services and devices and converts the classification results to the corresponding UDP message string. This dedicated connection manager processes all connect, disconnect requests and status updates received, prepares the next set of stimuli and feedback modalities to be

presented to the user and instructs the corresponding components to adjust their outputs accordingly.

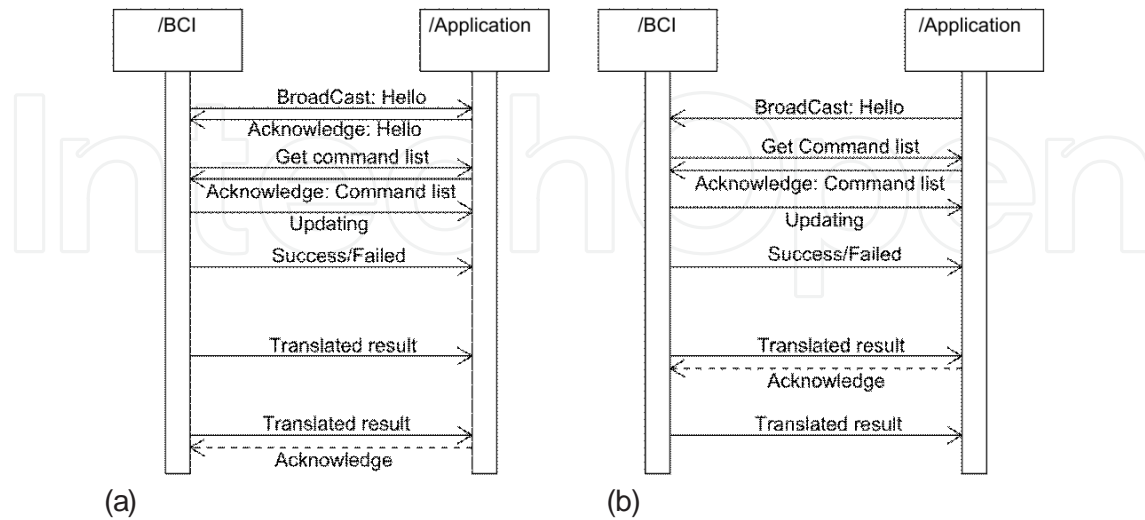


Figure 6. a) Handshake procedure used on startup by the BCI to identify all active applications and clients. b) Handshake procedure initiated by a starting application or client to register with a running BCI system.

The intendiX ACTOR protocol uses eXtensible Markup Language (XML) formatted message strings to exchange information between the BCI and the attached system. Whenever the BCI system is started, it broadcasts a dedicated hello message to identify the available and active applications, as shown in figure 6a. Each client responds by sending an appropriate acknowledgement message. As soon as the BCI has received this message, it will request from the application the list of applications and services available from this client. The BCI will acknowledge the received list of commands, services and actions and report whether it was able to process it successfully.

A similar handshake procedure is used when a newly started application connects to the BCI system. The only difference is that the BCI acknowledges the hello broadcast sent by the application by requesting the list of available commands, as shown in figure 6b.

Attached clients will receive a control message whenever the user has selected a service, action or task (figure 7a). The content of this message is fully defined by the application, which can ask the BCI to change this message and the related symbols, sounds and sequences presented to the user during stimulation and feedback. In this case, configured along with the definition of a single BCI control element, the BCI will pause the presentation of any stimuli and feedback until the application acknowledges the reception and execution of the control message or requests an update of the BCI user interface.

A single UDP message sent to the BCI may contain several distinct requests that are processed as an atomic batch. The presentation of stimuli and feedback that was paused while sending the last control message is not restarted before the last message within this batch has been processed. Each single message within such a batch may request the change of one single

stimulus or feedback element, or contain the content of a whole XML formatted configuration file that describes complex BCI screens and groups of stimuli. The latter is used by the IU to determine which stimuli and feedback modalities should be presented to the user. The detailed description of the configuration file format used is available, along with the definitions of the intendiX ACTOR protocol from [26].

Whenever the BCI receives an update request, it will return an acknowledge message indicating that it is trying to process and execute the update request. After the update is finished, a message indicating whether the request was executed successfully or failed is returned.

Whenever an application, service or device is terminated, it sends a good bye message to the BCI, which acknowledges this message sending a simple bye message. As shown in figure 7b, the same message is used by the BCI to inform connected clients that it will terminate operation.

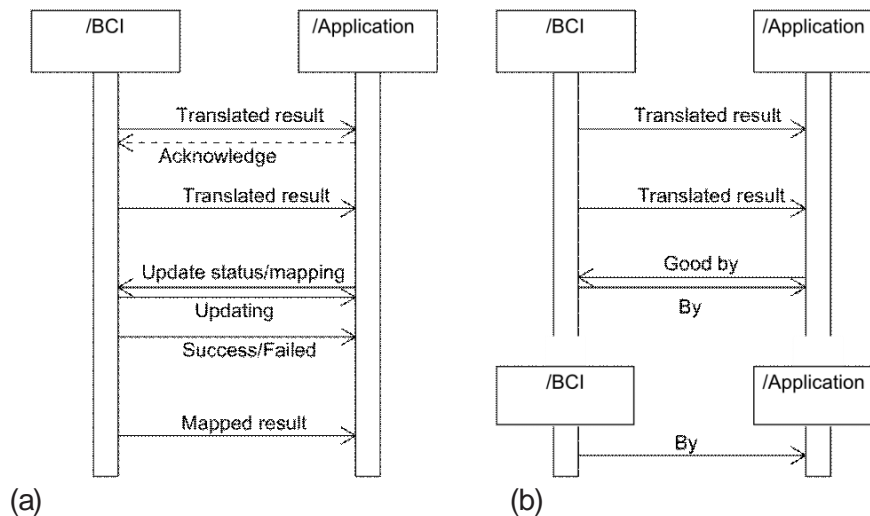


Figure 7. a) Sequences used by the BCI and application to transfer translated results and request updates. b) Sequences used by the application to disconnect from the BCI and by the BCI to dismiss applications on shutdown.

4.3. intendiX SOCI

For applications, especially virtual reality (VR) applications and remote control of robots, it is desirable to enhance the standard user interface by directly embedding the BCI stimuli (figure 8). The intendiX platform can be configured to remotely display its stimuli and feedback using the intendiX Screen Overlay Control Interface (SOCi) module [26]. The intendiX SOCI system implements a runtime loadable library based on OpenGL [28]. It is implemented as a dynamic linked library (DLL) for Microsoft Windows and as a shared object for LINUX, and can be used by OpenGL based host applications to embed targets for visual stimulation within the displayed scene. The host applications could be virtual reality environments or real-world videos acquired with a camera. Figure 8 presents an example of how to use the intendiX SOCI

module to simultaneously control the camera direction and select amongst different actions and objects.

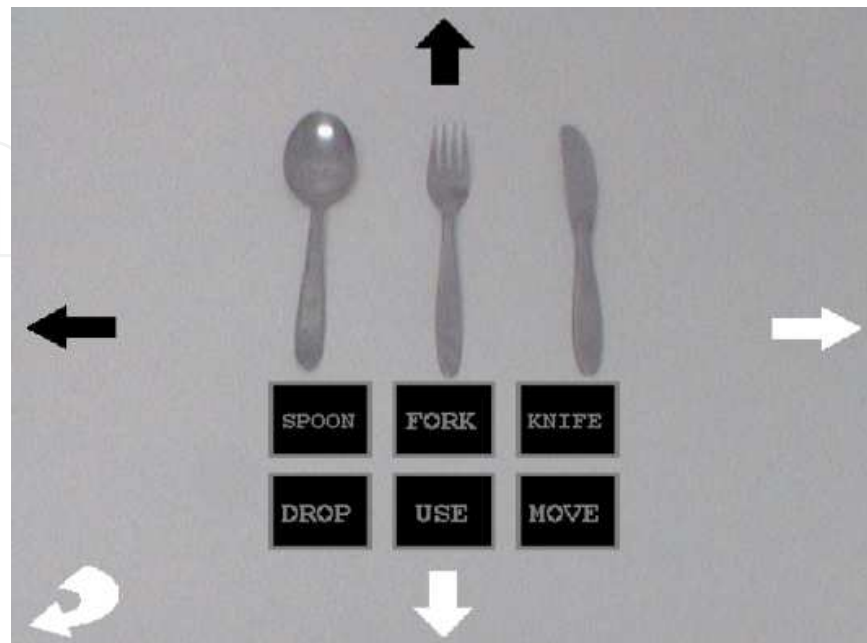


Figure 8. Example of how to use the intendiX SOCI module. Different BCI controls flash together within a running video application. In this example, the user uses the five outer controls to control the direction of the camera and the inner six controls to handle utensils like spoon, fork and knife.

The intendiX SOCI library is able to generate frequency (f-VEP) or code based (c-VEP) SSVEP stimuli, and supports single symbol and row column based P300 stimulation paradigms. It is initialized and fully controlled by the BCI system using a dedicated UDP based network connection. The application only needs to provide information on the network interface and port to be used when connecting to the BCI system and the screen refresh rate, which reliably can be achieved by the user display system. All other parameters are defined by the BCI system during startup and initialization phase. The intendiX SOCI module provides a standardized API, which allows the application to handle the stimulation and feedback devices attached to the user system appropriately.

Figure 9 shows the call sequence that the application must use to augment its own interface with the BCI controls. This sequence has to be repeated for each display, stimulation and feedback device that the BCI user wishes to use. The init function activates the BCI support for the selected device.

Every time before it calls the draw function, the application has to ensure that the OpenGL environment is initialized properly to display the BCI controls on top of any other graphical element. This is indicated in figure 9 by the call to the “set transformation” pseudo OpenGL function. After the swap buffer command from OpenGL has been called, the application has to call the displayed function to indicate that the stimuli have been updated.

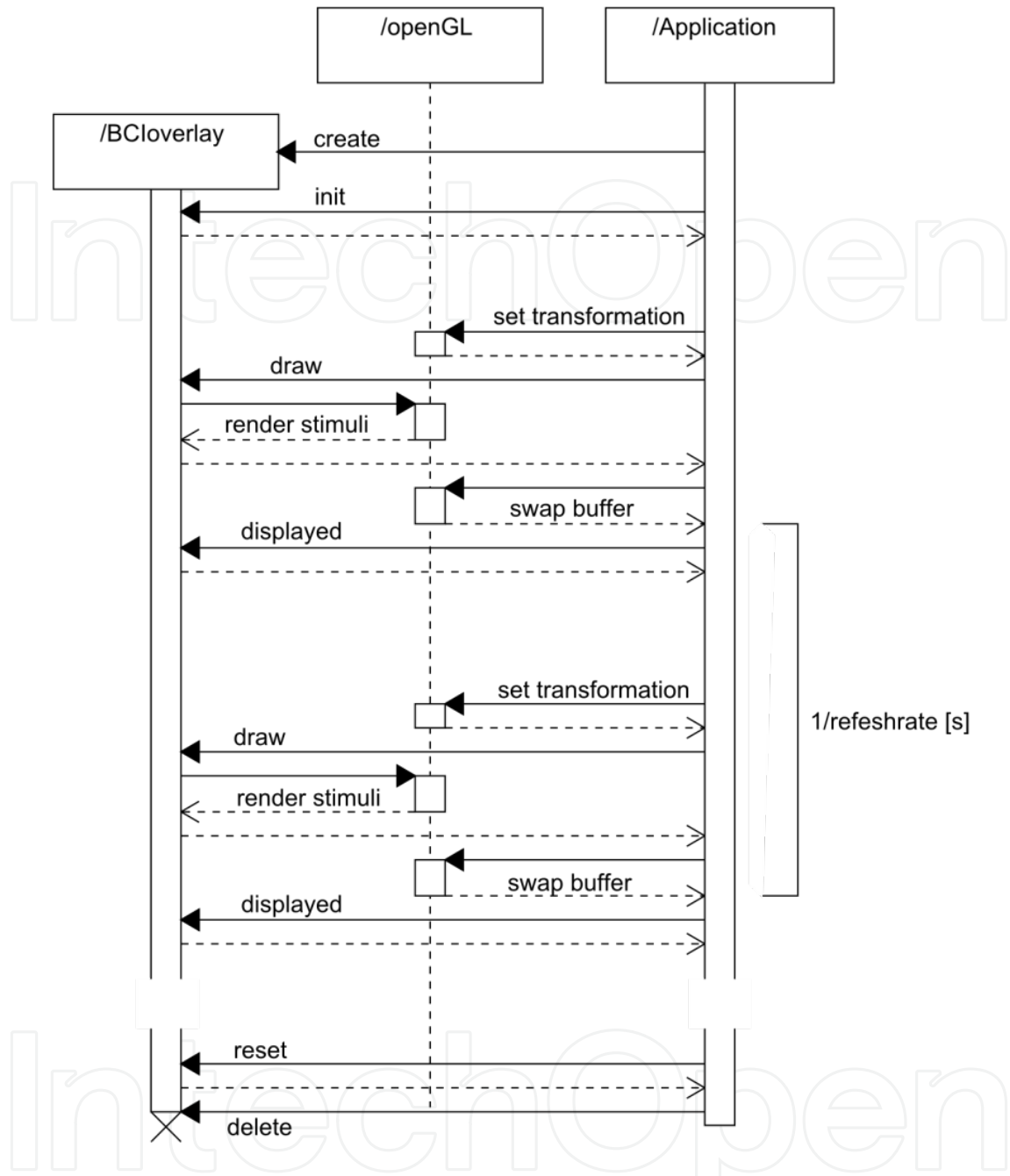


Figure 9. The order in which the init, draw, displayed and reset functions of the intendIX SOCI module have to be called by the application. The draw and displayed functions are called for every screen refresh cycle, which has a duration of 16.6 ms for a standard 60 Hz LCD flat screen.

When the application terminates, it disconnects the displays from the BCI by calling the reset function for each screen previously attached to the BCI through the init function. As indicated in figure 9, the application has to ensure that the interval between two consecutive calls to the displayed API function strictly corresponds to the screen refresh rate proposed by the application during initialization.

5. Applications

In section 3, the different concepts used to interconnect BCI systems and user applications were discussed. The application interfaces provided by the *intendiX*TM system were briefly presented in section 4. This section will list some projects and applications that utilize the *intendiX* ACTOR protocol and the *intendiX* SOCI API library to control virtual and robotic avatars and ambient assistance systems (sections 5.1, 5.2 and 5.3). Section 5.4 will present some examples in which the BCI was used to control cooperative multi player games such as *World of Warcraft* by Blizzard Entertainment [29] and social network platforms like Twitter (Twitter, Inc., USA). All of these applications either use the P300 matrix or a SSVEP paradigm with frequency coded stimuli.

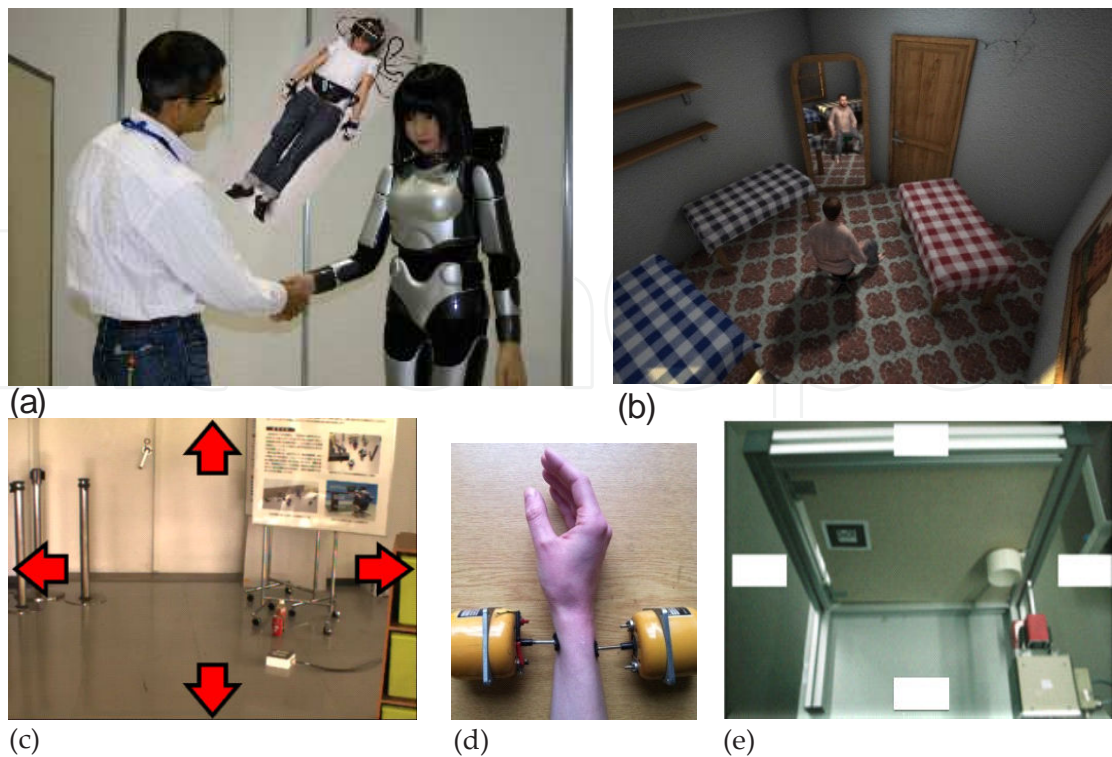
5.1. VERE

The VERE project [30] is concerned with embodiment of people in surrogate bodies so that they have the illusion that the surrogate body is their own body – and that they can move and control it as if it were their own. There are two types of embodiment considered. The first type is robotic embodiment (figure 10a), where the person is embodied in a remote physical robotic device and control it through a brain-computer interface. For example, a patient confined to a wheelchair or bed, who is unable to physically move, may nevertheless re-enter the world actively and physically through such remote embodiment. The second type of embodiment (figure 10b) is virtual, where participants enter into a virtual reality with a virtual body representation. The basic and practical goal of this type of embodiment is to explore its use in the context of rehabilitation settings.

The VERE project uses the *intendiX* ACTOR protocol (section 4.2) to access the BCI output from within the eXtreme Virtual Reality (XVR) environment (VRMedia S.r.l., Pisa, Italy) to control both the virtual and robotic avatars. The BCI is part of the intention recognition and inference component of the embodiment station, which is developed through the VERE project.

The intention recognition and inference unit takes inputs from fMRI, EEG and other physiological sensors to create a control signal together with access to a knowledge base, taking into account body movements and facial movements. This output is used to control the virtual representation of the avatar in XVR and to control the robotic avatar. The user gets feedback showing the scene and the BCI control via either the HMD or a display and the tactile and auditory stimuli provided by the so called embodiment station. Thereby the *intendiX* SOCI module is used to embed the BCI stimuli and feedback modalities within video streams recorded by the robot (figure 10c, e) and the virtual environment of the user's avatar.

The user is situated inside the embodiment station, which provides different stimuli and feedback modalities such as visual, auditory and tactile. Figure 10d shows the setup for inducing the illusion of hand movement by mechanically stimulating the flexor and extensor muscles of the hand.



Images courtesy of VERE, Event lab Universitat de Barcelona, Centre National de la Recherche Scientifique France, Perco laboratory Scuola Superiore Sant' Anna Pisa and Institute of Automatic control Engineering at the Technical University Munich, 2012.

Figure 10. VERE project aims at dissolving the boundary between the human body and surrogate representations in physical reality (a) and immersive virtual reality (b). One of the key aspects is to develop a brain body computer interface enabling the user to control the movement of his robotic avatar robot (c), open doors (e) and to provide him with visual (c, e) and tactile feedback on the body movements executed by the avatar (d).

Depending on the selected stimuli, the BCI system offers distinct levels of control. These levels range from high level commands, tasks and actions such as turning on the TV or grasping a can to moving the robot within its environment (figure 10c) or teaching the robot new high level tasks such as opening a door (figure 10e). The message based intendiX ACTOR protocol can switch smoothly between the different control levels and control paradigms.

5.2. ALIAS

The Ambient Assisted Living (AAL) research programme supports projects that develop technology to compensate for the drawbacks of the aging society by applying modern information and communication technologies (ICTs). The Adaptable Ambient Living ASsistant (ALIAS) project [31] is one the projects funded by AAL. It aims to improve the communication of elderly people, thus ensuring a safe and long independent life in their own homes. A mobile robot platform without manipulation capabilities serves as a communication platform for improving the social inclusion of the user by offering a wide range of services, such as web applications for basic communication, multimedia and event search and games.



Figure 11. The ALIAS robot utilizes several different sensors to perceive the user's input and intentions (a). In addition, it supports BCI systems through the intendiX ACTOR protocol and by embedding visual stimuli and feedback modalities using the intendiX SOCI module (b).

The ALIAS robot is equipped with sensing devices including cameras, microphones for speech input and a touch-screen (figure 11a) to perceive the user's input. The robot utilizes different modalities such as audio output, a graphical user interface (GUI) and proactive and autonomous navigation to interact with the user.

A so called dialog manager ensures that the dialog system can be controlled in a reasonable way. It is the central decision making unit for the behavior of the ALIAS robot and its interactions with the human user. It manages the interplay between input and output modalities of the ALIAS robot, communicates with all involved modules of ALIAS and controls them.

Besides the touch-screen, which is used to display the GUI and two independent automatic speech recognition (ASR) systems, a keyword spotter and a continuous context search also operate in parallel. The ALIAS dialog systems support the intendiX ACTOR protocol for receiving input from BCI systems and updating the presented stimuli and feedback online, based on the active state of the robot.

The intendiX SOCI module is used to embed the BCI stimuli within the GUI (figure 11b), aligned to their corresponding buttons. This tight integration of the BCI enables users to easily utilize the ALIAS platform during recovery, such as recover from stroke. The BCI interface of ALIAS allows them to navigate through the different menus, start programs, chat with friends using Skype, call and dismiss the robot or issue an emergency call.

5.3. BrainAble

The BrainAble project [32] conceives, researches, designs, implements and validates an ICT-based Human Computer Interface (HCI). Such an interface is composed of Brain Neural Computer Interface (BNCI) sensors combined with affective computing and virtual environments to restore and augment the two main shortcomings of people with disabilities. It entails inner and outer components. The inner component aims at providing functional independence for daily life activities and autonomy based on accessible and interoperable home automation.

The outer component provides social inclusion through advanced and adapted social network services. The latter component is expected to dramatically improve the user's quality of life.

Within the BrainAble project, the core structures of the intendiX ACTOR protocol were designed, developed and extended. The ACTOR protocol is used to interconnect the BNCI system with the user's living environment. This includes elements such as lighting, shades, heating, ventilation, audio, video services such as radio or TV, intercoms and many more.

It further provides access to social network and online communication services, thereby augmenting the user's social inclusion. The user has access to all of these devices, services and social interaction tools and is able to control them through the BCI system.

5.4. Games and social media

The intendiX ACTOR protocol, in connection with the intendiX SOCI API, can also be used to control games such as World of Warcraft (WoW) [29] and social media like Twitter (Twitter Inc. USA) or Second Life (Linden Lab, USA).

World of Warcraft is a common Massively Multiplayer Online Role-Playing Game (MMORPG) in which the player controls an avatar in a virtual environment. The BCI system uses an SSVEP paradigm to control an avatar in WoW [29]. For basic movements, selecting objects or firing weapons, four control icons as shown in figure 12 are required. The bottom three icons are used to move the avatar forward and turn left or right. The fourth icon, the action icon, is located top left. It used to perform actions like grasping objects or attacking other opponents. Stimulation is done on the same 60 Hz LCD-display that also renders the game itself.



Courtesy of g.tec medical engineering GmbH, 2012

Figure 12. The intendiX SOCI module is used in combination with the intendiX ACTOR protocol to control the movements and actions of an avatar within the World of Warcraft multiplayer online game from Blizzard Entertainment, Inc.

Twitter (Twitter Inc.) is a social network that enables the user to send and read messages. The messages are limited to 140 characters and are displayed in the author's profile page. Messages

can be sent via the Twitter website or via smart phones or SMS (Short Message Service). Twitter provides also an application programming interface to send and receive SMS.

Figure 13a shows a UML diagram of the actions required to use e.g. the Twitter service. The intendiX ACTOR protocol is used to interconnect the Twitter interface with the BCI. This system uses a standard P300 spelling matrix (figure 13b), which was extended with commands required for Twitter.

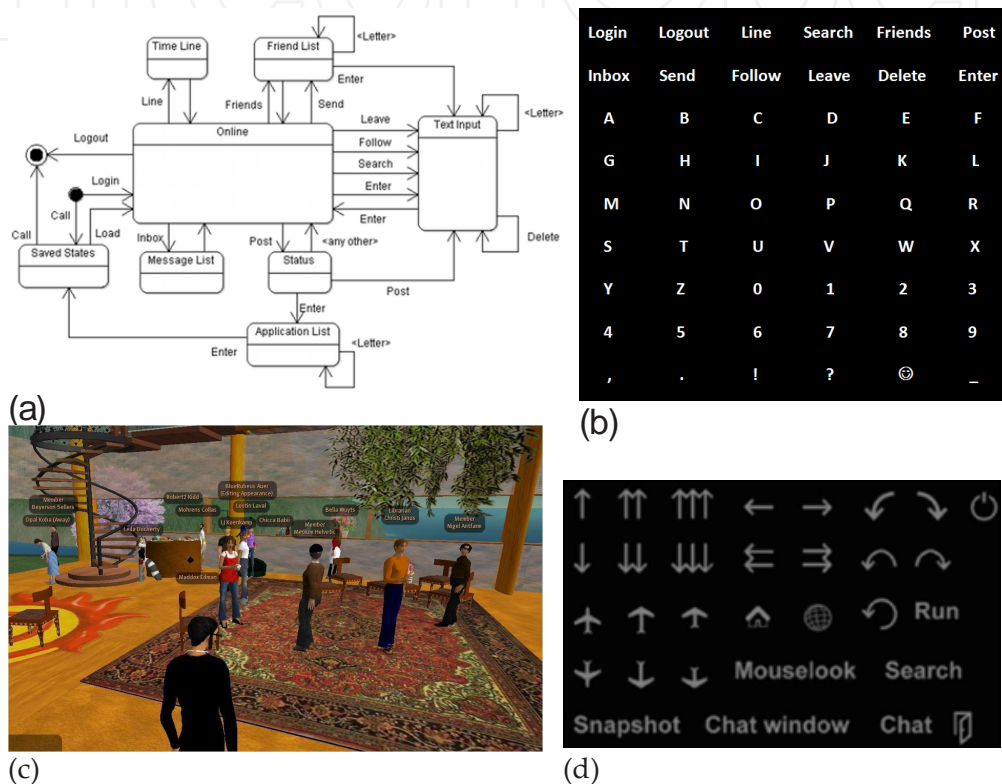


Figure 13. UML diagram of Twitter (a). A P300 BCI with a Twitter interface mask (b). Screenshot of a Second Life environment (c). The Second Life interface main mask for moving the avatar, climbing, running, flying, teleporting home, displaying a map, showing the search mask, taking snapshots, chatting with other members and managing the second life session (d).

Second Life is a free 3D online virtual world that can be accessed through the “Second Life Viewer”, which is free client software. A dedicated user account is necessary to participate in Second Life. One of the main activities in Second Life is socializing with other so-called residents. Each resident represents a person in the real world. Furthermore, it is possible to perform different actions like holding business meetings, taking pictures or making movies, attending courses, etc. Communication takes place via text chats, voice chats and gestures. Hence, handicapped people could also participate in Second Life just like any other user if an appropriate interface is available.

To control Second Life, three different interface masks were developed. Figure 13c displays a screenshot of a Second Life scene. The main mask (figure 13d) offers 31 different choices. The

masks for control like ‘chatting’ provided 55 control elements and the one for ‘searching’ 40 selections. Each of the icons represents an actual command to be executed within Second Life. Whenever a certain icon is selected, Second Life is notified to execute this individual action. Thereby, a dedicated keyboard event generator is used to convert messages based on the intendiX ACTOR protocol into appropriate key strokes. Further details on the BCI integration with Twitter and Second Life, including the results achieved by healthy subjects, are discussed in [33].

6. Conclusion

Current research projects aim to establish BCI systems as assistive technologies for disabled people, thereby helping them interact with their living environment and facilitating social interaction. These efforts rely on properly interfacing the BCI with supporting systems, devices, services and tools. For example, this interface could embody the user in an avatar or robot, as done within VERE, or in a robotic assistant, as implemented within ALIAS. The design and implementation of the interfaces between the BCI and the application do have an impact on the flexibility of the resulting assistive system or device and thus on the autonomy and independence of the user. Highly flexible interfaces like intendiX ACTOR and intendiX SOCI make it possible to adapt the BCI to the user’s needs while providing a standardized interface for using the BCI as control and interaction device with a large and constantly growing number of applications, assistive services and devices.

Acknowledgements

This work was supported in part by the European Union FP7 Integrated Project VERE, grant agreement no. 257695 and BrainAble by the European Community’s, Seventh Framework Programme FP7/2007-2013, grant agreement n° 247447. The authors gratefully acknowledge the support of the ALIAS project funded by the by the German BMBF, the French ANR and the Austrian BMVIT within the AAL-2009-2 strategic objective of the Ambient Assisted Living (AAL) Joint Programme.

Author details

Christoph Hintermüller, Christoph Kapeller, Günter Edlinger and Christoph Guger

g.tec Mmedical Engineering GmbH/ Guger Technologies OG, Austria

References

- [1] Wolpaw, J. R, Birbaumer, N, Mcfarland, D. J, Pfurtscheller, G, & Vaughan, T. M. Brain-computer interfaces for communication and control," *Clin. Neurophysiol.*, June (2002). , 113(6), 767-791.
- [2] Krusienski, D. J, Sellers, E. W, Mcfarland, D. J, Vaughan, T. M, & Wolpaw, J. R. Toward enhanced P300 speller performance," *J. Neurosci. Methods*, January (2008). , 167(1), 15-21.
- [3] Ortner, R, Bruckner, M, Prückl, R, Grünbacher, E, Costa, U, Opisso, E, Medina, J, & Guger, C. Accuracy of a Speller for People with Motor Impairments," *Proceedings of the IEEE Symposium Series on Computational Intelligence* (2011). in press., 300.
- [4] Guger, C, Krausz, G, Allison, B. Z, & Edlinger, G. (2012). A comparison of dry and gel-based electrodes for BCIs. *Frontiers in Neuroscience*, 6:60., 300.
- [5] Farwell, L. A, & Donchin, E. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalogr. Clin. Neurophysiol.*, December (1988). , 70(6), 510-523.
- [6] Finke, A, Lenhardt, A, & Ritter, H. The MindGame: A P300-based brain-computer interface game," *Neural Networks*, November (2009). , 22(9), 1329-1333.
- [7] Citi, L, Poli, R, Cinel, C, & Sepulveda, F. P300-Based BCI Mouse With Genetically-Optimized Analogue Control," *IEEE Trans. Neural Syst. Rehabil. Eng.*, February (2008). , 16(1), 51-61.
- [8] Friman, O, Volosyak, I, & Graser, A. Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces. *IEEE Trans Biomed Eng.*, 54(4): 742-750, (2007).
- [9] Wang, Y, Gao, X, Hong, B, Jia, C, & Gao, S. Brain-Computer Interfaces Based on Visual Evoked Potentials. *IEEE Eng Med Biol Mag.* (2008). Sep-Oct., 27(5), 64-71.
- [10] Faller, J, Müller-putz, G. R, Schmalstieg, D, & Pfurtscheller, G. An Application Framework for Controlling an Avatar in a Desktop-Based Virtual Environment via a Software SSVEP Brain-Computer Interface. *Teleoperators and Virtual Environments-Presence*, (2010). , 19(1), 25-34.
- [11] Kapeller, C, Hintermüller, C, Abu-alqumsam, M, Schau, T, Gro, B, Windhager, V, Putz, R, Prückl, A, & Peer, C. Guger, SSVEP based Brain-Computer Interface combined with video for robotic control, *IEEE Transactions on Computational Intelligence and AI in Games* (2012).
- [12] Guger, C, Ramoser, H, & Pfurtscheller, G. (2000). Real-time EEG analysis with Subject-Specific spatial patterns for a Brain-Computer Interface (BCI). *IEEE Transactions on Rehabilitation Engineering*, , 8-4, 447-456.

- [13] Blankertz, B, Tomioka, R, Lemm, S, Kawanabe, M, & Müller, K. R. Optimizing spatial filters for robust EEG, *IEEE Signal Processing Magazine*, 25(1) ((2008).
- [14] Schalk, G, Wolpaw, J, Mcfarland, D, & Pfurtscheller, G. EEG-based communication: Presence of an error potential," *Clin. Neurophysiol.*, (2000). , 111, 2138-2144.
- [15] Parra, L, Spence, C, Gerson, A, & Sajda, P. Response error correction-A demonstration of improved human-machine performance using real-time EEG monitoring," *IEEE Trans. Neural Syst. Rehabil. Eng.*, Jun. (2003). , 11(2), 173-177.
- [16] Blankertz, B, Dornhege, G, Schäfer, C, Krepki, R, Kohlmorgen, J, Müller, K. -R, Kunzmann, V, Losch, F, & Curio, G. Boosting bit rates and error detection for the classification of fast-paced motor commands based on single-trial EEG analysis," *IEEE Trans. Neural Syst. Rehabil. Eng.*, Jun. (2003). , 11(2), 127-131.
- [17] Ferrez, P, Del, J, & Millán, R. You are wrong!-Automatic detection of interaction errors from brain waves," in *Proc. 19th Int. Joint Conf. Artificial Intell.*, (2005).
- [18] g.tec medical Engineering GmbH. (2012). "g.HIGHsys." Retrieved September 2012, from <http://www.gtec.at/Products/Software/High-Speed-Online-Processing-under-Simulink-Specs-Features>.
- [19] Brunner, C, Andreoni, G, Bianchi, L, Blankertz, B, Breitweiser, C, Kanoh, S, Kothe, C, Lecuyer, A, Makeig, S, Mellinger, J, Perego, P, Renard, Y, Schalk, G, Susila, I. P, Venthur, B, & Müller-putz, G. (2013). BCI Software Platforms. In: *Toward Practical BCIs: Bridging the Gap from Research to Real-World Applications*, Allison, B.Z., Dunne, S., Leeb, R., Millan, J., and Nijholt, A. Springer-Verlag Berlin, , 303-331.
- [20] French National Institute for Research in Computer Science and Control (INRIA)INRIA), Rennes, France, "OpenViBe", Retrieved September (2012). from <http://openvibe.inria.fr>
- [21] Swartz Center for Computational NeuroscienceUniversity of California, CA, San Diego, USA, "BCILAB", Retrieved September (2012). from <http://sccn.ucsd.edu/wiki/BCILAB>
- [22] Department of Electronics and Intelligent SystemsTohoku Institute of Technology, Sendai, Japan, "xBCI", Retrieved September (2012). from <http://xbci.sourceforge.net>
- [23] Wadsworth CenterNew York State Department of Health, Albany, NY, USA, "BCI2000", Retrieved September (2012). from <http://www.bci2000.org>
- [24] TOBI Tools for Brain-Computer Interaction projectTOBI", Retrieved September (2012). from <http://www.tobi-project.org>
- [25] Realtime bio-signal standardsTobi iC Definition, implementation and scenarios", Retrieved August (2012). from <http://www.bcistandards.org/softwarestandards/tic>

- [26] g.tec medical engineering GmbH, "intendiX", Retrieved September 2012, from <http://www.intendix.com/> and <http://www.gtec.at/Products/Complete-Solutions/intendiX-Specs-Features>
- [27] Putz, V, Guger, C, Holzner, C, Torrellas, S, & Miralles, F. A Unified XML Based Description of the Contents of Brain-Computer Interfaces.,” in Proc. 5th International Brain-Computer Interface Conference 2011, (2011). pp. Pages.
- [28] SGI(2012). OpenGL- The Industry’s Foundation for High Performance Graphics.” Retrieved July 2012, from <http://www.opengl.org/>
- [29] Blizzard Entertainment Inc(2012). World of Warcraft.” Retrieved July 2012, from <http://us.battle.net/wow/en/>.
- [30] VERE Virtual Embodiment and Robotic Re-Embodiment Project Website, Retrieved September (2012). from <http://www.vereproject.eu/>
- [31] ALIASAdaptable Ambient Living Assistant, Project Website, Retrieved September (2012). from <http://www.aal-alias.eu/>
- [32] BrainAbleProject Website, Retrieved September (2012). from <http://www.brainable.org/en/Pages/Home.aspx>
- [33] Edlinger, G, & Guger, C. Social environments, mixed communication and goal-oriented control application using a Brain-Computer Interface”, Prof. Human Computer Interface Conference, Orlando, (2011). in press.

