



DIPARTIMENTO DI INGEGNERIA INFORMATICA  
AUTOMATICA E GESTIONALE ANTONIO RUBERTI



**SAPIENZA**  
UNIVERSITÀ DI ROMA

**My-World-in-My-Tablet:  
an Architecture for People with Physical  
Impairment**

Mario Caruso  
Febo Cincotti  
Francesco Leotta  
Massimo Mecella

Technical Report n. 2, 2012

# My-World-in-My-Tablet: an Architecture for People with Physical Impairment \*

Mario Caruso

*Sapienza Università di Roma,  
DIAG, Italy*  
caruso@dis.uniroma1.it

Francesco Leotta

*Sapienza Università di Roma,  
DIAG, Italy*  
leotta@dis.uniroma1.it

Febo Cincotti

*IRCSS Fondazione Santa Lucia,  
Italy*  
*Sapienza Università di Roma,  
DIAG, Italy*  
f.cincotti@hsantalucia.it

Massimo Mecella

*Sapienza Università di Roma,  
DIAG, Italy*  
mecella@dis.uniroma1.it

## ABSTRACT

Mobile computing, coupled with advanced types of input interfaces, such as Brain Computer Interfaces (BCIs), can improve the quality of life of persons with different disabilities. In this paper we describe the architecture and the prototype of an assistive system, which allows users to express themselves and partially preserve their independence in controlling electrical devices at home. Even in absence of muscular functions, the proposed system would still allow the user some communication and control capabilities, by relying on non-invasive BCIs. Experiments show how the fully-software realization of the system guarantees effective use with BCIs.

## Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces—*Haptic I/O*; H.5.2 [Information Interfaces and Presentation]: User Interfaces—*Input devices and strategies (e.g., mouse, touchscreen)*; C.3 [Computer Systems Organization]: Special-purpose and Application-based Systems—*Real-time and embedded systems*

## General Terms

Design; Human Factors

## Keywords

---

\*This work has been partly supported by the EU projects FP7-224332 SM4ALL (<http://www.sm4all-project.eu/>), FP7-224631 TOBI ([www.tobi-project.org/](http://www.tobi-project.org/)) and by the AriSLA grant for the project BRINDISYS (<http://www.brindisys.it/>).

Brain Computer Interfaces (BCIs); tablet; home appliances; communication capabilities; software architecture

## 1. INTRODUCTION

A cure for many neurodegenerative diseases is still unknown, yet advancements in life-supporting technologies and clinical practice allow a growing number of patients to survive longer. For instance, persons with Amyotrophic Lateral Sclerosis (ALS), undergo a degenerative process that last years, in which motor functions are progressively lost; due to the heterogeneity of the disease (e.g. bulbar versus spinal forms), each patient experiences his/her own path of function deprivation; finally any chance of communication and action on the environment is lost; unless a fatal event occurs (e.g. a respiratory crisis) these individuals enter a locked-in state. While the advancement of life support technology and clinical practice can prolong the life of these subjects, it also extends the period in which his/her motor functions are very poor or even absent, leading to a state of complete dependence on the caregivers. As a consequence, social inclusion and quality of life of people with neurodegenerative diseases is decreasing, while the social cost for their assistance is increasing. Beside neurodegenerative diseases, other congenital or acquired deficits of the neuro-muscular system may lead to mild to severe limitations of mobility (e.g. paraplegia and tetraplegia), motor skills (as in the case of dystonia), and speech (aphasia). Traumatic brain injury, spinal cord injury, cerebrovascular stroke, Parkinson's Disease are examples of possible causes.

In this paper we present the architecture and the prototype of an assistive system suited for different inputs, fitting the residual abilities of the user, and aimed at preserving his/her communication ability at any stage of a progressing disease. The system allows the user to express himself and partially preserve his/her independence in controlling electrical devices at home. Even in absence of muscular functions, the proposed system still allows the user some communication and control capabilities, by relying on non-invasive Brain-Computer Interfaces (BCIs). In fact, by relying on modulation of brain activity voluntarily induced by the user, and detected by processing his/her electroen-

cephalogram (EEG), BCI research has shown in the past decade the possibility of a communication even in absence of any muscular contraction. Its design allows an early adoption of the aid, when the user can still operate it by means of conventional interfaces (e.g. a manual or automatic scan button), and can be re-configured whenever the user, due to his/her decay of motor abilities, feels he/she is no more able to operate it.

The proposed architecture is based on a tablet device, and consists of two main software components: the AUXILHOME component, which provides basic communication tools and a flexible access to home automation appliances, and the FLEXYGLASS component, which allows operating different mainstream applications using a common interface supporting different kind of aids.

The rest of this paper is organized as follows. Initially, background information and state-of-the-art methods and techniques relevant to our work are presented; then the overall architecture is introduced. The following two sections are devoted to AUXILHOME and FLEXYGLASS components, respectively. A validation is then presented, aiming at showing the technical feasibility of the system, and finally concluding remarks and outline of possible future work are drawn.

## 2. RELATED WORK

The term *assistive technology* (AT) originally included all kinds of accessible, adaptive and rehabilitative devices addressed to people with disability, aimed at improving their activities and participation and thus their quality of life. Nowadays the term significantly changed its meaning, including, in addition, a wide variety of software solutions which replace, and in some case improve, the features originally provided by specific devices.

Input devices for AT systems can be classified into two main categories, namely *pointing devices* (e.g. trackballs, joysticks and touch screens, or more advanced ones such as trackers) and *switches* (or more generally binary input devices, used in combination with automatic scan or step scan systems). Beside these classical input methods, last years have seen a growing interest in *brain computer interfaces* (BCIs) and more generally in biosignal based interfaces (e.g. muscle-computer interfaces [15]).

BCI is intended as a mean for providing severely physically impaired people (locked-in subjects) a direct communication channel between the brain and an external electronic device. In particular, many studies have been conducted with so called non-invasive BCIs in order to translate electroencephalographic (EEG) activity or other electrophysiological measures of brain function into commands. Non-invasive BCIs exploit measurement electrodes which are placed using a cap over specific areas of the scalp without entering neither in the skull (partially invasive BCI) or in the brain (invasive BCI). A set of different EEG features which can be translated into control signals together with the needed processing steps are described in [18]. Selected features are subject dependant and may actually change over time by following physiological and psychological fluctuations. A BCI translates these features either time-continuous or time-discrete control signals.

While BCI performance improved over the years, the required hardware and software became cheaper and simpler to use opening up to the possibility of bringing BCI directly at home without the continuous assistance and support of a specialized technician. Nowadays, portable and relatively cheap EEG amplifiers are available on the market, and can be accessed by an off-the-shelf computer using a standard interface (e.g. RS232, USB and Bluetooth). Moreover, electrode technology has sensibly improved. Beside *wet* electrodes (which require the application of an high-viscosity conductive gel each of them and the scalp), reliable *active* electrodes, which require a very little application effort, are available. Manufacturers are also going to produce *dry* electrodes, installed on a cap or an helmet, to be worn without any kind of assistance. Additionally a set of established software platforms is freely available for real-time EEG signal processing devoted to providing BCIs (e.g. BCI2000<sup>1</sup> and OpenVibe<sup>2</sup>). Recently a middleware based approach has been proposed<sup>3</sup>, in order to ease researchers in developing their own BCI implementation exploiting a set of provided facilities.

*Augmented and alternative communication* (AAC) represents a classical AT application aiming at compensating for severe speech-language impairments in the expression or comprehension of spoken or written language. One of the most common AAC aid is the communication board which allows the user to select predefined words or sentences, that s/he wants to communicate to the interlocutor, with a single touch. Nowadays many software packages for communication running on a common personal computer are available; they can simulate communication boards (both alphabetical or symbolic), reproduce virtual keyboards, and can be equipped with word prediction systems and vocal output, giving an additional access the internet. These software accepts different kind of inputs and, recently, methods for BCI control have been proposed [1] [12].

Home automation (*domotics*) represents a promising AT application area. Commercial home automation aids are mostly stand-alone devices such as infrared remote transmitters for the control of multimedia devices, wireless alarm bells and speaker-phones. At Georgia Tech, a domotic home has been built for the elderly adults with the goals of compensating physical decline, memory loss and supporting communication with relatives [10]. This work also considers issues of acceptability of domotics identifying key issues for the adoption of the technology by the end user. Acceptability, dangers and opportunities are also surveyed in [13]. At Carnegie Mellon, people's behavior is studied by automatic analysis of video images [6]. This is fundamental in detecting anomalies and pathologies in a nursing home where many patients live. Pervading the environment with active landmarks, called Cyber Crumbs, aims at guiding the blind by equipping him/her with a smart badge [14]. A number of projects to give virtual companion to people, to monitor people's health and behavioral patterns, to help persons with Alzheimer, are presented in [8]. The Gator Tech Smart House [7] is a programmable approach to smart homes tar-

<sup>1</sup>cf. [www.bci2000.org](http://www.bci2000.org)

<sup>2</sup>cf. <http://openvibe.inria.fr>

<sup>3</sup>cf. <http://www.bcistandards.org>

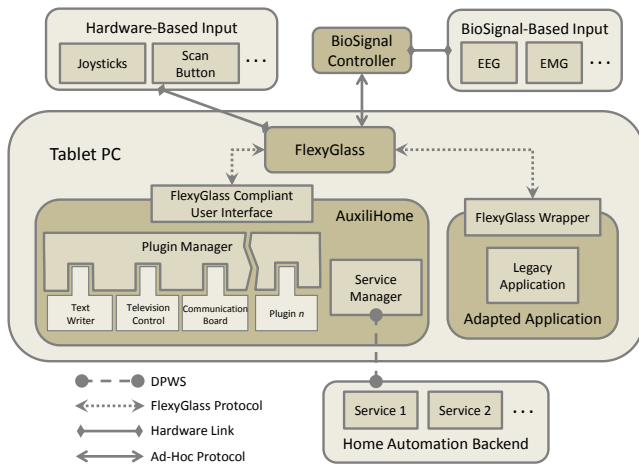


Figure 1: My-World-in-My-Tablet architecture.

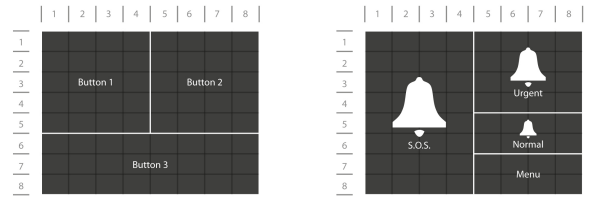
getting the elder citizen based on automatic service discovery and composition. This approach is close to what we propose in the AUXILIHOMe component (see further), though it does not commit to any open standard and, most notably, there is no specific attention toward user interfaces.

### 3. ARCHITECTURE

The goal of our work is the definition of a modular user-centric platform, depicted in Figure 1, in which an off-the-shelf tablet is used as a generic and extensible junction point for a set of technologies.

Input methods are split into *hardware-based* and *biosignal-based* ones. Different input methods allow for different information transfer rates from the user to the controlled devices, which influence the speed a selection or a task is performed. A user usually chooses the input method which, by taking into account his/her abilities, degree of impairment and required effort, offers the higher information transfer rate. So one of the advantages of such an architecture is the transparency of the input method with respect to the controlled application. Using a biosignal as input method is only possible using a biosignal controller, which is the component (hardware or software) devoted to the translation of the biosignal into a *control signal*. This work focuses on BCI as biosignal-based input method so, in the following, we will refer to this component as the BCI controller. This component may run directly on the tablet as well as on a different computer connected to the acquisition hardware. Its output may provide either a discrete control signal or a continuous one. At the actual stage of development, our prototype relies on BCI2000 as a BCI controller providing a discrete control signal based on P300 feature (see Section 5.1) of EEG.

The employed *home automation back-end* is a service-based home automation system, described in [2], which hides all the home automation functionalities behind well-defined software services. The relationship between usability and accessibility of the final application and service back-end have been inspected in [3]. The employed ubiquitous computing back-end has been designed to be enough versatile to be application agnostic, allowing to detect, use and compose every kind of device on a *semantic* base.



(a) Different size buttons (b) Alarm bell application

Figure 2: Buttons as main UI elements.

The tablet represents the way a user with impairments is able to communicate his needs and ideas and to control the domestic environment using the available input methods. This mobile device runs several different applications, most of which have not been designed to interact with special input aids. The most innovative component of our architecture, namely FLEXYGLASS, has been devised as a way to provide a standard interaction method with installed software, despite the variety of available input methods for the user. An application to be controlled via FLEXYGLASS has to adhere to a specific communication protocol (see Section 4.2). The AUXILIHOMe component represents an example of such a controlled application. It provides the user with an adaptive and extensible GUI, which gives access to a set of application plugins and to the services provided by the home.

*Personalized dynamic accessibility* [5] aims at achieving a more effective user interaction by making the software adaptive with respect to user’s needs and abilities changing over time. This is obtained through the *customizability* and the *dynamic adaptivity* of the user interface. Our work pursues the same purposes providing respectively (i) the possibility to configure and personalize the applications composing AUXILIHOMe (see Section 4.1) and (ii) the possibility of using several different input devices and modalities thanks to the novel FLEXYGLASS subsystem (see Section 5).

## 4. AUXILIHOMe

### 4.1 UI and Interaction

The graphical interface of AUXILIHOMe consists of several screens, each of them based on a grid layout into which graphical components can be placed. The number of rows and columns of the grid can be configured and determines the minimum size of a graphical component. Such a grid layout is easily described in a declarative manner, through a specific XML document, to be offered by services and external applications.

The key element of the UI is the button; as shown in Figure 2(a), different buttons belonging to the same screen can assume different sizes, allowing to strengthen the visibility of specific objects of interest according to specific user’s needs. Figure 2(b) depicts the screen of the alarm bell application; the latter allows the user the call the attention of the caregiver by producing different sounds according to different levels of priority of the request. The S.O.S. button is the biggest one, in this way it will be perceived and reached more easily and quickly. The variability of buttons size can

be used also as a way to prevent unwanted and erroneous selections.

As well as the screen layout, graphical components can be customized, as the user can choose colors, fonts, etc. In the design process we targeted a minimalistic interface, avoiding irrelevant or rarely needed information. The default look of the graphical user interface was designed aiming at complying with usability and accessibility guidelines [11]. Buttons come with a matte black background and white icons and labels. Only these two colours were used to code information in the whole interface and they maximize contrast ratio. The big contrast and the uniform background improve to screen clarity and readability.

The graphic is consistent in all the AUXILIHOME applications so that the user can familiarize with the system. Moreover, people with neurodegenerative diseases such as ALS, progressively loses movement abilities and consequently are forced to change input modality; AUXILIHOME can accommodate those changes, maintaining the same graphical interface paradigm.

## 4.2 External Services and Plugins

AUXILIHOME is a collection of applications that implement functionalities needed by users with disability; such applications are accessible through the graphical interface. The level and type of disability vary from user to user, but may also evolve over time for the individual user thus, for this reason, the organization of applications should be as dynamic and flexible as possible. Differently stated, applications can be added or removed easily from the system whenever necessary, must be configurable and, in order to provide maximum expandability of the system, can be developed and deployed by third parties.

We classify applications into two main categories: those that are *tightly coupled* and those that are *loosely coupled* to the graphical interface and to the tablet. Generally speaking, we can say that all the home automation applications can be considered loosely coupled to the tablet, this makes sense also because they are remote applications, that is, they live outside the tablet since they belong to a specific home environment and installation. *Light control* for example allows the user to turn on or off a light. In this case the behavior of the application and of the UI can be easily modeled with a descriptive approach. Conversely, all communication applications reside permanently on the mobile device and make use of its internal module for speech synthesis. *Text writer* is an example of such a kind of application allowing the user to write free text and to convert it into synthesized speech. Text writer is considered tightly coupled to the user interface, for each character selection, in fact, the character must be displayed in the text area, and the word prediction system must be updated.

From a practical point of view, loosely coupled applications are realized as *Web Services* while tightly coupled ones are *plugins* hosted by AUXILIHOME.

### 4.2.1 Web Services

Both sensors and actuators in the home make their functionalities available according to a *service oriented architecture*

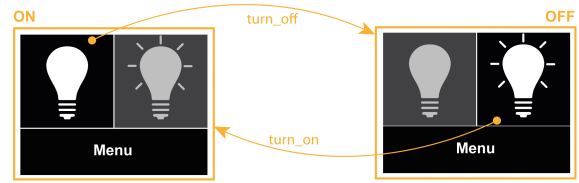


Figure 3: Light service model.

(SOA). The SOA approach in pervasive environments, presented in [2], motivates the use of Web Services as a way to face the heterogeneity of device’s specific protocols. According to a *rich service model*, a Web Service consists not only of the service interface specification, but also of its conversational description and of the graphical widgets (i.e., icons) needed when presenting the operations in the user interface. Services are thus software components offering complex operations by “wrapping” and abstracting the real devices realizing the functionalities.

The service model focuses on the behaviour of services, in terms of conversational states they traverse during the execution of the exposed actions. Each state of the device has a related graphical interface description which specifies the screen layout and the buttons’ icons/labels. A state may have one or more transitions, each of them is associated to a specific button (see Figure 3).

Among the currently available technologies for implementing Web Services, we chose Devices Profile for Web Services – DPWS<sup>4</sup>. As a distinctive feature DPWS brings Plug-and-Play capabilities to Web Services. DPWS defines a minimal set of implementation constraints to enable Web Service messaging, discovery, description, and eventing on resource-constrained devices. Relying on DPWS, our system does not need any centralized repository or event notification system; devices are discovered as soon as they join the local area network and the appropriate application is dynamically loaded on the tablet whilst, on the other hand, the asynchronous *event driven architecture* of DPWS allows the graphical user interface to immediately reflect changes in the state of each device.

### 4.2.2 Plugins

AUXILIHOME is the host application, it provides common functionalities that the plugin can use, e.g., the speech engine. Plugins must implement a simple interface consisting of some methods that are executed during their life-cycle (see Figure 4): (i) *Initialize*, executes all the operations that have to be executed at the startup of the plugin; (ii) *Run*, starts the normal flow of the plugin; (iii) *Pause*, suspends the plugin. The pause method is invoked, for example, when the “Menu” button is pressed; (iv) *Restore*, restores the plugin; (v) *Dispose*, is executed when the user closes the plugin; it is in charge of terminating all the active tasks freeing all the resources.

As said before, the plugin architecture grants maximum expandability to the system allowing third party applications

<sup>4</sup>cf.

<http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01>

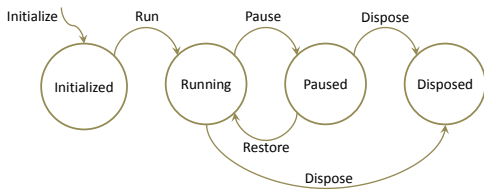


Figure 4: Plugin life-cycle.

to be easily developed and deployed. In addition, the developer has to focus his work only on the core of its application without worrying about the graphical interface or the input system details that are under the responsibility of the AUXILIHOME container.

#### 4.2.3 Currently available Services and Plugins

Currently AUXILIHOME offers two applications/plugins for communication purpose. The first one displays a list of frequently used sentences, when a sentence is selected, the related text is synthesized into speech. The list of sentences is fully customizable by the user or by the caregiver and can be extended unlimitedly. The other one consist of a virtual keyboard through which the user can write sentences and convert text into speech by selecting the appropriate button. The application shows a flipped copy of the inserted text at the top of the screen to make possible face-to-face conversations. In order to improve text insertion speed, the application provides a word completion system.

With regard to home automation, AUXILIHOME allows to command virtually any kind of device. In our tests we successfully controlled Konnex-based devices<sup>5</sup>, in particular we developed plugins able to turn on/off lights, raise/lower the armchairs and beds, ring the alarm bell in order to call the attention of the caregiver and open/close doors. Moreover we developed several application to drive infrared controlled devices such as TVs and DVD players.

## 5. FLEXYGLASS AND ADAPTERS

The FLEXYGLASS component is an independent software module which makes the employed input method totally transparent to the controlled application. The basic idea (see Figure 5) behind FLEXYGLASS is the definition of a transparent pane overlaid to the controlled application UI. This transparent pane contains a set of virtual controls which inherit size and position from the controlled application real controls (buttons, links or focusable objects) and which act in principle as proxies: if a virtual control is selected using the input method chosen by the user, the corresponding real control is triggered.

The described approach requires a direct connection between a controlled application and the FLEXYGLASS, and a communication protocol allowing FLEXYGLASS to (i) request the list of currently available controls to the controlled application in order to update virtual controls (in case of asynchronous UI changes the controlled application should also have the possibility of pushing a new configuration) and (ii) to communicate the last selection triggering the execution of

<sup>5</sup>cf. <http://www.knx.org/>

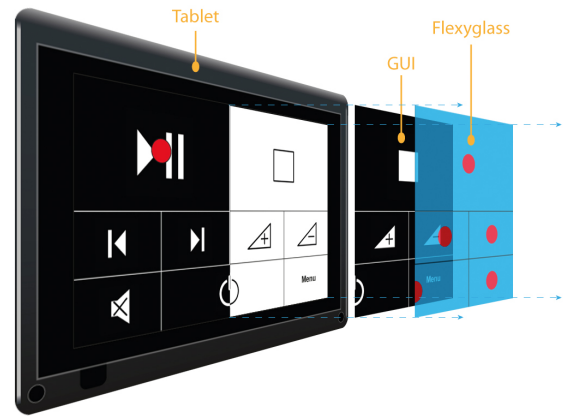


Figure 5: A layered perspective of FlexyGlass.

a real command (the controlled application should respond with an acknowledgement whenever the requested command has been executed).

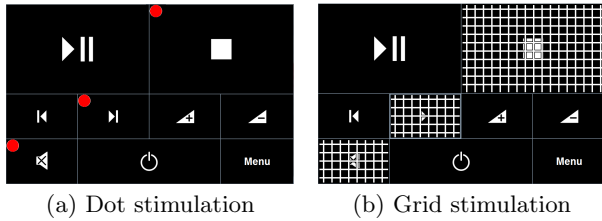
The FLEXYGLASS component is intended to support different kinds of input methods, each one requiring a different kind of interaction. As an example, in the case of a scan button, it can be employed in two different fashions: (i) the *manual scan* which requires the use of a button in order to move the focus from one control to another and using a longer pressure to execute the command and (ii) the *automatic scan* mode which requires the user clicking on the button whenever the focus is on the desired command while the focus is periodically and automatically moved from one control to another.

User who are able to move a pointer (for example a mouse) but not to click may benefit from the *dwell mouse* interaction modality. In this case the control is automatically selected after a predefined dwell time.

In all of these cases, a focused control has to be highlighted, but the same holds for the whole set of discrete selection input methods. By knowing the position of the available controls of the controlled applications, the FLEXYGLASS is able to move a graphic highlight over a control by drawing over the layered window.

FLEXYGLASS have been designed to be easily extensible with different highlight graphics. This is obtained by leaving the possibility of implementing new graphics starting from a set of virtual functions as skeleton. Currently, two different stimulation graphics are available namely the *dot stimulation* [12] (see Figure 6(a)), and the *grid stimulation* [16] (see Figure 6(b)) which have been proved to be very effective with P300-based BCIs. The FLEXYGLASS graphical configuration utility allows for customizing each aspect of the available stimulations (colors, sizes, etc.). These settings are then stored into an XML configuration file to be automatically loaded at startup.

From a technical point of view, the FLEXYGLASS component is based on *layered* windows provided by Microsoft Win-



**Figure 6: The stimulation modalities shown over the DVD remote control interface of AuxiliHome.**

dows<sup>6</sup>. This kind of windows features the possibility of having a transparent or semi-transparent background. The approach has been depicted in Figure 5.

Applications which are compliant with FLEXYGLASS specification are automatically discovered through a *hello* protocol. There exist two kinds of application: those which are natively built to be controlled via FLEXYGLASS (it is the case of AUXILIHOME) and those (the vast majority), which we refer to as *legacy* applications, which require a the development of a *wrapper* in order to be controlled. A wrapper may come in different forms; e.g., in the case of a browser like Mozilla Firefox or Google Chrome, a wrapper may be implemented as an extension. As an additional note, depending on the selected input method, only some applications are suitable to be controlled via an interface like FLEXYGLASS; this is particularly true if the chosen input method is a BCI which requires buttons to be enough sized and spaced.

## 5.1 Control via a P300 BCI

Currently, FLEXYGLASS supports P300 as the BCI-based input method. The P300 potential is a large and positive deflection of the EEG activity which reaches a maximum of amplitude over the centro-parietal scalp areas between 250 and 400 ms after a relevant stimulus (called *target* stimulus), presented within a sequence of frequent irrelevant stimuli (called *no-target* stimuli), is recognized.

In the *oddball* paradigm, the user is asked to determine to which of two categories belong a sequence of frequent stimuli. If one of the two categories is rare, stimuli belonging to this category are followed, about 300 ms after the onset of the stimulus, by a P300 potential. A more powerful, in terms of information transfer rate, was obtained with the introduction of the *speller* paradigm [4] [9], which allowed to obtain an information transfer rate of 12 bits/min.

The speller paradigm is based on the use of a selection matrix divided in stimulation classes, one for each row and column (a 6 by 6 matrix features 12 stimulation classes) composed by a set of symbols stimulated together. The stimulation sequence is made up by a fixed number of consecutive shuffling of the whole set of stimulation classes. In the rest of the paper we will name *trial* the execution of a whole stimulation sequence together with the final selection. A specific symbol belongs to both a single row and column

<sup>6</sup>For an introduction to layered windows, the reader can refer to the technical documentation freely available on the MSDN - Microsoft Developer Network web site.



**Figure 7: Pause interface for P300 BCI input.**

so, if we suppose to have a  $n$  by  $m$  speller with  $k$  repetitions, a user concentrating on a symbol will see it flashing  $2k$  times within a train of  $2k(n + m)$  stimuli. At the end of a trial, a score is assigned to each stimulation class; the system then selects the row and column with the highest score and returns the symbol which lay on the intersection.

The usage of P300 as an overlaid stimulation interface was first introduced in [12]. Here a P300 overlaid interface was used to control the commercial assistive technology application suite QualiWorld. FLEXYGLASS makes that idea more general, by allowing to use coherent graphics to control generic applications with different kinds of inputs ranging from BCIs to hardware switches.

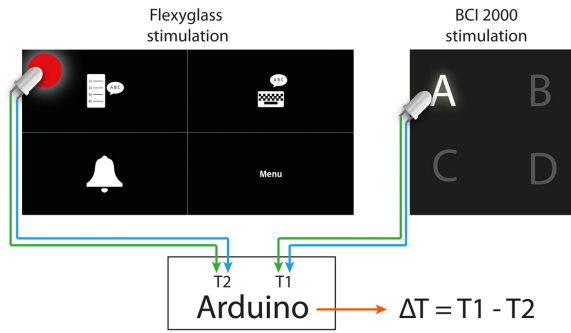
As previously described, the use of a BCI input requires a BCI controller (BCI2000 in our case). In order to make it possible to stimulate the user with an application external to the BCI2000 (i.e., the FLEXYGLASS), a communication protocol has been devised, which allows in particular (i) to indicate to the BCI2000 whenever the stimulation has to start, and (ii) to communicate to the controlled application not only the result of the classification, but also the stimulation sequence.

Creating an overlaid interface raises the problem of how the classical matrix layout of the P300 speller may be adapted to a more general layout. In order to solve this problem, before beginning a trial, FLEXYGLASS analyses the controls available on the controlled application, chooses the minimum matrix size which contains all the controls and create a one-to-one association between controls and matrix positions.

Controlling an application using P300 is very different from any traditional input method, especially as it requires a continuous attention to the stimulation in order to avoid incorrect selection bringing the controlled application into an unwanted state. The user might desire to take a break from the stimulation, by putting the system in pause mode because he is either tired or occupied in some other tasks. FLEXYGLASS introduces a pause mechanism consisting in an additional fake control shown over the transparent window. If the user selects this control during the stimulation, FLEXYGLASS goes in pause mode stimulating over the window shown in Figure 7. In order to exit from the pause mode, the user will have to select a fixed number of consecutive times the play control.

## 6. EXPERIMENTS

The experimental section of this work represents a feasibility study with respect to P300 BCI functioning prerequisites while usability test over users is still ongoing. In particular, what we wanted to analyse was the visualization delay of the stimuli over the FLEXYGLASS layered window, as one the novelties of the architecture proposed in this paper is indeed the FLEXYGLASS component. The BCI2000 includes a built-in speller which is directly connected to the sequence



**Figure 8: The hardware platform employed for the feasibility test.**

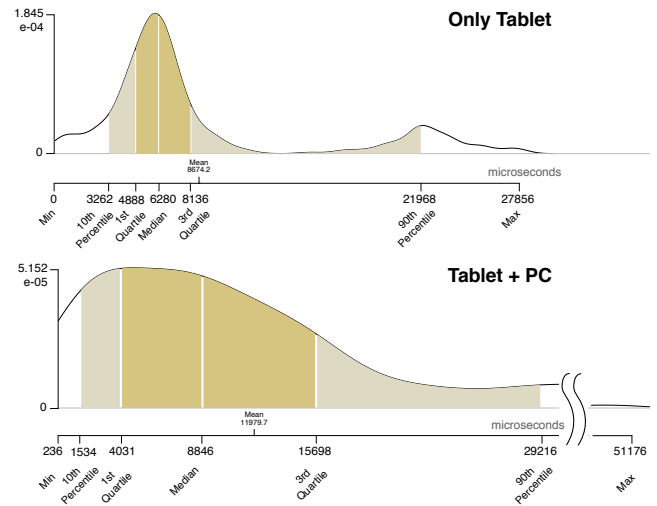
generation allowing for the best delay between the visualization request and the effective onset of the stimulus on the screen. In our case, the stimulation sequence is transmitted using an interprocess communication channel, thus potentially introducing unpredictable transmission delays. In order to measure the magnitude of the delays and their variability, we take the time instant a stimulus is shown on the BCI2000 interface as a “gold standard”, thus measuring the time difference until the correspondent stimulus is shown over the layered window of the FLEXYGLASS. In order to make this measure consistent we have developed a specific hardware test bed.

The test platform (Figure 8) consists of 2 photo-transistors (BPW-85) connected to the analog inputs of an Arduino One board<sup>7</sup> (a similar method has been proposed in [17]). The two photo-transistors are placed directly over the monitor screen; one is placed over an element of the BCI2000 speller and the other one over the corresponding control of the FLEXYGLASS transparent window. The board is in charge of detecting light flashes onsets on both transistors and calculating the time difference which is transmitted using a serial connection on a logging application running on a laptop. Calculating the differences directly on hardware ensure that the measurement are not affected by transmission delays. The only uncertainty to take into account is the raise time of the diode (which is in the order of  $2\mu s$  for BPW-85 photo-transistor) together with a maximum difference between photo-transistors trigger of 2 ms measured also experimentally.

The tablet used in our prototype and tests is an Acer Iconia Tab W500, equipped with Microsoft Windows 7, processor AMD C-50 1 GHz and 2 GByte RAM. Test results are shown in Figure 9. Two experiments have been performed. In both cases a sequence of 20 trials has been performed with 10 repetition of stimulus classes (so a single element flash 20 times per trial).

During the first experiment, the BCI2000 ran directly on the tablet together with FLEXYGLASS and AUXILIHOME. We can see how 75% (third quartile) of delay measurements are below 8.2 ms while the half of the measurements are comprised between 4.9 (first quartile) ms and 8.2 ms. During the second experiment, the BCI2000 ran on a separate ma-

<sup>7</sup>cf. <http://arduino.cc/en/Main/ArduinoBoardUno>



**Figure 9: Density function of measured delays.**

chine (an off-the-shelf laptop) connected via Ethernet to the tablet. In this case though the first quartile is similar to the first experiment, the third quartile is significantly higher then before (about 16 ms) with a maximum of 51 ms. This can be translated in a significant enlargement of variance.

Despite the fact that our tests showed a measurable delay in the FLEXYGLASS stimulation, this does not impact on the P300 BCI performances. In fact, the P300 waveform shows 1 or 2 order of magnitude slower time constants (e.g., the peak occurs 300 to 500 ms after the icon flashed). Thus, pattern recognition algorithms will be hardly affected by such a small relative time shift of the waveform.

A preliminary validation has been performed in the last months (February and March 2012) with 3 users, namely two disabled ones (with ALS, but not yet in the locked-in state) and a normal bodied clinician, used as “control” user. The tests consisted in some sessions of usage of the system, in each session the user employs a different device (i.e., the direct touch screen, a button / switch and the BCI) and then provides qualitative feedbacks on the usability and usefulness of the system. Such tests do not yet represent a deep usability study (due both to the absence of quantitative results and to the low number of involved users) but allowed us to gain some insights on possible improvements of the prototype and to validate the provided functionalities.

The reader should note that in Fondazione Santa Lucia, Rome, a fully equipped smart home (with 27 smart devices, i.e., sensors and actuators, cf. [2]) has been set-up, and it is currently used in our tests, specifically for the domotic features of our prototype.

The interface has been judged *very learnable and intuitive* from all users (i.e., it is simple to remember how the controls have been placed and to jump among them, especially with a BCI); users particularly appreciated the domotic functionalities and the possibility to communicate through the system; some of them emphasized the need of also having simple games as provided features, as for static patients (in partic-



ular young ones) this would provide an interesting amusement. A video showing some user section is available at <http://dl.dropbox.com/u/5460464/mhiat.zip><sup>8</sup>.

## 7. CONCLUDING REMARKS

This work presented a general architecture, and the related prototype, based on a tablet device, for allowing physically impaired people to interact with the surrounding in a fully automated home environment. The proposed approach allows for a great extensibility in terms of provided applications and input methods and adaptivity in terms of automatic acquisition of the available home automation services.

This is possible through a specific software architecture, whose pillars are the wrapping of home appliances as Web services, a plugin approach for incorporating specifically designed applications, and the FLEXYGLASS component for incorporating legacy applications and allowing specific input methods requiring stimulation (such as BCIs). The FLEXYGLASS may be used, in principle, in combination with every kind of application (provided with a specific adapter).

We have demonstrated, through some tests, that the introduction of FLEXYGLASS as overlay over existing applications does not degrade performances of the input methods, even in the case of BCIs.

A note should be reported about the possibility of using whichever application in conjunction with FLEXYGLASS. This is not completely true if the selected input method is P300 BCI. In fact P300 classification is very influenced by stimulus distribution over the screen, suffering in particular of stimulation very close one to each other. A possible way of addressing this issue could be the possibility for the FLEXYGLASS to automatically analysing the set of provided controls and reorganizing it using call out and subsets definition. Additionally while FLEXYGLASS is able to automatically detect compliant application (native or legacy), no accessible interface is currently provided to select among these applications or switch from an application to another one.

A future enhancement is about the menu structure of AUXILIHOME, which is currently fixed. Unfortunately, some aids provide a low information transfer rate which can potentially make a single selection expensive. Future version of AUXILIHOME could provide a menu structure which dynamically evolve following user favourite selections (learnt over past executions).

## 8. REFERENCES

- [1] M. Bensch, A.A. Karim, J. Mellinger, T. Hinterberger, M. Tangermann, M. Bogdan, W. Rosenstiel, and N. Birbaumer. Nessi: an EEG-controlled web browser for severely paralyzed patients. *Intell. Neuroscience*, 2007:6:1–6:10, April 2007.
- [2] C. Di Ciccio, M. Mecella, M. Caruso, V. Forte, E. Iacomussi, K. Rasch, L. Querzoni, G. Santucci, and G. Tino. The homes of tomorrow: service composition and advanced user interfaces. *ICST Trans. Ambient Systems*, 11(10-12), 2011.
- [3] W.K. Edwards, V. Bellotti, A.K. Dey, and M.W. Newman. The challenges of user-centered design and evaluation for infrastructure. In *Proc. CHI 2003*, pages 297–304, ACM.
- [4] L. Farwell and E. Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, 70(6):510 – 523, 1988.
- [5] K.Z. Gajos, A. Hurst, and L. Findlater. Personalized dynamic accessibility. *Interactions*, 19(2):69–73, Mar. 2012.
- [6] A. Hauptmann, J. Gao, R. Yan, Y. Qi, J. Yang, and H. Wactlar. Automatic analysis of nursing home observations. *IEEE Pervasive Computing*, 3(2):15–21, 2004.
- [7] S. Helal, W. C. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. The Gator Tech smart house: a programmable pervasive space. *IEEE Computer*, 38(3):50–60, 2005.
- [8] A. Joseph. Successful aging. *IEEE Pervasive Computing*, 3(2):36–41, 2004.
- [9] D. Krusienski, E. Sellers, F. Cabestaing, S. Bayouduh, D. McFarland, T. Vaughan, and J. Wolpaw. A comparison of classification techniques for the P300 speller. *Journal of Neural Engineering*, 3(4):299 – 305, 2006.
- [10] E. Mynatt, A. Melenhorst, A. Fisk, and W. Rogers. Understanding user needs and attitudes. *IEEE Pervasive Computing*, 3(2):36–41, 2004.
- [11] J. Nielsen. Ten usability heuristics. [http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html), 2002.
- [12] A. Riccio, F. Leotta, L. Bianchi, F. Aloise, C. Zickler, E.-J. Hoogerwerf, A. Kuebler, D. Mattia, and F. Cincotti. Workload measurement in a communication application operated through a P300-based Brain-Computer Interface. *Journal of Neural Engineering*, 8(2), 2011.
- [13] J. Roberts. Pervasive health management and health management utilizing pervasive technologies : Synergy and issues. *The Journal of Universal Computer Science*, 12(1):6–14, 2006.
- [14] D. Ross. Cyber crumbs for successful aging with vision loss. *IEEE Pervasive Computing*, 3(2):30–35, 2004.
- [15] T. S. Saponas, D. S. Tan, D. Morris, R. Balakrishnan, J. Turner, and J. A. Landay. Enabling always-available input with muscle-computer interfaces. In *Proc. 22nd annual ACM Symposium on User Interface Software and Technology (UIST 2009)*, pages 167–176, ACM.
- [16] M. Tangermann, M. Schreuder, S. Dahne, J. Hohne, S. Regler, A. Ramsay, M. Quek, J. Williamson, and R. Murray-Smith. Optimized stimulation events for a visual ERP BCI. *International Journal of Bioelectromagnetism*, 13(3):119–120, 2011.
- [17] J. Wilson, J. Mellinger, G. Schalk, and J. Williams. A procedure for measuring latencies in Brain-Computer Interfaces. *Biomedical Engineering, IEEE Transactions on*, 57(7):1785 –1797, July 2010.
- [18] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. Brain-Computer Interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767 – 791, 2002.

<sup>8</sup>The video is protected with password *reviewer* for opening