Addressing the Users' Diversity in Ubiquitous Environments through a Low Cost Architecture

Tatiana Silva de Alencar[1], Luciano Rodrigues Machado[1], Luciano de Oliveira Neris[2], and Vânia Paula de Almeida Neris[1]

[1] Flexible and Sustainable Interaction Laboratory - Department of Computing – UFSCar - São Carlos, SP – Brazil
[2] AGX Technology - São Carlos, SP – Brazil
{tatidealencar,vania}@dc.ufscar.br, 337790@ead.ufscar.br, neroso@msn.com

Abstract. A ubiquitous environment allows the system to infer the users' needs and preferences, making adaptations to the interface. However, the best way to make such adaptations is still under debate by researchers. This paper proposes an architecture that supports the adaptation of user interfaces in ubiquitous environments according to the users' profiles. The proposed architecture is shown simple and low cost, has low implementation complexity and high extension capability. The user profile data are stored on the user's mobile device for privacy. As the profile is defined by the user, it is expected that the interface adaptation occurs more accurately. A prototype is presented as a proof of concept.

Keywords: Ubiquitous Environment, User Profile, Ubiquitous Accessibility, Context-Aware, Adaptive Interface, Raspberry PI.

1 Introduction

The term ubiquitous computing, defined by [22], is used to describe systems which allow their resources to be available everywhere, in an intuitive and transparent way to the user. To achieve transparency of use, [17] computers should anticipate the user's needs and act proactively to provide appropriate assistance. Systems that have this ability are called context-aware [16]. Therefore, in order to sustain the ubiquitous environment existence, the system must proactively oversee and control the context conditions and make the necessary changes by an adaptation process [21].

One of the inputs for the adaptation process should be the users' needs and preferences [15]. In the literature, most of the works which focus on the user interface adaptation is related with the flexibility between different devices [13, 14, 18], but not on the interface adaptation to different types of users. In addition, it is possible to find researches about accessibility in ubiquitous environments that focus on restricted groups of users [1, 12, 20], without considering the interaction needs of different groups in a more universal and inclusive approach.

C. Stephanidis and M. Antona (Eds.): UAHCI/HCII 2014, Part III, LNCS 8515, pp. 439–450, 2014.

Alencar and Neris [3] have demonstrated that known ubiquitous environments partially meet the interaction requirements of visually impaired people, people with low literacy and people with attention deficit and/or memorization problems. These authors argued that ubiquitous environments must be able to adapt themselves to the diverse users' capabilities, related to the physical and cognitive characteristics, and users' interaction preferences.

This paper proposes an architecture that supports the adaptation of user interfaces in ubiquitous environments according to the users' interaction profiles. In this architecture, the users' characteristics and preferences are mapped into an ontology, and are available to ubiquitous applications in an eXtensible Markup Language[1] (XML) file containing the user profile. Aiming to show that it is not necessary many resources[2] to offer adapted interfaces to supply the users' needs and preferences, the proposed architecture is shown simple and low cost by using a Raspberry PI[3] computer. As other advantage of this architecture, we can mention that the user profile data is stored on the user's mobile device (and not in a web server) for privacy. Backup in private storage media is also provided.

In order to prove the concept, a prototype was developed. Considering the emerging context of smart cities, the prototype is set in a bus stop. In this scenario, for instance, a visually impaired person may find it difficult to take the correct bus. The same can happen to an elderly person. The prototype uses a monitor that displays the bus schedules as well as the next bus identification. The system shows the most suitable user interface considering the user's profile.

This paper is organized as follows: Section 2 summarizes the related work on modeling users' profiles for ubiquitous environments. Section 3 shows how the user's profile is mapped, in our proposal, using an ontology. Section 4 presents the proposed architecture for ubiquitous environments that supports adaptive user interfaces, detailing their modules and communication flow. Section 5 presents the prototype build to develop the concept of the proposed architecture and the feasibility study. Finally, we highlight the contributions and limitations of this work and discuss future works in section 6.

2 Related Work

2.1 Architectures for User Diversity in Ubiquitous Environments

In the literature, a few studies on the development of architectures that support the interface adaptation to different users can be found. The architecture proposed by [11] makes use of sensors in the environment to capture contextual information. Data on

---

[1] `http://www.w3.org/XML/`

[2] Considering that the user may already have a mobile device, the hardware involved in the architecture considers the Raspberry PI (at about US$30.00) and a Bluetooth device (at about US$ 8.00). The software solution is also simple and extensible, making it a low-weight architecture.

[3] `http://www.raspberrypi.org/`

the user's device are also captured and tests are performed during the interaction with the mobile interface to identify some of the user's characteristics (e.g. disability). This information is sent to a server that queries an ontology to determine the best adaptation to the interface. The changes to be made are sent to the user's device in an XML file.

The Project Aura[4] uses an architecture to adapt the ubiquitous environment according to user location and the tasks he is performing [18]. In this architecture, a Context Observer controls the physical environment and the collected information is sent to a Task Manager. Based on information about the environment and context, and the activity being performed, the Task Manager adapts the interface to the user.

Abascal et al. [2] propose an architecture for adapting interfaces for people with disabilities. In this architecture, the user logs into the interface adaptation system through his mobile device. The system consults an ontology in order to identify the user and determine their needs and preferences. Then the system generates an adapted user interface in eXtensible Hypertext Markup Language (XHTML) and sends to the user's mobile device.

The proposed architectures for [11] and [18] have a medium to high cost due to the use of sensors to identify the user context. Besides the cost, the architecture proposed by [18] has a high complexity in their implementation. In the presented architectures [11, 18, 2], the best interface adaptation is determined by the system, according to the user profile. Although these adaptations are guided by the user needs, the user preferences are not taken into consideration. In [11], the user profile is created during the interaction time. Therefore, in each new interaction, all the identification tasks have to be performed again.

2.2 Modeling Users' Profiles

In this paper, user's profile is defined as a set of characteristics that identifies the users' needs and preferences. These characteristics can be physical (e.g., low vision, visual impairment), cognitive (e.g., attention deficit, memory problems), level of literacy, interests (e.g., news, social networks, books), among others.

In a ubiquitous environment, the details that compound the user's profile may come from various sources, such as sensors, social networking profiles, mobile and semantic web technologies. Each of these sources provides details in different data models. To solve this problem, [10] propose the use of ontologies based on the Simple Knowledge Organization System for the Web[5] (SKOS) in order to model the user's profile with data from these different sources. To provide a standard model, SKOS makes use of Resource Description Framework[6] (RDF), a framework that describes web resources. Using RDF allows the user's profile to be shared among different applications in an interoperable way.

---

[4] http://www.cs.cmu.edu/~aura/

[5] http://www.w3.org/2004/02/skos/

[6] http://www.w3.org/RDF/

Intelligent learning environments also receive data from different sources to provide tailored services and resources to users. To model the user's profile in such environments, [5] propose the use of RDF as a standard model of the user's profile. The proposed architecture combines descriptions and performance requirements from a resource to a particular student in RDF format to make the necessary adjustments. The availability of a standard model allows this profile to be shared with other applications.

Aiming to provide tailored interfaces with focus on the end user, [11] propose an architecture that uses the concepts of Pervasive Computing and a representation of user's profile. The user's profile takes into account their disabilities, preferences, experiences and demographic data. This profile is represented by an ontology expressed in OWL. The ontology is updated through preferences and deficiencies details that are passed by the user application and the data captured by sensors, all made available in an XML file. To perform the necessary adjustments, rules are executed crossing captured data from sensors, user's profile and device profile.

Heckmann et al. [7] propose a General User Model Ontology, an ontology represented in Web Ontology Language (OWL) to uniform interpretation of user models. The GUMO ontology divides the dimensions of the user model in three parts: auxiliary, predicate and range. This ontology makes use of the UbisWorld model [6] to identify the dimensions of the basic user model. The UbisWorld allows the user information representation, such as demographics data, interests, psychological and physiological states, personal characteristics etc.

As can be seen in Table 1, the studies cited do not include all the users' needs and preferences in the user's profile templates. In addition, it is observed that there is a tendency towards the use of XML, either in pure form or as a form of writing in other languages or frameworks such as OWL and RDF.

Table 1. User characteristics present in the user's profile templates: (a) [10], (b) [5], (c) [11], (d) [7]

Paper / Users' Characteristics	(a)	(b)	(c)	(d)
Physical	X	X	X	X
Cognitive			X	
Literacy level		X		X
Interests	X	X		X
Interaction preferences			X	X

In our work, the user's profile considers the physical and cognitive characteristics, interests and user's interaction preferences. Furthermore, the proposed architecture makes use of the XML language to describe the user's characteristics and preferences, because of its simplicity and standardization to enable data exchange between applications.

3 Mapping User's Profile Based on an Ontology

Considering that ubiquitous environments must adapt themselves to the different users' characteristics, this paper proposes the use of a more comprehensive user interaction profile that is mapped on an ontology. The GUMO ontology [7] was chosen to model the user profile as it is the most complete among those analyzed (see Table 1).

The classes of the GUMO ontology to be used in modeling or the user profile were chosen based on the application of the UbiCARD technique [4]. The following classes (and their subclasses) were chosen: Basic User Dimensions and Domain Dependent Dimensions. Figure 1 shows the predicates that represent the characteristics, demographics data and user emotional states (Basic User Dimensions).

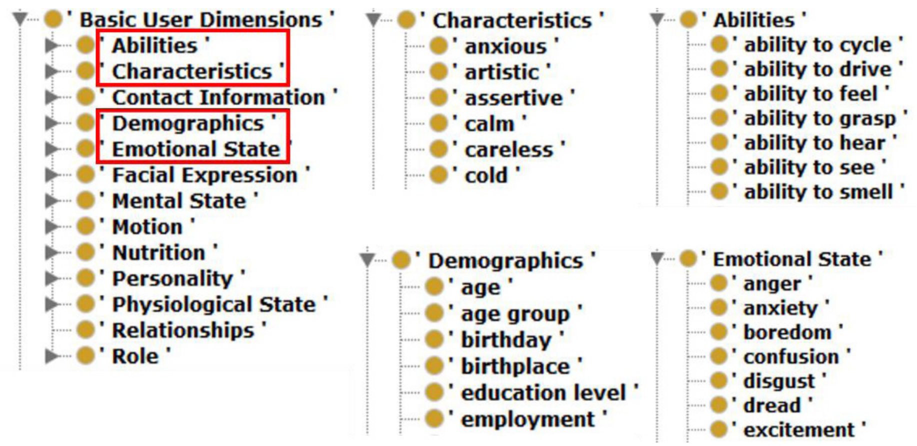

Fig. 1. Predicates that represent the characteristics, demographics data and user emotional states

```
<?xml version="1.0" encoding="UTF-8"?>
<user-profile>
 <statement
  auxiliary = ""
  predicate = ""
  range = ""
  object = ""
  group = ""
 />
</user-profile>
```

Fig. 2. The XML file structure for the user profile representation

The UserML [8] is a markup language for ubiquitous environments divided into two levels: (1) simple XML structure, composed of twenty-five predefined attributes and (2) the ontology that defines the categories. This approach allows applications to use their own ontologies and the UserML language to map them. For this reason, the UserML was chosen as the language to structure XML file that contains the user model profile.

Among the twenty-five attributes of UserML, the following were selected according to the chosen GUMO ontology classes: auxiliary, predicate, range, object, start, end, durability and group (see Figure 2). The attributes start, end and durability are used only for the predicates related to the emotional state.

The next section describes an architecture to support adaptive user interfaces for ubiquitous environments and how an XML file, in the user mobile phone, used to represent the user's interaction profile.

4 An Architecture to Support Adaptive User Interfaces for Ubiquitous Environments

This section describes the proposed architecture for adapting interfaces of ubiquitous environments that focuses on the application adequacy according to the user's profile. In this architecture, the users' characteristics and preferences are mapped into an ontology, and are available to ubiquitous applications in a XML file containing the user profile. As the profile is defined by the user, i.e., it is not inferred/captured by sensors, it is expected that the interface adaptation occurs more accurately. Moreover, the stored profile in the mobile device provides greater privacy to the user. Besides the low cost highlighted in Section 1, the proposed architecture has low implementation complexity and high extension capability.

Aiming to propose a simple and inexpensive architecture, we have adopted a Raspberry PI computer to execute the ubiquitous application. The Raspberry PI is a small computer (as the same size as a credit card) and cheap. It was designed by the Raspberry PI Foundation for children around the world to learn programming. Despite the small size, the Raspberry PI can be used for various tasks and performs as well as a desktop computer.

The proposed architecture is divided into two main areas – user's mobile device and ubiquitous application - as we can see in Figure 3. The user's mobile device contains the XML file describing the profile based on GUMO ontology. The ubiquitous application has the adaptation rules to be applied in the interface according to the user's profile. The data exchange between these devices is controlled by an application in the user's mobile device and in the Raspberry PI. The mobile device application sends the XML file containing the user's profile and the ubiquitous application reads this file and obtains information about the user's needs and preferences.

The communication between the user's mobile device and the ubiquitous application is established via Bluetooth[7]. The data exchange between these devices is

[7] `http://www.bluetooth.com/Pages/Bluetooth-Home.aspx`

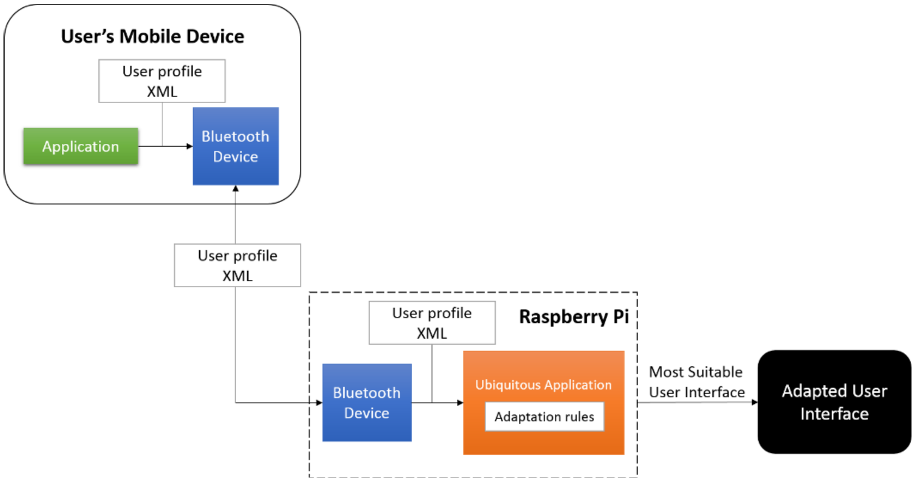

Fig. 3. Proposed architecture for adapting user interfaces in ubiquitous environments

controlled by an application in the user's mobile device and in the Raspberry PI. At first, the mobile device application initializes the Bluetooth and sends a connection request to the application running on the Raspberry PI. After connecting, the mobile device application sends, via Bluetooth, the XML file containing the description of the user's profile. The ubiquitous application reads the XML file and obtains information about the user's needs and preferences.

From the data obtained from the XML file, the ubiquitous application queries that there are adaptations to be performed according to a set of rules defined in the application. As an example, let's consider that the user's profile indicates the user has low vision. In the ubiquitous application there is an adaptation rule which states that when the user has low vision, the application should provide a sound output. If another user indicates in his profile that he prefers textual output and he is elderly, the application must adapt the textual output to the user preference, for instance, increasing the font size of the text. To cope with the concurrency between devices of different users, the ubiquitous application must implement a queue. The requests of each user are met as the progress of the queue. However, if there is no conflict between the requests, the server can attend more than one user at the same time (e.g.: a user needs an audio output and another user needs a visual output from the same information).

5 Proof of the Concept

In order to demonstrate the possibility to provide adaptive interfaces using the proposed architecture, we have been developing a prototype. Considering the emerging context of smart cities, the prototype is set in a bus stop. In this scenario, for instance, a visually impaired person may find it difficult to take the correct bus. The same can happen to an elderly person. The prototype uses a monitor that displays the bus schedules as well as the next bus identification. The system shows the most suitable user interface considering the user's profile, for example, adjusting the font size considering the interaction profile of an elderly person. By identifying the elderly person needs through its interaction profile (XML), the system makes adjustments according to your needs and preferences. As the elderly person indicated in his/her profile that he/she has low vision, the system infers that the font size should be increased.

To test the adaptability of the prototype, we used three different users' profiles. The first profile represents a user with visual impairment, the second an elderly user and the third represents a teenager. The details for creating the users' profiles are based on the personas described by [3].

The visually impaired user is represented by the persona Patricia, the elderly is represented by the persona Francisca and the teenager is represented by the persona Danilo. Their interaction preferences are shown in Table 2. Because not all the information needed for the instantiation of user's interaction profile was available in the description of the personas, some other information was inferred. Based on the information contained in Table 2, the three XML files have been created. Figure 4 shows how the XML file would be instantiated with Francisca's profile.

Table 2. Interaction preferences of the three user's profiles

Profile \ User	Patricia persona	Francisca persona	Danilo persona
Input	Keyboard	Voice	Keyboard
Output	Sound	Text	Text
Physical and/or Cognitive characteristics	Visual impairment	Low vision, attention deficit, problems memorizing	-
Interests	Books, Songs, Social Networks	Novels	Games
Literacy level	Higher	Basic	Secondary
Age	20	80	13

A preview of the proposed scenario, considering the three defined profiles and the adjustments to be made are available in Figure 5. The prototype has been implemented in Java language on the Raspberry PI and Java on Android platform in the mobile device. The ubiquitous application runs in a Raspberry PI model B in which an adapter to enable communication via Bluetooth was plugged (see Figure 5). The mobile device application is running on a smartphone with Android operational system in version 2.3.5. The ubiquitous application output is made on a TV via HDMI.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<user-profile>
  <statement
    auxiliary = "has"
    predicate = "AbilityToSee"
    range = "low-medium-high"
    object = "low"
    group = "Abilities"
  />
  <statement
    auxiliary = "has"
    predicate = "Age"
    range = "years"
    object = "80"
    group = "Demographics"
  />
  <statement
    auxiliary = "has"
    predicate = "EducationLevel"
    range = "basic-primary-secondary-higher"
    object = "basic"
    group = "Demographics"
  />
  <statement
    auxiliary = "hasPreference"
    predicate = "Voice"
    range = "primary-secondary"
    object = "primary"
    group = "InterfacePreferences"
    subgroup = "Input"
  />
  <statement
    auxiliary = "hasPreference"
    predicate = "Visual"
    range = "primary-secondary"
    object = "primary"
    group = "InterfacePreferences"
    subgroup = "Output"
  />
</user-profile>
```

Fig. 4. Francisca's profile mapped into XML

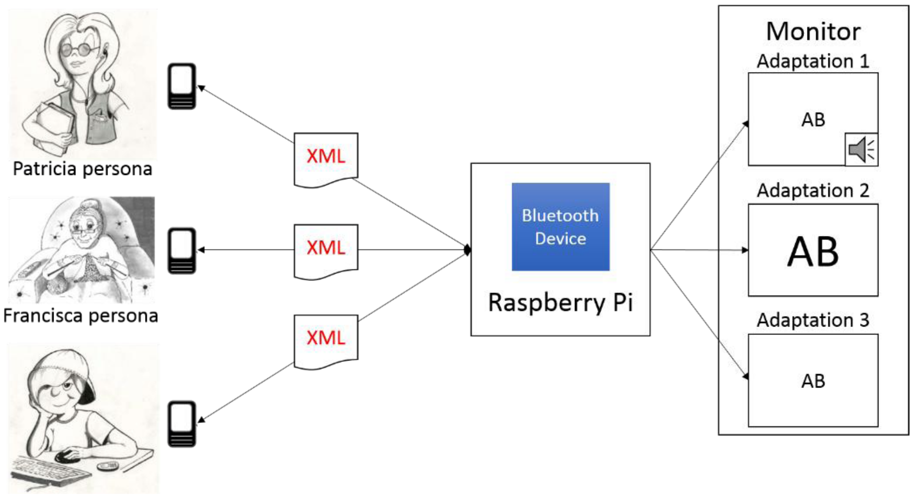

Fig. 5. Proposed scenario

The ubiquitous application has the following modules:

1. Bluetooth Connection Manager;
2. Command Receiver: responsible for receiving the XML file with the user's interaction profile, decoding and making it available to the Command Processor;
3. Command Processor: responsible for performing the adaptation according to the user's profile and the adaptation rules defined in the application.

The mobile device application provides a screen for the user to start or stop the service. When initializing the service, a connection is established between the user's mobile device and the Raspberry PI. The various user requests are placed in a queue and adjustments are performed one by one. In Figure 4 we can see how it is done, for example, adjusting the font size for the interaction profile of Francisca's persona. By identifying the persona Francisca through her interaction profile (XML), the system makes adjustments according to her needs and preferences. In Figure 6, we observe

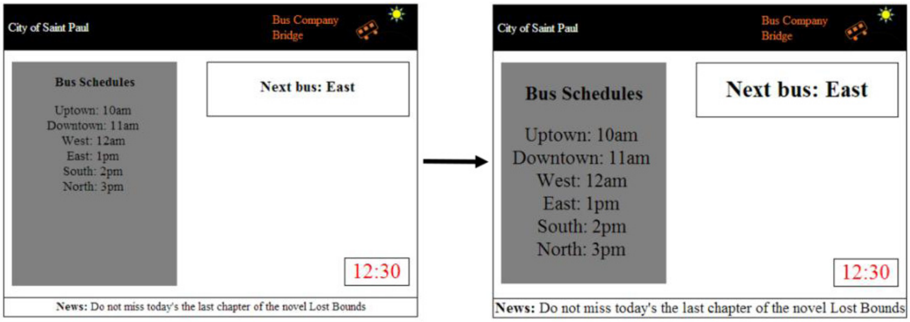

Fig. 6. Adapting the application interface to the persona Francisca

that the font size of the text is increased, since the persona Francisca has indicated that she prefers output as text. However, as the persona has low vision, the system inferred that the font size should be increased.

6 Conclusions and Future Work

This paper presented an architecture for adapting user interfaces of ubiquitous environments considering the user's interaction profile written in an XML specification. As the main contributions of this work, it is possible to highlight:

1. The proposed architecture shows that it is possible to provide adapted user interfaces in a simple, low-cost and extensible approach using a Raspberry PI;
2. Unlike the architectures found in the literature, this allows that the user profile is defined by the user himself, allowing him/er to set his/er needs, preferences and interests;
3. The profile goes with the user (mobility of the profile). As the profile is on the user's mobile device, when the user moves, he/s carries the defined profile. Thus, other applications can easily read it;
4. Furthermore, one of the advantages of this architecture is that the user profile data is stored on the user's mobile device for privacy;
5. This paper contributes to overcoming the challenge indicated by [9]: acquire knowledge of user needs, and provide appropriate solutions for different combinations of users' characteristics and functional limitations.

The prototype developed shows the feasibility of processing different interaction profiles and providing outputs adapted to diverse users. Initially, the treatment of different profiles is done by the simplistic use of a queue. In future work, this issue will be addressed to provide a better solution, for instance considering the user mobile as an output device to deal with the common output device problem. Moreover, future works consider the extension of GUMO ontology to contemplate, for example, information about the user cognitive capabilities. The mobile application for collecting the user profile is in progress.

References

1. Abascal, J., et al.: Automatically Generating Tailored Accessible User Interfaces for Ubiquitous Services. In: Proceedings of ASSETS (2011)
2. Abascal, J., et al.: A modular approach to user interface adaptation for people with disabilities in ubiquitous environments. Internal Technical Report N. EHU-KAT-IK-01-11 (2011)
3. Alencar, T.S., Neris, V.P.A.: Ubiquitous Environments and Brazilian Personas: Can our citizens universally access this technology? In: SEMISH, Curitiba, Brazil (2012)
4. Alencar, T.S., Neris, V.P.A.: Sistemas Ubíquos para Todos: conhecendo e mapeando os diferentes perfis de interação. In: IHC, Manaus, Brazil (2013)

5. Dolog, P., Nejdl, W.: Challenges and benefits of the semantic web for user modelling. In: AH 2003 Workshop at WWW 2003 (2003)
6. Heckmann, D.: Introducing situational statements as an integrating data structure for user modeling, context-awareness and resource-adaptive computing. In: ABIS (2003)
7. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendorff, M.: Gumo – The General User Model Ontology. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 428–432. Springer, Heidelberg (2005)
8. Heckmann, D., Krueger, A.: A User Modeling Markup Language (UserML) for Ubiquitous Computing. In: Brusilovsky, P., Corbett, A.T., de Rosis, F. (eds.) UM 2003. LNCS, vol. 2702, pp. 393–397. Springer, Heidelberg (2003)
9. Margetis, G., Antona, M., Ntoa, S., Stephanidis, C.: Towards Accessibility in Ambient Intelligence Environments. In: Paternò, F., de Ruyter, B., Markopoulos, P., Santoro, C., van Loenen, E., Luyten, K. (eds.) AmI 2012. LNCS, vol. 7683, pp. 328–337. Springer, Heidelberg (2012)
10. Martinez-Villaseñor, L.M., Gonzalez-Mendoza, M., Hernandez-Gress, N.: Towards a Ubiquitous User Model for Profile Sharing and Reuse. In: Sensors (2012)
11. Martini, R.G., Librelotto, G.R.: Uma abordagem para a personalização automática de interfaces de usuário para dispositivos móveis em Ambientes Pervasivos. In: SEMISH, Curitiba, Brazil (2012)
12. Miñón, R., Abascal, J., Aizpurua, A., Cearreta, I., Gamecho, B., Garay, N.: Model-Based Accessible User Interface Generation in Ubiquitous Environments. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (eds.) INTERACT 2011, Part IV. LNCS, vol. 6949, pp. 572–575. Springer, Heidelberg (2011)
13. Nakajima, T., et al.: Middleware design issues for ubiquitous computing. In: MUM 2004, pp. 55–62. ACM, New York (2004)
14. Newman, M., et al.: Designing for Serendipity: Supporting End-User Configuration of Ubiquitous Computing Environments. In: Proceedings of ACM DIS 2002 (2002)
15. Saha, D., Mukherjee, A.: Pervasive computing: a paradigm for the 21st century, pp. 25–31. IEEE Computer Society, New York (2003)
16. Schilit, B.N., Theimer, M.M.: Disseminating active map information to mobile hosts. IEEE Network, 22–32 (1994)
17. Schmidt, A.: Context-Aware Computing: Context-Awareness, Context-Aware User Interfaces, and Implicit Interaction. The Encyclopedia of Human-Computer Interaction (2013)
18. Sousa, J.P., Garlan, D.: Aura: an Architectural Framework for User Mobility in Ubiquitous Computing Environments. In: Proceeding of the 3rd Working IEEE/IFIP Conference on Software Architecture, Montreal (2002)
19. Tandler, P.: Software Infrastructure for Ubiquitous Computing Environments: Supporting Synchronous Collaboration with Heterogeneous Devices. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) UbiComp 2001. LNCS, vol. 2201, p. 96. Springer, Heidelberg (2001)
20. Vanderheiden, G.: Anywhere, Anytime (+Anyone) Access to the Next-generation WWW. In: Computer Networks and ISDN Systems, pp. 1439–1446 (1997)
21. Yamin, A.C.: Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva. Thesis (Ph.D. In: Computer Science) - Institute of Informatics, UFRGS, Brazil (2004)
22. Weiser, M.: The Computer for the 21st Century, pp. 94–104. Scientific American (1991)