Smart Homes for All: Collaborating Services in a for-All Architecture for Domotics

Tiziana Catarci[1], Febo Cincotti[2], Massimiliano de Leoni[1],
Massimo Mecella[1], and Giuseppe Santucci[1]

[1] Dipartimento di Informatica e Sistemistica
SAPIENZA - Universita di Roma, Italy
{catarci,deleoni,mecella,santucci}@dis.uniroma1.it
[2] Fondazione Santa Lucia, Rome, Italy
f.cincotti@hsantalucia.it

Abstract. Nowadays, control equipments such as automobiles, home appliances, communication, control and office machines, offer their functionalities in the form of services. Such service pervasivity is particularly evident in immersive realities, i.e., scenarios in which invisible embedded systems need to continuously interact with human users, in order to provide continuous sensed information and to react to service requests from the users themselves. The SM4ALL project, which will be presented in this paper, is investigating an innovative middleware platform for collaborating smart embedded services in immersive and person-centric environments, through the use of composability and semantic techniques.

1 Introduction

Embedded systems are specialized computers used in larger systems or machines to control equipments such as automobiles, home appliances, communication, control and office machines. Such pervasiveness is particularly evident in immersive realities, i.e., scenarios in which invisible embedded systems need to continuously collaborate with human users, in order to provide continuous sensed information and to react to service requests from the users themselves. Examples of such scenarios are digital libraries and eTourism, automotive, next generation buildings and infrastructures, eHealth, domotics.

This human-service collaboration poses many new challenges to current technologies, in terms of *(i)* dynamism, *(ii)* scalability and dependability, and *(iii)* security and privacy. Indeed sensors/devices/appliances/actuators offering services are no more static, as in classical networks, (e.g., for environmental monitoring and management or surveillance), but the overall distributed system needs to continuously adapt on the basis of the user context, habits, etc. That is done by adding, removing and composing on-the-fly basic elements, that are the offered services. Moreover, in order to really immerse the users in the system, the number of sensors/devices/appliances/actuators should be huge, at least an order of magnitude more than the current situations. As an example, the current best-in-class smart houses count for tenths of sensors/devices/appliances/actuators, the

E. Bertino and J.B.D. Joshi (Eds.): CollaborateCom 2008, LNICST 10, pp. 56–69, 2009.

next generation smart houses for all will count hundreds of devices. Finally, when users make public their sensible data, the security of the overall invisible environment is crucial. Indeed if the system were hacked, it could potentially provide any sensible information of users. And that is especially critical when users have some diseases, disabilities, etc.). In the light of that, the design of the collaborating system should take into consideration privacy preservation. It should be built-in in the system, and not added-on later, as in current design practices.

The paper intends to describe the European-funded project SM4ALL (Smart hoMes for All - `http://www.sm4all-project.eu`), started on September 1st, 2008 and finishing on August 31st, 2011. SM4ALL aims at studying and developing an innovative platform for the collaboration of human-based and software smart embedded services in immersive and person-centric environments through the use of composition and semantic techniques. This is applied to the challenging scenario of private home and building in presence of users with different abilities and needs (e.g., young, elderly or disabled people).

In order to introduce the novel idea of collaborating services underlying SM4ALL, the reader should consider the following scenario: a person is at home and decides to take a bath. He/she would like to simply express this goal to the house (e.g., through a touch screen, but we will see later on other possible interfaces are being considered in the project) and have the services of the house collaborate in order to move the house itself to a new state which is the desired one. The temperature in the bathroom is raised through the heating service, the guardrobe in the sleeping room is opened in order to offer the bathrobe, the bath is filled in with 37 ^{o}C water, etc. If we suppose the person is a disabled one, some services cannot be directly automated, e.g., the one of helping the person to move into the bath. In this case, a service still exists, but it is offered by a human, e.g., the nursery, which is doing her job in another room, and that at the right moment is notified – through her PDA or any other device – to go into the bath and help the patient. Maybe this service is offered also by the son of the patient (or any other person), living in a nearby house, which is notified at the right moment, and if the nursery is not present at home, to help the patient. The scenario shows the idea of a society of services, some offered in a completely automated way through sensors/appliances/actuators, other realized through the collaboration of other persons, which moves continuously from a desired state to a new one, in order to satisfy user goals. Clearly, as in all societies, there are trade-offs to be considered (the goal of the person willing a relaxing bath is in contrast with the availability of the nursery/son offering the "help" service), which in our case are handled by suitably composing the most appropriate services, and possibly adapting their involvement during the runtime.

In this paper, after giving an overall insight of the SM4ALL architecture, we will outline a couple of techniques adopted in SM4ALL: Brain-Computer Interaction and Service Composition.

Brain-Computer Interaction (BCI) is a specific set of techniques, based on the interplay of hardware and software, that allows people to interact with a screen "through their mind". The project envisions the possibility of selecting a desired goal, which a user in the house would like to reach, among possible ones in the given state of the house. Such goals are proactively offered by the SM4ALL system on the basis of the available services and the current user context, which are perceived through the sensors and a profiling of previous actions and goals. Therefore, BCI interfaces are considered in the project as a suitable solution for disabled persons in order to drive the house.

Service Composition plays also an important role. We aims at proposing techniques to define the most suitable way of coordinating the available collaborating services, and will deploy such an orchestration specification on top of the infrastructure, through which the services interact each other in order to deliver some final composite service to the user. Such a composite service can effectively satisfy the user's goal, or can take the infrastructure "closer" to it.

Section 2 describes research works related to the intends of SM4ALL. Section 3 describes, firstly, the overall architecture and, then, details the initial techniques for BCI and Services' Composition. Finally, Section 4 concludes the paper, delineating the future work direction for the project.

2 Related Work

Presently, we are assisting at a blooming of research projects on domotics for assisting people with physical or mental disabilities.

For instance, at Georgia Tech a domotic home has been built for the elder adult with the goals of compensating physical decline, memory loss and supporting communication with relatives [1]. This study 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 [2]. At Carnegie Mellon people's behavior is studied by automatic analysis of video images [3]. 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 [4]. A number of projects to give virtual companion's to people, to monitor people's health and behavioral patterns, to help Alzheimer patients are presented in [5]. The social dimension is considered in [6], where social networks are used to model the social relationships of the user. This network is used for providing information or issuing alarms related to the home. The Gator Tech Smart House [7] is a programmable approach to smart homes targeting the elder citizen. The idea is to have a service layer based on OGSi [8] in order to enable service discovery and composition. This work is close to what we propose as for the SOA spirit, though it does not commit to any open standard or XML based technology hindering openness and dynamic scalability of the approach. No reference is made to the communication model adopted in the home and, most notably, there is no attention toward brain interfaces.

As far as service composition, there are been in the last years several works addressing it from different point of views. So far, the work on services has largely resolved the basic interoperability problems for service composition (e.g., standards such as WS-BPEL and WS-CDL exist and are widely supported in order to compose services, even if their applicability in embedded systems is still to be demonstrated), and designing programs, called orchestrators, that execute compositions by coordinating available services according to their exported description is the bread and butter of the service programmer [9].

The availability of abstract descriptions of services has been instrumental to devising automatic techniques for synthesizing service compositions and orchestrators. Some works have concentrated on data-oriented services, by binding service composition to the work on data integration [10]. Other works have looked at process-oriented services, in which operations executed by the service have explicit effects on the system. Among these approaches, several consider *stateless* (a.k.a., atomic) services, in which the operations that can be invoked by the client do not depend on the history of interactions, as services do not retain any information about the state of such interactions. Much of this work relies on the literature on Planning in AI [11,12,13]. Others consider *stateful* services which impose some constraints on the possible sequences of operations (a.k.a., conversations) that a client can engage with the service. Composing stateful services poses additional challenges, as the composite service should be correct w.r.t. the possible conversations allowed by the component ones. Moreover, when dealing with composition, data usually play an important role: typically they are sent back, forwarded during operation invocations and manipulated by the service. This work relies on research carried out in different areas, including research on Reasoning about Actions and Planning in AI, and research about Verification and Synthesis in Computer Science [14,15,16,17].

In SM4ALL, we focus on composition of process-oriented stateful services, in particular we aim at considering and extending the framework for service composition adopted in [18,19,20,21,22,23], sometimes referred to as the “Roman Model” [24]. In the Roman Model, services are represented as transition systems (i.e., focusing on their dynamic behavior) and the composition aims at obtaining an actual composite service that preserves such an interaction. The composite service is expressed as a (virtual) target service specifying a desired interaction with the client.

Finally, we would like to point out that some projects (e.g., EU-PUBLI.com [25] in an e-Government context and WORKPAD [26] in emergency management) have considered the issue of collaborating services, in which some services actually are not classical software applications, but human operators which executing actions are abstracted by the system as services and therefore seamlessly integrated into a general architecture.

3 Architecture

The vision of collaborating services pursued by SM4ALL requires that the following features are enforced:

Person-centric awareness. Humans are at the heart of new immersive environments and all efforts to develop such environments should be initiated and motivated by needs to provide interesting and/or novel experiences to users. This requires novel technologies for *(i)* data dissemination, *(ii)* their integration, *(iii)* user profiling, *(iv)* context computation.

Globally distributed, service-centric functionalities. The middleware and provided infrastructure services (e.g., storage and retrieval of service descriptions, communication, etc.) should be managed in a widely distributed manner to guarantee dynamism, scalability and dependability.

Openness and maximum-reuse. For economy of scale, reusability and extensibility, generic embedded middleware should be developed that capture all common aspects of the immersive scenarios. Openness is also worthy as it allows to consider any service for integration.

Figure 1 shows the SM4ALL architecture, consisting of different components:

Sensors and devices. Sensors are devices for the measurement of physical quantities. There is an ever increasing variety of sensor types, ranging from simple thermometers to self-calibrating satellite-carried radiometers. They are in charge of measuring properties of the area surrounding the sensor or at a certain distance. SM4ALL points to a vast plethora of sensors without

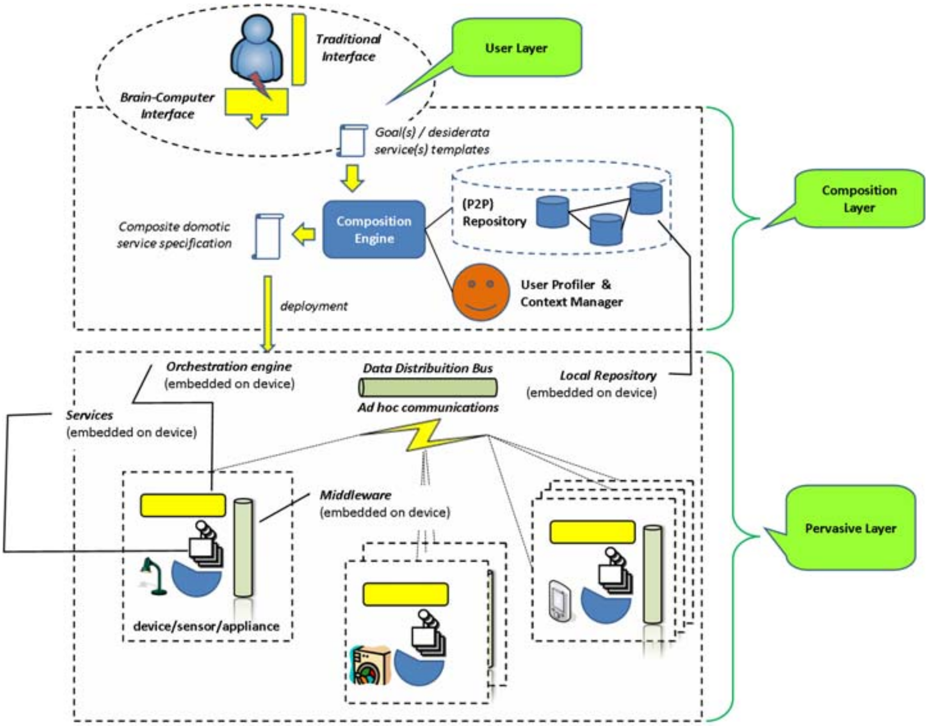
Traditional Interface
User Layer
Brain-Computer Interface
Goal(s) / desiderata service(s) templates
(P2P) Repository
Composition Layer
Composite domotic service specification
Composition Engine
User Profiler & Context Manager
deployment
Orchestration engine (embedded on device)
Data Distribuition Bus
Ad hoc communications
Local Repository (embedded on device)
Services (embedded on device)
Middleware (embedded on device)
Pervasive Layer
device/sensor/appliance

Fig. 1. The Overall SM4ALL Architecture

putting any limit. There exists, then, other devices (e.g., typical home appliances) that act on the environment for changin its properties (e.g., an appliance for opening the window can change the luminosity value sensed by a sensor).

Services. Both sensors and devices make their functionalities available according to the service oriented paradigm. Such embedded services can be either synchronous (e.g., when requesting an operation to a domestic appliance) or asynchronous (e.g., the service offered by a sensor sending the fridge temperature at regular interval). Moreover, asynchronous services can be arranged according to a publish & subscribe paradigm, in a direct-querying one, etc. Therefore, in SM4ALL, the full-fledged merge of SOAs (Service Oriented Architectures) and EDA (Event Driven Architectures) will be realized, on the basis of the standard Web service stack [27]. In order to be dynamically configured and composed, embedded services need to expose semantically rich service descriptions, comprising *(i)* interface specifications, *(ii)* specifications of the externally visible behaviors, *(iii)* offered QoS and *(iv)* security requirements.

Moreover, human actors acting in the environment can be abstracted as services, and actually "wrapped" in order to include them in the general architecture and make them collaborate with the software ones for pursuing a certain goal. Also these human-based services should be semantically described in order to include them in the more appropriate way. Their metrics can be derived through a continuous monitoring of their "performances" and user-profiling techniques); for instance, if the nursery is quite far away from a disabled person, the service "help" may present low response time and therefore, during a particular composition, may not be the best solution for reaching a given goal).

Embedded middleware. Devices, appliances and sensors are inter-connected in a wireless ad-hoc fashion, and a specific middleware needs to be available for allowing the effective interoperability among the offered services.

Embedded distributed orchestration engine. Collaboration of services needs to be carried out by a specific orchestration component. As the middleware, also this component is not centralized (as it currently happen with WS-BPEL engines in eBusiness and eGovernment scenarios), but need to be embedded in a P2P fashion in all devices/appliances/sensors of the house.

Composition engine. When the user selects a desired goal (e.g., expressed as a state of the house she would like to have realized), some automatic techniques need to synthesize the right orchestration of services able to satisfy such a goal. These techniques are realized by a composition engine, which is deployed on a special node of the infrastructure, and which is able, after the synthesis, to automatically deploy the orchestration specification on the orchestration engine for execution. Section 3.1 is devoted to give more details on this concern.

Repository of service description. Service descriptions, in order to be available during the synthesis process, need to be stored on some repository in a decentralized fashion.

Context-aware user profiler. The SM4ALL system aims at continuously profiling users and houses in order to anticipate their wishes and wills before users explicitly express them. That is supported by the composition engine as well as by BCI techniques. Morevoer, user profiling is crucial for deriving QoS metrics describing human-based services.

User interface. The user is able to interact with her own house through many interfaces, either centralized (e.g., in a home control station) or distributed and embedded in specific interface devices. User Interfaces exploits Brain-Computer-Interaction techniques in order to allow specific user categories (e.g., disable or elderly people) to access and interact with the domotic infrastucture.

Figure 1 depicts also the deployment of the components described above in order to highlight the pervasive and embedded nature of most of them. In order to provide a better integrated vision of the overall SM4ALL system, we grouped components in three layers: Pervasive, Composition and User layer.

3.1 Service Composition

Sensors and devices provide information about the environment that is remarkable to offer the best service to the user. Context-aware computing allows the dynamic adaptation of applications and services to guarantee an optimum usage of device and network resources, and to properly handle runtime requirements of applications in order to provide added-value services or to carry out complex activities. A context model should provide an unambiguous definition of the context's semantics and usage. The context-awareness needs this model to cope with context information belonging to heterogeneous context sources.

The user selects a desired goal expressed as a state of the environment (e.g., the house he/she would like to have realized). Later, some automatic techniques are needed to synthesize the right orchestration of services able to satisfy such a goal. These techniques are realized by a composition engine, which is deployed on a special node of the infrastructure, and which is able, after the synthesis, to automatically deploy the orchestration specification on distributed (i.e., embedded into the sensors/devices/appliances/actuators) orchestration engines for execution. The composition engine will discover available services from the repository and compose new services as desired by the customers.

More in details, Figure 2 shows the adopted framework. The synthesis, given the requirements of the target goal and the service descriptions of the available services, produces a skeleton of the process of the composed service realizing the target. Service descriptions comprise both functional features (including the specification of the behaviour, e.g., expressed as a transition system over the operations offered by the service) and non-functional ones, such as QoS, cost, performance, etc. The skeleton of the composite service, together with the non functional requirements of the target service and of the orchestration, is used at execution-time by the orchestration and the monitoring for dynamically coordinating the available services. Specifically, the orchestration is in charge of interacting with the client and scheduling service invocations, whereas the monitoring is in charge of monitoring the matches of the available services with respect to the specification, possibly substituting them with more suitable ones or performing remedial actions.

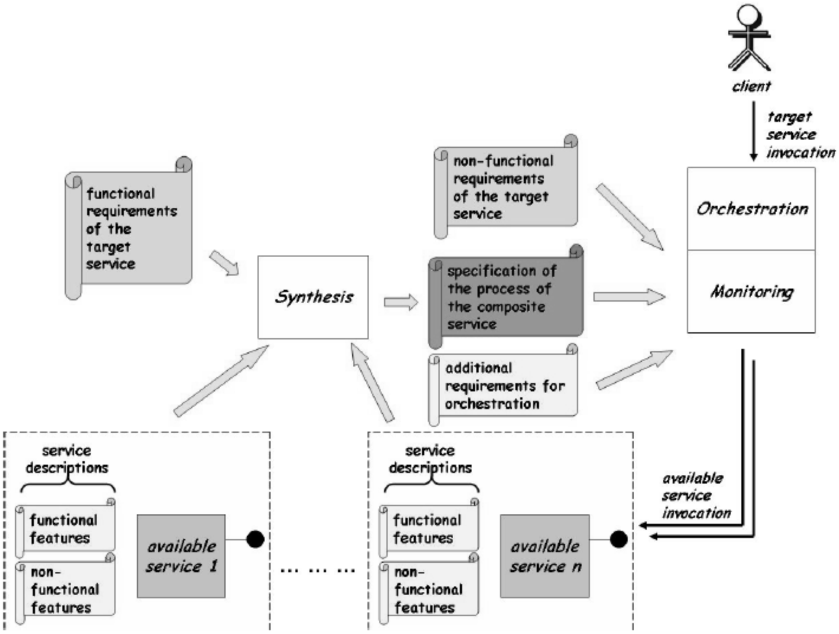
client
target service invocation
functional requirements of the target service
non-functional requirements of the target service
Synthesis
specification of the process of the composite service
additional requirements for orchestration
Orchestration
Monitoring
service descriptions
functional features
non-functional features
available service 1
... ...
service descriptions
functional features
non-functional features
available service n
available service invocation

Fig. 2. The Service Composition Subsystem

3.2 Brain-Computer Interaction

Brain-Computer Interfaces (BCIs) allow individual people to communicate by detecting the user's neural activity, thus with no involvement of his/her muscles [28]. In fact, such systems are considered the only possible aid for persons with severe motor disabilities. Compared to other user interfaces, a BCI allow a reduced communication speed (up to some tens of bits per minute).

A BCI relies on the automatic detection of the user's intent, based on the classification of patterns of her brain waves. In SM4ALL, we are considering 4 steps to achieve such a detection:

Collection. Bioelectrical signals are collected from the surface of the scalp using electrodes, whose number ranges from 2 (simple applications) to 128 (brain mapping studies). These signals, whose amplitude is just a few microvolts, are amplified, digitized and sent to a processor.

Processing. Relevant features are extracted from bio-signals. Processing may consist in averaging over a few repetitions of the same response of the brain to an external stimulus (as in the case of BCIs based on P300), or in the analysis of spectral properties of the electroencephalographic signal (as in the case of BCIs based on sensorimotor rhythms).

Translation. This is a two-step process. First, features of the biological signal are combined (linearly or nonlinearly) into a logical signal, which may either be an analogical signal (e.g., a degree of displacement from baseline values), or a discrete output (e.g., an actual classification). Second, the logical signal is further transformed into a semantic signal, which is meaningful for the application control interface (e.g., in the case of a computer application, how many pixels a cursor should be displaced, which command was selected in a menu, etc.).

Application Control. Semantic symbols are finally translated into physical controls for the actual application, which may consist of, for instance, a computer program, an assisting device, a robot, or a whole a domotic enviromnemt.

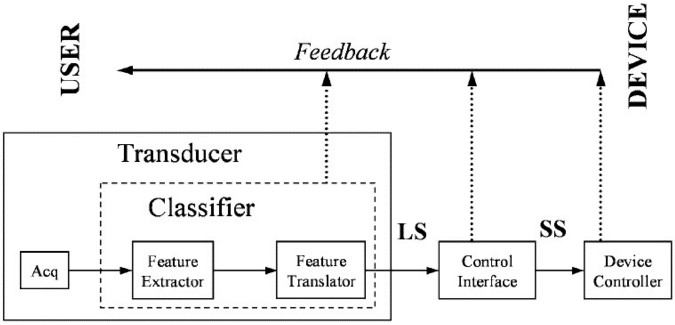

Fig. 3. Functional model of a BCI system [29]

The above steps are supported by a typical BCI architecture (see Figure 3, [30,31]) consisting of a commercial electroencephalographic (EEG) system connected to a personal computer running software that implements, processes, interfaces and communicates with the applications. An EEG can record a positive deflection in voltage at a latency of roughly 300 ms. This event related potential (ERP) is named *P300 (P3) potential* wave and is typically measured most strongly by the electrodes covering the parietal lobe. The presence, magnitude, topography and time of this signal are often used as metrics of cognitive function in decision making processes. While the neural substrates of this ERP still remain hazy, the reproducibility of this signal makes it a common choice for psychological tests in both the clinic and laboratory.

Figure 4 shows the current prototypal version developed in Fondazione Santa Lucia. The figure depicts a user who controls home appliances using a Brain Computer Interface. Electrodes fixed on the user's head measure brain potentials, which are processed by a PC (not shown). The visual interface shows an icon based cascading menu. When a certain icon is flashing and a P300 potential is recognized, that means the user is concentrating on it.

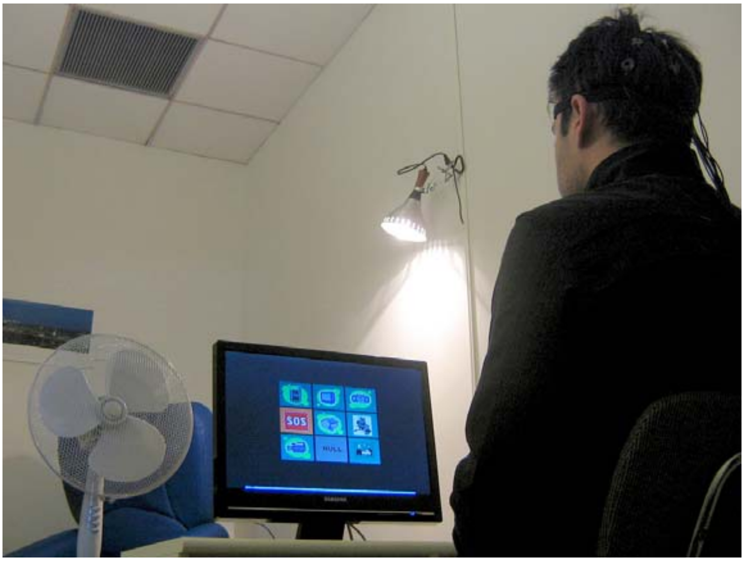

Fig. 4. A prototype developed in Fondazione Santa Lucia

After the command is recognized by the BCI, a command is sent to the target switch to enable/disable specific domotics features (light, fan, motorized armchair, etc).

In particular, the BCI system currently utilized in the SM4ALL project is composed by the following parts:

Acquisition hardware. A portable 8-channel EEG amplifier (g.Mobilab, gTec GmbH, Austria) collects brain potentials from an array of electrodes mounted on the user's head by means of an elastic cap. The signal is amplified, digitized and preprocessed (e.g., bandpassed) and transmitted wireless over a Bluetooth connection.

Processing hardware/software. Feature extraction and translation is performed by means of the BCI2000 software [32] running on a portable PC. Feature extraction consists in the analysis of short (500 ms) segments of EEG following each cue on the Control Interface screen (see below), in order to detect whether a P300 potential is present. P300 reveals that the user was in fact concentrating on that specific cue, thus detecting his intention.

Control interface. The control interface (CI) prompt the user with a set of possible choices, and delivers the stimulation (cue) which may trigger a P300 wave. The CI builds the current set of choices based on: (i) previous selections, thus acting as a multi-layer menu, and (ii) information from the SM4ALL architecture (through the Application Interface), which holds the user profile and the current state of the environment.

In SM4ALL, a bi-directional interface provides connection between the BCI and the rest of the system. From the BCI point of view, it provides information about the state of the environment; this information is used by the Control Interface to build up the set of possible selections, which is thus dynamically linked to the most probable choices (i.e., the possible target that the user would like to reach and the system realized by composing and orchestrating the services).

As for the user interaction, in general she will be able to interact with the house through many interfaces, either centralized (e.g., in a home control station) or distributed and embedded in specific interface devices. Specifically, the BCI system will provide interfaces particularly suited to some specific categories of users (disabled, etc.), allowing them to exploit almost all services offered by the platform.

Concerning the users, we have to characterize their roles (end users, doctors, relatives, etc.), their needs and capabilities, and their usual behaviour, in order to activate the right UI functionalities and to suitably adapt the interface. As for the system, a formal description of available devices, in terms of I/O operations and bandwidth is needed, together with a characterization of the environment hosting the user, in terms of topology, installed sensors, people and object locations, equipment (TV, radio, phone, etc.). In this way, the system is aware of the actual user role, needs, and capabilities, and knowing the environment and the installed sensors and devices, it is able to compute the actual state (e.g., temperature value, end user watching the tv, phone ringing, etc.), selecting the applicable commands and controls and presenting them to the user according to her role and capabilities and the installed UI I/O channels. As an example, when the phone rings, the system activates an icon on the PC screen interface and two flashing lights, one in the kitchen and another one near the phone.

Adaptivity is the key feature to allow an effective interaction between user and the system through a low-speed communication channel like BCI. The adaptation involves both the way in which the interface commands are presented to the user (e.g., large icons, pop up menus, physical buttons or switches, etc.) and their order (as an example, the command *open the door* is presented as a first choice when the door is ringing). This implies to build and instantiate a user profile able to capture user preferences and capabilities and an environment profile, corresponding to the house status (temperature, light, phone/door ringing, and so on). Moreover, a set of parameters/constraints will model some relationships among users, interface, and environment. As an example, it could be possible to pose some temporal constraints on the maximum time of action completion (e.g., the answering time for an incoming phone call must not exceed 10 seconds) or to model the fact that the average selecting time of an icon using a low-speed communication channel, for a given user, is 4.5 seconds (and variance is 1.4 seconds). Such figures allow to define some quality metrics and give precise indications and constraints on how to arrange the interface command order (e.g., putting the *answer the phone* command on the third menu level may result in an answering time exceeding the 10 second constraint). On the other end, the adaptation mechanism should not be invasive, allowing the user to quickly switch the interaction to a prefixed, static, behavior. This is quite useful when contrasting happened situations unforeseen and the user wants to override the automatic system choices. Roughly speaking, the system should allows to associate each user to a static and optimized interface and to a set of rules/metrics allowing to adapt the static structure to the context and, at same time, leaving the user free to switch between the two modalities at any time.

4 Conclusions

In this paper we have outlined the SM4ALL project, which aims at studying and developing a pervasive service oriented architecture, in which the collaboration of services (either software-based or human-based) seamlessly allow users (including those ones with disabilities) to continuously interact with the environment.

The project, recently started, is expected to follow a User-Centered Design approach, in which continuous interactions with concrete users is pursued through the development of mock-ups, prototypes, etc. till the final system. Concurrently, a deep analysis over available sensors/actuators/devices/appliances and their capabilities to host services is undertaken, in order to better characterize the possible performance level that may be obtained.

Acknowledgements. The work is partly supported through the FP7-224332 SM4ALL project. The authors would like to thank the other project partners for useful discussions, and BTicino for providing the hardware equipments currently used in some experimentations.

References

1. Mynatt, E., Melenhorst, A., Fisk, A., Rogers, W.: Understanding user needs and attitudes. IEEE Pervasive Computing 3, 36–41 (2004)
2. Roberts, J.: Pervasive health management and health management utilizing pervasive technologies: Synergy and issues. The Journal of Universal Computer Science 12, 6–14 (2006)
3. Hauptmann, A., Gao, J., Yan, R., Qi, Y., Yang, J., Wactlar, H.: Automatic analysis of nursing home observations. IEEE Pervasive Computing 3, 15–21 (2004)
4. Ross, D.: Cyber crumbs for successful aging with vision loss. IEEE Pervasive Computing 3, 30–35 (2004)
5. Joseph, A.: Successful aging. IEEE Pervasive Computing 3, 36–41 (2004)
6. Consolvo, S., Roessler, P., Shelton, B., LaMarca, A., Schilit, B., Bly, S.: Technology for care networks of elders. IEEE Pervasive Computing 3, 22–29 (2004)
7. Helal, S., Mann, W.C., El-Zabadani, H., King, J., Kaddoura, Y., Jansen, E.: The gator tech smart house: A programmable pervasive space. IEEE Computer 38, 50–60 (2005)
8. Tuecke, S., Foster, I., Frey, J., Graham, S., Kesselman, C., Maquire, T., Sandholm, T., Snelling, D., Vanderbilt, P.: Open service grid infrastructure (2003)
9. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services. Concepts, Architectures and Applications. Springer, Heidelberg (2004)
10. Michalowski, M., Ambite, J.L., Knoblock, C.A., Minton, S., Thakkar, S., Tuchinda, R.: Retrieving and semantically integrating heterogeneous data from the web. IEEE Intelligent Systems 19, 72–79 (2004)

11. Wu, D., Parsia, B., Sirin, E., Hendler, J.A., Nau, D.S.: Automating DAML-S Web Services Composition Using SHOP2. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 195–210. Springer, Heidelberg (2003)
12. Blythe, J., Ambite, J.L. (eds.): Proceedings of ICAPS 2004 Workshop on Planning and Scheduling for Web and Grid Services. ICAPS (2004)
13. Cardoso, J., Sheth, A.: Introduction to semantic web services and web process composition. In: Cardoso, J., Sheth, A.P. (eds.) SWSWPC 2004. LNCS, vol. 3387, pp. 1–13. Springer, Heidelberg (2005)
14. Bultan, T., Fu, X., Hull, R., Su, J.: Conversation specification: a new approach to design and analysis of e-service composition. In: Proceedings of the Twelfth International World Wide Web Conference, WWW, pp. 403–410 (2003)
15. Pistore, M., Traverso, P., Bertoli, P.: Automated composition of web services by planning in asynchronous domains. In: Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling (ICAPS 2005), pp. 2–11 (2005)
16. Gerede, C.E., Hull, R., Ibarra, O.H., Su, J.: Automated composition of e-services: lookaheads. In: Service-Oriented Computing - ICSOC 2004, Second International Conference, Proceedings, pp. 252–262 (2004)
17. McIlraith, S.A., Son, T.C.: Adapting golog for composition of semantic web services. In: Proceedings of the Eights International Conference on Principles and Knowledge Representation and Reasoning (KR 2002), Toulouse, France, April 22-25, pp. 482–496 (2002)
18. Berardi, D., Calvanese, D., Giacomo, G.D., Lenzerini, M., Mecella, M.: Automatic composition of e-services that export their behavior. In: Orlowska, M.E., Weerawarana, S., Papazoglou, M.P., Yang, J. (eds.) ICSOC 2003. LNCS, vol. 2910, pp. 43–58. Springer, Heidelberg (2003)
19. Berardi, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Mecella, M.: Automatic service composition based on behavioral descriptions. International Journal of Cooperative Information Systems (IJCIS) 14, 333–376 (2005)
20. Berardi, D., Calvanese, D., De Giacomo, G., Mecella, M.: Composition of services with nondeterministic observable behavior. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 520–526. Springer, Heidelberg (2005)
21. Muscholl, A., Walukiewicz, I.: A lower bound on web services composition. In: Seidl, H. (ed.) FOSSACS 2007. LNCS, vol. 4423, pp. 274–286. Springer, Heidelberg (2007)
22. Giacomo, G.D., Sardiña, S.: Automatic synthesis of new behaviors from a library of available behaviors. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), pp. 1866–1871 (2007)
23. Sardina, S., Patrizi, F., De Giacomo, G.: Behavior composition in the presence of failure. In: Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2008) (2008)
24. Hull, R.: Web services composition: A story of models, automata, and logics. In: Proceedings of the IEEE International Conference on Web Services (ICWS 2005), Orlando, FL, USA, July 11-15 (2005)
25. Contenti, M., Mecella, M., Termini, A., Baldoni, R.: A Distributed Architecture for Supporting e-Government Cooperative Processes. In: Böhlen, M.H., Gamper, J., Polasek, W., Wimmer, M.A. (eds.) TCGOV 2005. LNCS, vol. 3416, pp. 181–192. Springer, Heidelberg (2005)
26. Catarci, T., de Leoni, M., Marrella, A., Mecella, M., Salvatore, B., Vetere, G., Dustdar, S., Juszczyk, L., Manzoor, A., Truong, H.: Pervasive Software Environments for Supporting Disaster Responses. IEEE Internet Computing 12, 26–37 (2008)

27. Aiello, M., Dustdar, S.: Are our homes ready for services? a domotic infrastructure based on the web service stack. Pervasive and Mobile Computing 4, 506–525 (2008)
28. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. Clinical neurophysiology 113, 767–791 (2002)
29. Bianchi, L., Quitadamo, L.R., Garreffa, G., Cardarilli, G.C., Marciani, M.G.: Performances evaluation and optimization of brain computer interface systems in a copy spelling task. IEEE Transaction on Neural Systems and Rehabilitation Engineering 15, 207–216 (2007)
30. Mason, S.G., Birch, G.E.: A general framework for brain-computer interface design. IEEE Transaction on Neural Systems and Rehabilitation Engineering 11, 70–85 (2003)
31. Mason, S.G., Bashashati, A., Fatourechi, M., Navarro, K.F., Birch, G.E.: A comprehensive survey of brain interface technology designs. Annals of Biomedical Engineering 35, 137–169 (2007)
32. Schalk, G., McFarland, D.J., Hinterberger, T., Birbaumer, N., Wolpaw, J.R.: BCI2000: a general-purpose brain-computer interface (BCI) system. IEEE Transaction on Neural Systems and Rehabilitation Engineering 51, 1034–1043 (2004)