A Prototype of Context Awareness Architecture for Ambience Intelligence at Home

Pablo A. Haya[1], Abraham Esquivel[1], Germán Montoro[1], Manuel García-Herranz[1], Xavier Alamán[1], Ramón Hervás[2] and José Bravo[2]

[1]Universidad Autónoma de Madrid - Escuela Politécnica Superior
c/ Francisco Tomás y Valiente 11
Madrid 28049 – SPAIN
Pablo.Haya@uam.es
[2]Universidad de Castilla – La Mancha - Escuela Superior de Informática
Paseo de la Universidad, s/n
Ciudad Real 13071 – SPAIN
rhlucas@inf-cr.uclm.es - Jose.Bravo@uclm.es

Abstract. Ambient intelligence is an interdisciplinary field that brings together ubiquitous computing, intelligent environments and context-awareness technologies. This paper describes a context-aware prototype for home environments, and the underlying concepts that we have developed. We propose a standardised representation of the environment context, in such a way that the integration of new developments (i.e. smart agents) is made simpler. A middleware that supports this standardised representation has been developed. This paper makes an especial emphasis on a proposed ontology that deals with information flows in the smart room. As an example, a ubiquitous broadcasting digital audio application has been implemented and tested in a real environment. Identification-based services by RFID sensor fusion are presented as a complement to the smart room.

1 Introduction

Ambient Intelligence is a new vision that merges the achievements of existing technologies such as ubiquitous computing [1], intelligent environments [2] and context-aware computing [3]. Ambient Intelligence promotes a user-centred design and the use of highly interactive systems that will steer the pervasive deployment of these technologies into everyday life.

During the last years Ambient Intelligence has achieved relevance in the European Union agenda. For example, the Sixth Framework Programme (FP6) includes this issue as one of its priorities [4]. Moreover, industrial initiatives are funding projects and research centres [5, 6, 7] dedicated to these technologies.

These emerging technologies enable new opportunities and challenges [8]. We are especially interested in those that have an impact in home environments. In particular, we are focusing on how context information can enhance user interaction within a smart home environment. We propose that the context gathered from the environment should be collected in a common model shared by every context-aware application [9]. This model should include the available resources and the relations among them. In this direction, we have implemented a middleware between the model of the smart home and the physical world in such a way that changes in the model are immediately reflected into the real world, and vicecersa.

There are several groups researching in how to model context as a web of relations among concepts, such as the Cobra project [10], Henricksen et al [11], Aire project [12] and the initiative "pervasive semantic web" [13]. Our proposal is specifically focused on home environments and represents explicitly the information flows in such applications.

This paper makes an emphasis in the part of our model that deals with information flows among environment resources [14], and in how these flows are modified depending on user context. One of the goals of our smart environment is to achieve continuous access to services while the user moves through the rooms of the house (for instance, a ubiquitous audio distribution service). To do this, our approach represents explicitly each connection between streaming (audio and video) resources and provides a simple mechanism to manage them.

A laboratory has been converted into a real home environment to test our prototypes in a similar way than projects such as the Adaptative House [15], the AwareHome Project [16] and The Intelligent Room Project [17].

The first section of the paper shows the core of the model, with a focus on how to represent the management of information flows. The following section explains a middleware that has been developed to support this model. This middleware stores the model that represents home resources and users, and provides the functionality that links the model to the real world. The next section describes a context-aware application that has been implemented to test the proposed model and the implemented middleware. Finally, the identification-based services are studied, focussing in the mosaics of information.

2 Modelling the environment

We have proposed a hierarchical classification of the relevant concepts for Ambient Intelligence in home environments. This section presents the part of this ontology that deals with information flows at home environments. The following sections will explain how these concepts are implemented and managed.

In the proposed ontology each concept[1] is represented by a class name and a set of properties. Each property has a value that can be a literal or another concept. When the value of a property is a concept, a relation is established between the two concepts. This relation is considered as having an explicit "direction", that is, in case it holds, the inverse relation has to be also explicitly asserted.

We have adopted Dey's definition of *context* to develop our model for a home intelligent environment [18]. The model starts with four main concepts: **Person**, **Place**, **Resource** and **Information**. A *Resource* is every component that can be used

[1] The terms *class* and *concept* are used indistinctly in this paper.

by a *Person* (or other *Resource*) located in some *Place. Information* can be either a user preference (see below for details) or an environmental variable, such as the current sound level or luminosity. The description of a *Resource* always includes the following set of relations: *handles, is-handled-by, is-composed-by, allowed-user* and *is-located-in*. The *handles* relation establishes that a resource is being used by other resource. Reciprocally, the relation *is-handled-by* defines which *Person* or *Resource* is controlling a given resource. The *composed-by* property allows describing a *Resource* as composed of other resources. The *allowed-user* property defines the access policy. Finally, the *is-located-in* property represents in which place the resource is located.

The *Resource* concept is refined by the following four concepts: **Device**, **Document, Application,** and **Processor**. A *Device* represents a physical object (i.e. a microphone, a light bulb, a speaker). Each *Device* always includes the *status* property that, at least, indicates if the device is turned on or off. We have split the device category into **Output** and **Input** devices, depending on whether they produce or consume information. Output devices include video and audio consumers -such as screens and speakers-, as well as mechanical actuators such as door locks, blinds, lights and home appliances. On the other side, Input devices comprise mice, keyboards, video and audio sources -such as microphones and video cameras-, and physical sensors. *Device* classes represent simple devices; it is possible to define composite devices by means of the *is-composed-by* relation. Thus, a TV set is composed by an instance of Screen class and two or more instances of Speaker class. Finally, *Documents* and *Applications*, do not correspond to tangible objects. The first class represents digital files that store some information, while the second class represents computational services. Therefore, devices, documents and applications represent existing resources. On the other side, a *Processor* denotes a capability, something that can be performed by a resource. This allows, for example, distinguishing between the sensing capability and the sensor itself. As we will show latter, a device can be composed of one o more processors.

We propose two sub-classes for the **Processor** class: **Source Processor** and **Sink Processor**. A Source Processor generates and provides information, while a Sink Processor consumes it. Each source processor includes the *is-connected-to* property. This property establishes a relation between the source and a sink, which models the stream of information that is flowing between the two processors. At this level of description the model does not specify the nature of the information being exchanged. In particular, there is no distinction between synchronous and asynchronous information. So, the same relation can be used to model a temperature sensor feeding a thermostat and an audio server broadcasting to several audio players. The maximum number of connections supported by the source and by the sink is bounded by the property *max_number_of_connections*.

In order to avoid forbidden or useless connections, the *allowed-connection-to* relation is defined. A source can be connected with a sink if they are related by means of the *allowed-connection-to* property. The classes **Source** and **Sink Processor** can be extended by sub-classes that represent concrete capabilities, such Audio Source, Video Source, Audio Sink, etc.

2.1 User preferences

A preference states how a user would like to experience the environment. Each user preference defines a particular configuration of one or more environment resources. Preferences take into account context, in particular, time and user location.

A user preference is stored as a tuple that contains at least the following fields: type, name, location, time and value. The type field provides a classification for user preferences, according to the domain to which the preference will be applied. Examples of values for this filed are lighting, heating&cooling, audio and video. The name field identifies each preference. Given a preference type, the name is guaranteed to be unique. This name points to the resources that will be affected. If the name is a resource, the preference will be applied to this particular resource. Alternatively, the name may indicate a generic user preference, which will affect several target resources. The location field sets the room where the preference is applicable, and the time field establishes the time interval when the preference is valid. Finally, the value field stores the action that will be applied if the user preference becomes active.

For example, Table 1 shows two user preferences related to room ligthing. The first one represents a generic user preference for natural illumination of the room B-403 from 09:00 a.m. to 08:00 p.m. It will involve the status of several devices. The second file is a preference, related to the same room but in a different time interval, which establishes that a concrete light resource, called res: lamp1, should be turned on.

Type	Name	Location	Time	Value
Lighting	natural	B-403	09:00 – 19:00	yes
Lighting	res:lamp1	B-403	19:00 – 21:00	On

Table 1. Examples of preferences

Context-aware applications read the preferences for a particular user, and when the user context changes, check whether any preference is active. However, context-aware applications can freely interpret it. For example, a lighting manager can decide to open the blinds in response to the first user preference, while a different manager may ignore it because there are not automatic blinds in a particular room.

3 Supporting the model

3.1 Context layer

The proliferation of communication networks and protocols complicates the seamless integration of environment devices, as we discuss in depth in [19]. We propose a "context" layer as the glue to achieve the required synergy among computing entities in order to constitute a home smart environment. We believe that a global "world model", although implemented in a distributed way, is the best approach to achieve complex interactions among devices in order to provide pro-active services to the user.

The context layer provides:

1. A repository where the instances of the ontology are stored.
2. A single interface that abstracts from the communication details of the various physical devices.

The context layer implementation lies on a global data structure: a *blackboard* [20]. This blackboard stores a model of the world, where all the relevant information related to the environment (including users) is represented. In particular, the blackboard also holds a representation of the flow of information among the physical devices (microphones, speakers, cameras, displays, etc.).

Information from the blackboard is used by pervasive devices to understand the context and to adapt to it. For example, the number of persons in the room, the desired audio volume and the status of several physical devices (lights, heating, video/audio displays) are represented in the context layer.

For each multimedia resource, an entity is defined in the blackboard. When two multimedia resources have to be connected, the corresponding relation is added to the blackboard. Then, both resources configure themselves to satisfy the new situation. A similar behaviour occurs when the relation is removed from the blackboard and the flow of multimedia information has to stop.

Each blackboard is a server that can be accessed using client-server TCP/IP protocols. HTTP has been chosen as the application protocol because it is simple and widely spread. To exchange information between smart agents and a blackboard server an XML-compliant language is employed.

3.1.1 Resolving conflicts

In the same direction than Johanson and Fox [21], who propose an event heap to store and retrieve context changes, we propose a *command heap* to manage conflict resolution. Command heaps are necessary when two or more smart agents can modify the same entity of the blackboard model; for instance, when two applications are sending contradictory commands about the state of the lights. A command heap is a pre-emptive prioritized command queue. Each command represents the willingness of a smart agent to change the value of a property or a relation in the blackboard. Each command is composed of an identifier, a priority, an expiration time and the owner's identification. Whenever a command arrives, it is stored in the command heap. If there is no other command with higher priority, it will become the active command. Otherwise, it will be placed in the corresponding position of the heap. When a command is activated, the blackboard forwards it to the corresponding device. This command will remain active at the top of heap while its expiration time is valid and the owner does not remove it. In both cases, the new highest priority command will become the active command.

When a user preference has to be applied, it is operated as any other command and sent to the corresponding heap. The priority of a user preference is determined using the next rules:

1. By default, every preference receives the same priority. This implies that preferences that arrive first are activated first. So, no user prevails over other users.

2. If a user has an *owned-by* relation with the corresponding resource, her/his preferences will get the highest priority.
3. In certain special cases, preference conflicts are resolved by prioritizing the most restrictive preference (for example, when resolving the conflict among preferences about the volume for the ambience music, the preference with the lower volume will prevail).

4 Putting all together

This section describes a demonstrator that has been developed to test the model and the context middleware above described.

4.1 Environment description

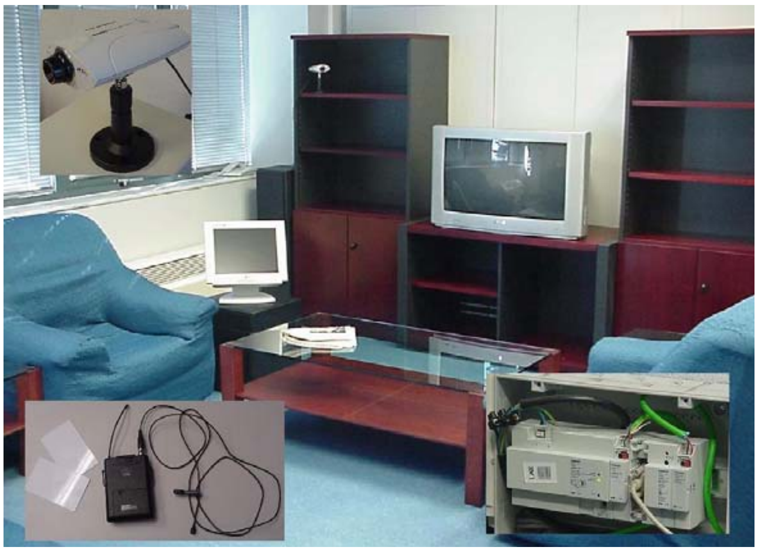


Figure 1. View of the laboratory B-403

Our testbed is composed by three rooms: two laboratories (named B-403 and B-207) and one office (named B-420). B-403 is equipped as a household living-room. A mix of heterogenous devices can be found, such as sensors, actuators, a TV set, two hi-fi speakers, a DVD player and several flat monitors (see Figure 1). Several hidden PCs are spread through the room. These PCs hold context-aware smart agents that monitor the context changes and control the environment resources. This laboratory is shared by many people, and it is used as a TV room and as a meeting room. Laboratory B-207 is a shared office for PhD students. Each student has her/his own PC and two speakers. Finally, B-420 is the office of their PhD advisor. He has the same equipment than his students.

There is one blackboard that stores the model for the three rooms. On the other hand, each user saves her/his preferences in her/his own computer.

4.2 ContextAudio: an example of context-aware application

Once we have implemented the Intelligent Environment above described, we are currently experimenting with context-aware agents that incorporate Ambience Intelligence characteristics. One example of such applications is the *ContextAudio* smart agent. This agent implements audio distribution that adapts to user's context and preferences. The audio resources are audio servers and speakers. Audio servers are PCs with a digital radio and a CD player. Speakers are attached to PCs, and are also controlled by an associated software driver.

Each audio server is represented in the blackboard by a device composed of an audio source. Each speaker is also represented in the blackboard by a device composed of an audio sink. Whenever the *is-connected-to* relation is established between an audio source and an audio sink, the corresponding speaker starts to play the audio generated by the audio source. If this relation is removed, the speaker driver turns the speaker off. The connection graph between sources and sinks is stored at the blackboard.

Whenever a user switches on his/her audio system, a session is created that will be active until it is explicitly terminated by the user. Each session is managed by a *session manager*.

Each session is attached to an audio server. The session manager tries to achieve that the user be listening the audio produced by this server. To do this, the session manager keeps a connection relation between the audio server and the nearest speaker to the user. Moreover, the session manager tries to maintain session settings (speaker volume, balance, radio channel …) as close as possible to user preferences, taking into account the following context information:

– **User's location**: the session manager is subscribed to changes in the user location, so the audio stream can follow the user from room to room. When the user leaves a room, the current speaker is turned off, and when s/he enters a room, one of its speakers, if any, is turned on. The session manager adds and deletes relations between audio sources and sinks to achieve these effects.
– **Room sound level**: this context variable defines the background audio volume level the room occupants want. The volume of the speakers must adjust in such a way that sound level will never be louder than this parameter. Each room occupant can establish a maximum sound level that s/he is willing to tolerate. The current desired room sound level will correspond to the minimum of the sound levels established by all the occupants. This is achieved by the conflict resolution mechanisms above described.
– **Room's owner(s)**: If there are several users in the room, all of their commands and preferences should be taken into account. However, if a room owner is present, her/his preferences and her/his session manager commands will have more priority than these from other occupants. This is also achieved by the conflict resolution mechanisms.

Finally, the ContextAudio agent also takes into account other user preferences, such as the audio server status, the favourite radio channel and the speaker volume. The user can decide the rooms where s/he want to listen the audio stream and s/he can choose a favourite radio channel that it is selected at start up, and can establish her/is prefered speaker volume.

4.3 Identification-Based Services

Another application currently implemented in our prototype is based in identification processes. It offers user-aware services with no requirements of explicit interactions, but just by wearing a RFID tag. As a result of simple actions such as entering a room or approaching to a monitor, the system offer services to the users. Some of these are access control, location management, inventory (for objects) and the most important for us, a visualization service named "mosaics of information".

In this technology three kinds of objects are clearly differentiated: the reader or transceiver has the job of handling all readings and writings, the antenna supplements these processes, making it possible to read and write at a distance and finally, the tag or transponder (smart label) is carried by the users and objects to be identified, with the added advantage of being able to use the dynamic information stored in it as a context interaction. This small device contains a circuit that reflects the electromagnetic energy emitted continuously by the reader, reading and writing the information contained in them. Figure 2 shows some types of RFID sets.

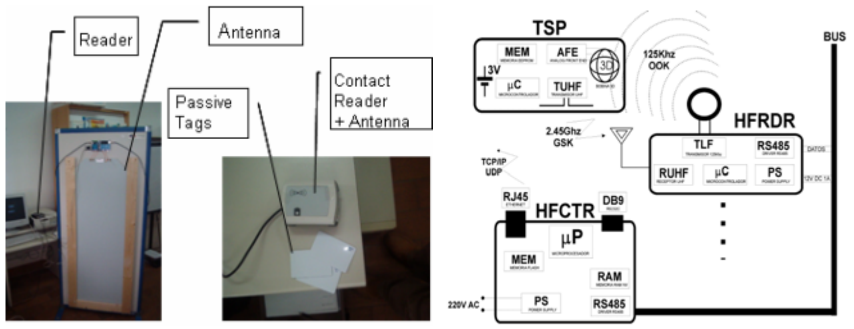
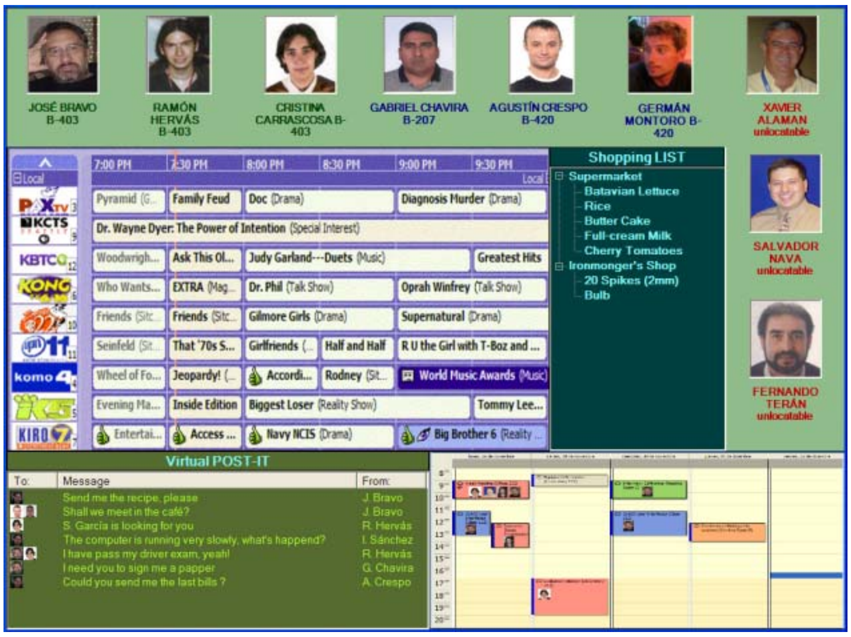


Figure 2. RFID devices

Once these devices have been installed at home, some available services are offered:

- **Access Control**: This service allows users entering at home. When the user is near the door, the door is unlocked, and the alarms are automatically disabled if there is nobody at home. Also, when the last person leaves home, the alarms are automatically activated.
- **User's location**: It is possible to know the location of everyone at home, while no explicit interaction with the system is needed.
- **Inventory:** Every object at home can be controlled with a RFID tag, activating the alarm for robbery.
- **Mosaic of Information**: the system shows information in the monitors according to the user' profile and the information contained in the tag. There are some

important aspects to be considered: the mosaic of information itself, the places in which this mosaic contain objects of information, the time for every object remains into the mosaic, the cycles to present information, etc. See Figure 3.



7:00 PM	7:30 PM	8:00 PM	8:30 PM	9:00 PM	9:30 PM
Pyramid (G...	Family Feud	Doc (Drama)		Diagnosis Murder (Drama)	
Dr. Wayne Dyer: The Power of Intention (Special Interest)					
Woodwrigh...	Ask This Ol...	Judy Garland---Duets (Music)			Greatest Hits
Who Wants...	EXTRA (Mag...	Dr. Phil (Talk Show)		Oprah Winfrey (Talk Show)	
Friends (Sitc...	Friends (Sitc...	Gilmore Girls (Drama)		Supernatural (Drama)	
Seinfeld (Sit...	That '70s S...	Girlfriends (...	Half and Half	R U the Girl with T-Boz and ...	
Wheel of Fo...	Jeopardy! (...	Accordi...	Rodney (Sit...	World Music Awards (Music)	
Evening Ma...	Inside Edition	Biggest Loser (Reality Show)			Tommy Lee...
Entertai...	Access ...	Navy NCIS (Drama)		Big Brother 6 (Reality ...	

Shopping LIST
- Supermarket
  - Batavian Lettuce
  - Rice
  - Butter Cake
  - Full-cream Milk
  - Cherry Tomatoes
- Ironmonger's Shop
  - 20 Spikes (2mm)
  - Bulb

Virtual POST-IT

To:	Message	From:
	Send me the recipe, please	J. Bravo
	Shall we meet in the café?	J. Bravo
	S. García is looking for you	R. Hervás
	The computer is running very slowly, what's happend?	I. Sánchez
	I have pass my driver exam, yeah!	R. Hervás
	I need you to sign me a papper	G. Chavira
	Could you send me the last bills ?	A. Crespo

Figure 3. Mosaic of Visualization

In this mosaic we can see different information items placed strategically according the identification of people in the room. On the top, the location service is activated. Below that, the TV Guide for all habitants of the home and the Shopping list are presented. Finally, at the bottom, the messages and schedule are shown.

The RFID technology offers support to daily activities at home. However, there are some cases where interaction supported by the identification process need to be complemented. When that happens, we are attempting to merge the identification with other kind of inputs. To do this we have located some sensors in the monitor, that are activated by passing a hand across them quickly or by leaving our hand steady near the sensor for a few seconds.

Aknowledgements

This work has been partially funded by the 2005-2006 SERVIDOR project (PBI-05-034) from Junta de Comunidades de Castilla-La Mancha, and by the Ministerio de Ciencia y Tecnología in Mosaic Learning project 2005-2007 (TSI2005-08225-C07-07).

References

1. Weiser, M.: Some computer science issues in Ubiquitous Computing. Communications of ACM, 36(7). July (1993) 75-84
2. Coen, M.: Design Principles for Intelligent Environments. Proceedings of the AAAI Spring Symposium on Intelligent Environments. (1998)
3. Chen, G., Kotz, D.: A Survey of Context-Aware Mobile Computing Research. Technical Report, TR2000-381. Darmouth College (2000)
4. The priorities of the Sixth Framework Programme 2002 – 2006. RTD Info Magazine for European Research. Special Edition. ISSN 1023-9006 November (2002)
5. http://www.philips.com/Assets/Downloadablefile/LEAFLET8x-1504.pdf
6. http://www.ibm.com/websphere
7. http://www.cooltown.com
8. Satyanarayanan, M.: Pervasive Computing: Vision and Challenges. IEEE Personal Communications, 8(4). August (2001) 10-17
9. Alamán, A., Cabello, R., Gómez-Arriba, F., Haya, P., Martínez, A., Martínez, J., Montoro, G.: Using context information to generate dynamic user interfaces. 10th International Conference on Human-Computer Interaction, HCI International 2003. Crete, Greece. June 22-27, (2003)
10. Chen, H., Finin, T., Joshi, A.: Semantic Web in the Context Broker Architecture. In Proceedings of PerCom 2004. March (2004)
11. Henricksen, K., Indulska, J., Rakotonirainy, J.: Modeling Context Information in Pervasive Computing Systems. Pervasive Computing, First International Conference, Pervasive 2002. LNCS 2414. Springer-Verlag (2002) 167-180
12. Peters, S., Shrobe, H.: Using Semantic Network for Knowledge Representation in an Intelligent Environment. In Proceedings of PerCom'03. March (2003)
13. http://pervasive.semanticweb.org
14. Nagel, K. S., Abowd G. D.: Challenges in Developing a Context-Aware Family Intercom. In Proceedings of Computer Human Interaction, CHI (2002)
15. Mozer, M.: The neural network house: An environment that adapts to its inhabitants. In Proceedings of the AAAI Spring Symposium on Intelligent Environments (1998)
16. Kidd, C. K., Orr, R., Abowd, G. D., Atkenson, C. G., Essa, I. A., MacIntyre, B., Mynatt, E., Starner, T. E., Newstetter, W.: The Aware Home: A Living Laboratory for Ubiquitous Computing Research. In Proceedings of the Second International Workshop on Cooperative Buildings, CoBuild'99 (1999)
17. Le Gal, C., Martin, J., Lux, A., Crowley, J. L: SmartOffice: Design of an Intelligent Environment. IEEE Intelligent Systems, 16(4). July-August (2001) 60-66
18. Dey, A.: Understanding and using context. Personal and Ubiquitous Computing, 5(1) (2001)
19. Haya, P., Alamán, X., Montoro, G.: A Comparative Study of Communication Infrastructures for the Implementation of Ubiquitous Computing. UPGRADE 2(5) (2001)
20. Engelmore, R., Morgan, T.: Blackboard Systems. Addison-Wesley (1998)
21. Johanson, B., Fox. A.: The EventHeap: A Coordination Infrastructure for Interactive Workspaces. In Proceedings of WMCSA 2002. (2002)