International Conference on Control, Engineering & Information Technology (CEIT'13)
Proceedings Engineering & Technology - Vol.2, pp. 130-135, 2013

Context Awareness Systems: Architecture and Context Modeling

Lamia Zouhaier, Yousra Hlaoui Ben Daly, Leila Jemni Ben Ayed
Laboratory LaTICE – ESSTT
University of Tunis- Tunisia
Zouhaier.lamia@yahoo.fr yousra.bendalyhlaoui@esstt.rnu.tn Leila.jemni@fsegt.rnu.tn

***Abstract*— the big needs of disabled and normal users who want to receive the needed information using computational system at anytime and everywhere is now a necessity. The need for services architectures that is aware of the context that to build content adaptation applications that maximize the user satisfaction. The development of context-aware applications should be supported by adequate context information modeling and reasoning techniques. In this paper, we present how to design a conceptually layered framework that supports context aware application to explain the different elements common to most context-aware architectures, but also how a context can be modeled and shared. In this paper we focus on existing work in this research issue in order to determine the different elements common to most context-aware architectures and determine different approaches of modeling context in ubiquitous computing.**

***Keywords*— Adaptation, User Interface, User with special needs, Context-awareness, Context Modeling, Context-awareness architecture.**

I. Introduction

Today, the huge emergence of technological developments have encouraged integrating mobile devices into computational applications in order to facilitate daily users activities. Pervasive or ubiquitous computing system makes the information available everywhere and at any time, it must be able to perceive the surrounding environment [23]. But, sensory, cognitive or physical disable people are usually excluded from computing services since they are unable to use interactive systems that have been designed ignoring human diversity. People with disabilities confront access barriers on their daily activities. They are almost unable to interact with user interfaces IUs of the emerged devices. There is a lack of enforcement for these people on how to proceed with such system as well as how to use easily and satisfactory. It is indispensable that computational system gives accessible interface for disable people in order to provide equal access and equal opportunity between all people.

Context-Awareness is important for pervasive computing environments to adapt computational entities to changing situations such as the user's needs and technical capabilities [10]. Existing work in the area of context awareness focused on all aspects of capture, interpretation, modeling, storage and dissemination of context but there are no generic and global solutions that include all steps of adaptation from context acquisition to generated final interface.

Interface should adapt to every user profile, contextual information is considered relevant if it influences the way the user performs the task in a natural environment. Our first target in adaptation is the user according to user preferences or physical characteristics as auditory or visual impairments.

The majority of existing contributions in the field of context-awareness are interested on context management without providing a support of adaptation to these new contexts of use. That is why we try in this paper to survey some existing infrastructures that support context-awareness. But, we give a little overview of some related works in adaptation of computing systems.

The remainder of this paper is as follows: in section two, we present some relevant work in relation with our problematic. The section three is about the understanding of context and context-awareness concepts and design principals. We give some of the more relevant and emerged architectures and we discuss some important characteristics of basic design principles. In section four, we describe some modeling concept of context and we opt to classify the existing context models. Finally, we discuss and conclude the essentials points to be considered for adaptation of user interface and how can context awareness systems reinforce, as a starting point, such strategy.

II. Related work

In this section, we discuss some works that deals with adaptation of user interface according to disabled users.

Adaptive and adaptable interactions techniques are increasingly emerged in recent research. There are no generic solutions oriented towards accessibility of user interface, but different terminologies are employed as Universal Access [6], User Interfaces for All [8], Design For All [21] Unified User Interfaces [8] because of the range of the population which may gradually be confronted with accessibility problems extends beyond the population of disabled and elderly users to include all people [7]. Universal Access refers to the global requirement of coping with diversity in: (i) the characteristics of the target user population (including people with disabilities); (ii) the scope and nature of tasks; and (iii) the different contexts of use and the effects of their proliferation into business and social endeavors [7].

Universal accessibility system should be accessible for all users, although the design is focused on people with special needs. But none of these projects resulted in any concrete solutions for users with special needs. The scope of User Interfaces for All, as a perspective on HCI, is necessarily broad and complex, involving challenges, which pertain to issues such as context oriented design, diverse user requirements and adaptable and adaptive interactive behaviors. This diversity of needs is generally ignored at the present time. Occasionally, it is addressed in one of several ways: manual redesign of the interface, limited customization support, or by supplying an external assistive technology.

AVANTI [6] is the first project to employ adaptive techniques in order to ensure accessibility and high quality of interaction for all potential users [7]. It put forward a conceptual framework for the construction of systems that support adaptability and adaptivity at both the content and the user interface levels [7]. The distinctive characteristic of the AVANTI browser is its ability to dynamically tailor itself to the abilities, skills, requirements, and preferences of the end-users, to the different contexts of use, and to the changing characteristics of users as they interact with the system.

EGOKI [1, 20] is a system that generates accessible mobile user interfaces adapted for people with disabilities in order to grant them access to ubiquitous services. These interfaces are intended to provide access to ubiquitous services in intelligent environments. EGOKI dynamically creates an instance of the interface running on the user device. To adapt the interface to the user characteristics, it is necessary to take into account what the most suitable communication modalities are for each user, mapping them to the appropriate media.

III. CONTEXT-AWARENESS INFRASTRUCTURES

The most commonly used definition of context is that of Dey [2, 3]. All work in this emerged domain research of context-awareness system are based on this definition. This definition resumes the other previous definitions of context given by Schilit and Theimer [4], Brown et al. [14], Ryan and Pascoe [25]. A context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves.

Contextual information is any type of information that can characterize the state of an entity regarding a specific aspect or situation. In an interactive system, an entity is an object which can be a person (user), a place (location) or an object of interaction (device).

In [13], Context-awareness system must be able be aware of the characteristics and constraints of the user's preferences and environment characteristics, i.e. contextual information, and accordingly perform a number of actions and operations automatically to adapt to changes of the sensed environment without explicit user intervention. Four core features that constitute a context awareness application [29]:

a. Contextual sensing is the ability to sense context information and present it to the user.
b. Contextual adaptation is the ability to execute or modify a service automatically based on the context.
c. Contextual resource discovery allows context-aware applications to locate and exploit resources and services that are relevant to the user's context.
d. Contextual augmentation is the ability to associate digital data with the user's context.

Different ways are used to implement context-aware systems. In fact, there are factors which are special requirements and conditions that traces the architecture of such systems such as the location of sensors (local or remote), the number of users, the available resources of the used devices (high-end-PCs or small mobile devices) or the facility of a further extension of the system [24]. Accordingly some applications were developed as specific application of context aware computing system.

Active Badge [27] is a project for building system for phone calls delivery according to the location of the called person. The system uses badges which continuously emit infra-red signals at a given frequency.

Cyberguide [11] is project which provided a context-aware tour guide to visitors. It help the tourist to visit places by supplying him with a set of information (i.e paths to follow, interesting sites to visit based on his location, etc.) based on his current location.

We are interested to describe different distributed architectures of some frameworks Context Toolkit, CoBrA, SOCAM and CMF to distinguish the common and the different points.

A. Context-Toolkit Architecture

Context Toolkit based on widget component is a set of toolkit composed of the four building blocks (see figure 1):

- *Context-Widget* acquires information from the environment through the use of software and/or hardware-based sensors.
- *Context-Interpreter* gives meaning to the captured context.
- *Context-Aggregator* collects related context together. It helps the framework in supporting the delivery of a specified context to an application, by collecting related context about an entity that the application is interested in.
- *Context-Service* is responsible for controlling or changing state information in the environment using an actuator. It provides reusable context-aware behaviors or services to applications

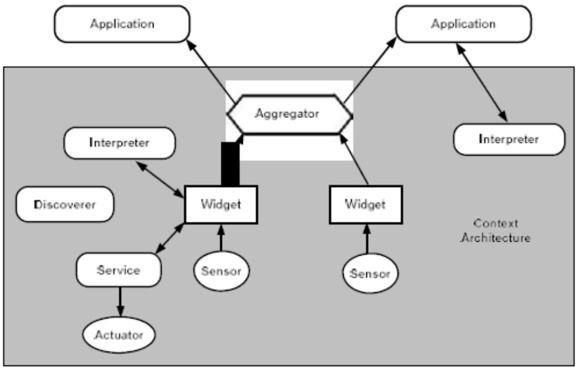

Figure 1: Architecture of Context Toolkit [2] (taking from [19])

Dey [4] has used a basic component "context-widget" which is responsible for retrieving state information about the environment from a sensor. In [29], Dey cited some benefits of the use of widget: it provides a separation of concerns by hiding the complexity of the actual sensors used from the application; it abstracts context information to suit to the expected needs of applications. It provides, also, an easy access to context information through querying and notification mechanisms and reusable and customizable building blocks of context sensing. The architecture does not permits context reasoning because the context modeling is based on representation key/value representation which impede reasoning.

The Context-Toolkit Architecture is simple for implementation. It offers a reusable widgets and a distributed communication between devices. One of the defects of this architecture is at the discovery mechanism. It centralisation does not make it a perfect peer-to-peer communication model. And also, it suffers at the extensibility which is limited when we increase the number of devices.

B. Context Broker Architecture (CoBrA)

It's a framework that its architecture is agent oriented of context-sensitive systems in smart environments [15].

The context broker is an intelligent agent that represents the central element of this architecture. Its role is to maintain a shared model of context for a community of agents, services, devices and sensors.

The broker agent is composed of four layers of components: context knowledge, context reasoner engine, context acquisition module and privacy management module (see figure 2). In fact, the broker agent has the role of collecting context from different parts: devices, sensors of its surrounding environment and from others agents [15]. And after, it merges them in a coherent model. This model will be shared among devices and their corresponding agents. CoBrA uses OWL to define ontologies of context. It provides a set of CoBrA-ONT which is ontologies for supporting context reasoning and knowledge sharing. It allows, also, the users to control the sharing and their situational information throw privacy policy. Contrary to other systems that the computing entities are usually free to share any acquired situational information of any user.

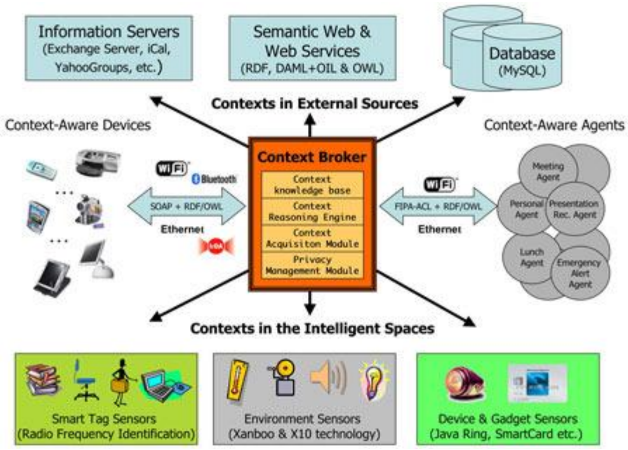

Figure 2: Context-Broker Architecture (extract from [18])

C. Context Management Framework CMF

The framework [26] permits semantic reasoning on context in real time with the presence of noise and incertitude and rapid variation of context. The contextual information is delivered to applications by using a communication model based on event-based mechanism. The framework uses a client/server architecture composed of the following basic components:

- *Context manage component* which is responsible for the storage of contextual information on server and the delivery of context to clients using different techniques.
- *Resource server* which is responsible for the acquisition of contextual information from physical sensors and their interpretation according to a specific format before sending them to the context manager.
- *Context recognition service* responsible for the conversion of the data stream to a presentation defined in the context ontology.
- *Change detection service* responsible for the detection of service change and therefore the context change
- *Security* responsible for the verification and control of contextual information.

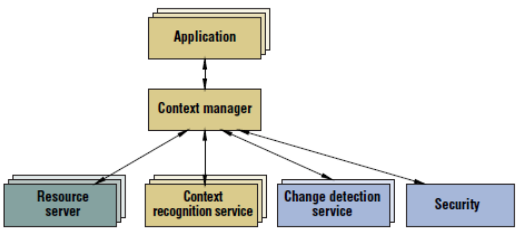

Figure 3: Context Management Framework CMF Architecture (extract from [26])

D. Service-Oriented Context Aware Middleware SOCAM [14]

SOCAM is a framework of a service oriented context-aware middleware to build context-aware mobile services in an intelligent car. The SOCAM architecture is composed of the following components (see figure 3):

- Context providers which abstract useful contexts from heterogeneous sources External or Internal; and convert them to OWL representations so that contexts can be shared and reused by other service components.
- Context interpreter which provides logic reasoning services to process context information.
- Context database responsible of storing context ontologies and past contexts for a sub-domain. There is one logic context database in each domain, i.e. home domain.
- Context-aware services make use of different level of contexts and adapt the way they behave according to the current context.
- Service locating service provides a mechanism where context providers and the context interpreter can advertise their presence. It enables users or applications to locate these services.

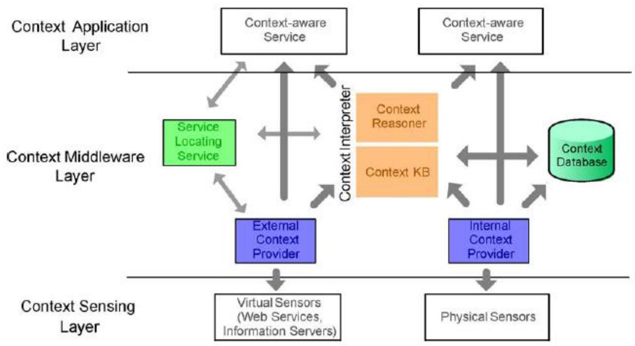

Figure 4: The SOCAM architecture (extract from [14]).

The main strength of the SOCAM architecture is its context reasonner which uses ontology for context description and allows a robust reasoning on context.

E. Discussion

Investigating context-aware computing is a complex task due to the diversity of context that can be sensed and how can be sensed and the variability of ways of context interpretation. There is no common architecture providing all context sensing and interpreting components. We note that all infrastructures described above, has built their own basic context-component (widget, agent, sentient component or service component).

Different characteristics that must be checked in such context-aware system: sensitivity which is the capability of perception and capture of the state of their surrounding environment by using physical sensors; autonomy which is their capability to operate in distributed manner, independently of human control.

The majority of these frameworks of context sensitive use a layered architecture supporting the important aspects of sensor capture, context extraction and reasoning. These layers are used in order to adapt system to the current context. They are based specially on:

- *Layer of capture* based on physical sensors. This layer acquires information from surrounding environment.
- *Layer for interpretation* in order to interpret and make useful the context captured.
- *Layer for reasoning* based on ontology responsible of to reasoning about contextual information.
- *Layer of service* which embodies a set of services to manage contextual information.

A survey made by [12] show that there are two approaches depending on whether contextual information are centralized or distributed. Architectures based on a server suffer from implementation cost. In fact with a centralized system, the others components of the system will be affected if a problem occur within the centralized system (i.e. a breakdown of server). A centralized architecture doesn't represent the best choice in such computing system because of the two characteristics of context-awareness system: mobility and distribution.

CoBrA differs from other similar architectures by using the Web Ontology Language OWL. In fact, ontology is defined to support knowledge sharing and data fusion. For that reason, CoBrA uses logic inferences for detecting and resolving inconsistent context knowledge acquired from unreliable physical sensors and Rei policy language to give users the control of their contextual information [15]. In Context-Toolkit and other systems, context is often implemented as programming language objects (e.g., Java classes). That can be considered a lack of expressiveness to support context reasoning and high-level knowledge sharing [16, 18]. Context-Toolkit enables the integration of context data into applications, but did not provide mechanisms for performing sensor fusion, reasoning about context, or dealing with mobility [12]. CoBrA architecture uses a context reasonner which is strong point in its architecture. This context-reasonner uses ontology for context description and allows a robust reasoning on context. The CMF uses ontology for context representation but does not offer a context reasoning module.

IV. Context Modeling

The fundament for context-awareness is giving a formal context model which is needed to represent the context to be able to interpret it [10]. To manage the context information, different methods are used in order to define a common structure for representing contextual information. In the previous section, we note that most of context- aware systems are based on ontology for reasoning process and knowledge sharing. It is considered as the best description of context. We present in this section some of existing context modeling approaches. Context Modeling Approaches

We survey the most relevant approaches that address context modeling. These approaches differ on the scheme of data structure used to exchange contextual information in the respective system. Our Survey is based on the survey of [28].

a) Key-value and Markup Model

This model is the simplest way and one of earliest approach defined by Schilit et al. [4] to model contextual information. It is based on a pairs Key/Value in order to provide the value of context information to the application as an environment. It is frequently used in distributed service frameworks. In such frameworks, the services itself are usually described with a list of simple attributes in a key-value manner, and the employed service discovery procedure operates an exact matching algorithm on these attributes [28].

b) Markup Scheme Models

The Markup-based context information model is hierarchical data structures which using a variety of markup languages based on markup tags with attributes and content. Profiles are typical representatives of Markup based context information model. These kinds of context modeling approaches usually extend and complete the basic Composite Capabilities/Preference Profile (CC/PP) and *User Agent Profile* (UAProf) vocabulary and procedures to try to cover the higher dynamics and complexity of contextual information compared to static profiles. *Comprehensive Structured Context Profiles (CSCP)* is an example of this approach.

c) Object Oriented Models

The details of context processing is encapsulated on an object level and hence hidden to other components. Access to contextual information is provided through specified interfaces only. Representatives for this kind of approach are the cues which cues provide an abstraction from physical and logical sensors. A cue is regarded as a function taking the value of a single physical or logical sensor up to a certain time as input and providing a symbolic or sub-symbolic output. A finite or infinite set of possible values is defined for each cue. The output of each cue depends on a single sensor.

d) Graphical Models

Unified Modeling Language (UML) is one of the graphical models appropriate to model the context due to its generic structure. It is based on diagrams considered as strong graphical component. Other model are an extension to the Object-Role Modeling (ORM) which is based on fact the basic modeling concept, and the modeling of a domain using ORM involves identifying appropriate fact types and the roles that entity types play.

e) Ontology Based Model

Ontology based models of context information is one of semantic description that can be used to model context. Various OWL ontologies have been proposed for representing shared descriptions of context. Among the most prominent proposals are the SOUPA [17] ontology for modeling context in pervasive environments, and the CONON [9] ontology for smart home environments. OWL-DL ontological models of context have been adopted in several architectures for context-awareness. The Context Broker Architecture (CoBrA) [18] and the SOCAM [14] middleware has adopt the SOUPA and CONON ontologies, respectively.

F. Context Representation

The different entities must have a common structure for representing information. Each context expression must contain at least *Context type* and *Context value*. An example is given in table 1:

- *Context type:* Each context must belong to a category (Sound, time, temperature, etc.). These types will be used in a context subscription or query. Context type concepts form a tree structure.
- *Context value* refers to the semantic or absolute "value" of context type and is usually used together with *context type*, forming a verbal description. In some cases, *context value* might contain an absolute numerical value or feature describing context.
- *Attributes* specify the context expression and might contain any additional details not included in the other properties e.g Timestamp describing the date/time when context was sensed, Source containing how information was gathered, etc.

Table 1: Example of Context vocabulary (Source: [26])

Context type	**Context value**
Environment:Sound:Intensity	{Silent, Moderate, Loud}
Environment:Light:Intensity	{Dark, Normal, Bright}
Environment:Light:Type	{Artificial, Natural}
Environment:Location:Building	{Indoors, Outdoors}

V. CONCLUSION

This work will allow us to frame the issues for user interface using context-awareness and open research perspectives related to the adaptation of interfaces.

We have tried to survey some of relevant context-aware architectures that were proposed in others works to support context awareness applications but also context modelling. There is no common architecture providing all context sensing and interpreting components. We note that all infrastructures described above, has built their own basic context-component (widget, agent, sentient component or service component).

A context modeling must support the ability to model different types of context, such us physical, social or computational context and unify the context representation. But also, it must provide a uniform way of representing and sharing context. Thus, a good context modeling approach must include modeling of context information quality to support reasoning about context and providing a good formalism to reduce the complexity of context-aware applications and to improve their maintainability and evolvability.

REFERENCES

[1]. A. Julio, A. Amaia, C. Idoia, G. Borja,G. Néstor, M. Raúl, "A modular approach to user interface adaptation for people with disabilities in ubiquitous environments", Internal Technical Report No. EHU-KAT-IK-01-11, 2007.

[2]. A.K. Dey, D.G. Abowd, and D. Salber, "A Conceptual Framework and A Toolkit for Supporting The Rapid Prototyping of Context-Aware Applications", Human-Computer Interaction, 2001.

[3]. A.K.Dey, “Providing Architectural Support for Building Context-Aware Applications “, Phd Thesis, Georgia Institute of Technology, 2000.
[4]. B. Schilit, N. Adams, R. Want et al., “Context-aware Computing Applications”, Xerox Corp., Palo Alto Research Center, 1994.
[5]. C. Bettini , J. Indulska , D. Nicklas , A. Ranganathan , D. Riboni , et al, “A Survey of Context Modelling and Reasoning Techniques”, In Journal Pervasive and Mobile Computing (2): pp 161-180, 2010.
[6]. C. Stephanidis, A. Paramythis, M. Sfyrakis, A. Stergiou, N. Maou, A. Leventis, G. Paparoulis, C. Karagiannidis, “Adaptable And Adaptive User Interfaces For Disabled Users In The Avanti Project”, 5th International Conference On Intelligence In Services And Networks: Technology For Ubiquitous Telecom Services, pp 153-166, 1998.
[7]. C. Stephanidis, A. Savidis, “UniversalAccess in the Information Society: Methods,Tools, and InteractionTechnologies”, UAIS, 1: 40–55, 2001.
[8]. D. Akoumianakis, A. Savidis and C. Stephanidis, “Encapsulating Intelligent Interactive Behaviour in Unified User Interface Artefacts”, Interacting With Computers, Vol. 12, No. 4, pp. 383-408, 2000
[9]. D. Zhang, T. Gu, X.Wang, “Enabling Context-aware Smart Home with Semantic Technology”, International Journal of Human-friendly Welfare Robotic Systems 6 (4), pp. 12–20, 2005.
[10]. F. Ay, “Context Modeling and Reasoning using Ontologies”, University of Technology Berlin, 2007.
[11]. G. Abowd, C. Atkeson, .J Hong, S. Long, R. Kooper, and M. Pinkerton, “Cyberguide: A mobile context-aware tour guide”, CHI'96, 1997.
[12]. G. Biegel, V. Cahill, “A framework for developing mobile, context-aware applications”, Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communication, pp.361 365, 2004.
[13]. G. Rey, J.Coutaz, “Le contexteur: Capture et distribution dynamique d’information contextuelles”, Mobilité et Ubiquité (Ubimob04), ACM. Publication, 2004, pp. 131-138, Nice, France.
[14]. G.Tao, K.P.Hung, Q.Z. Da, “A Service-Oriented Middleware for Building Context-Aware Services”, Journal of Network and Computer Applications, Vol. 28, No. 1, January 2005.
[15]. H Chen, T.Finin, A.Joshi, “A Context Broker for Building Smart Meeting Rooms”, In International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004.
[16]. H Chen, T.Finin, A.Joshi, “Semantic Web in a Pervasive ContextAware Architecture”, In Artificial Intelligence In Mobile System, 2003.
[17]. H. Chen, F. Perich, T. W. Finin, A. Joshi, “SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications”, in: 1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004), IEEE Computer Society, 2004.
[18]. Harry Chen, “An Intelligent Broker Architecture for Context-Aware Systems”, Phd Thesis, University of Maryland Baltimore County, January 2003.
[19]. Harry Chenn, http://cobra.umbc.edu/about.html, 2004.
[20]. J. Abascal, I. Fernández de Castro., A. Lafuente, J.M. Cia, "Adaptive Interfaces for Supportive Ambient Intelligence Environments", In Proceedings of the Int. Conf. on Computers Helping People with Special Needs 2008. LNCS Vol. 5105. Springer, pp. 30-37, 2008.
[21]. J.V. Lopez, F. Montero, C. A. Fermandez, and M. Lozano, “Towards Adaptive User Interface Generation: One Step Closer to People”, In Proceedings of ICEIS (3), 2003, pp.97-103.
[22]. K.R. Sang, K. Lee and H. Kim, “Ontology-based Context and Preference Model for Personal Service Robot”, 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 2012.
[23]. M. Miraoui, C. Tadj, C. Ben Amar, “Architectural Survey of Context-Aware Systems in Pervasive Computing Environment”, Ubiquitous Computing and Communication Journal (UBICC), Vol. 3, No. 3, 2008.
[24]. Matthias Baldauf, “A Survey on Context-Aware Systems”, International Journal Ad Hoc and Ubiquitous Computing, Vol. 2, No. 4, 2007.
[25]. N. Ryan, J. Pascoe and D. Morse, “Enhanced reality fieldwork: the context-aware archaeological assistant”. Computer Applications and Quantitative Methods in Archaeology. V. Gaffney, M. van Leusen and S. Exxon, Editors. Oxford, 1998.
[26]. P. Korpipää, J. Mantyjarvi, J. Kela, H. Keranen, E-J. Malm, “Managing context information in mobile devices”, IEEE Pervasive Computing, Vol. 2, No. 3, July–September, pp.42–51, 2003.
[27]. R. Want, A. Hopper, V. Falcao, and J. Gibbons, “The Active Badge Location System”, ACM Transcation on Information Systems 10 (1), pp. 42-47, 1992.
[28]. T. Strang and C.L. Popien, “A Context Modeling Survey”, Proc. Int'l Workshop Advanced Context Modeling, Reasoning and Management, 2004.
[29]. Vitor Hugo, “Ontologies for Context-Aware Applications”, Porto, March 2008.