# A Global Approach of Accessible Context-sensitive User Interface

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:*** *User's tasks, habits and preferences require ubiquitous and mobile context aware interactive system. Disable people find themselves excluded from services that can be offered by pervasive system. It is essential to take into account all categories of user with disability in the design and development of computer systems in order to provide opportunity to make their daily activities easily and simple. The problem is so complex when the context of use changes dynamically over time. The computing system becomes more confronted to a need to be adapted to multiple configurations of the context of use. The present paper describes a position paper relative to recently existing works that deals with context awareness applications as a way of adaptation process in order to provide a global vision and to deduce the architecture of context aware system supporting adaptation.*

**Keywords:** Context-awareness, Architecture, Adaptation Rules, Disable Users, Ontology

**Received:** 20 November 2012, Revised 18 December 2012, Accepted 24 December 2012

## 1. Introduction

The advance of new technology reflects an increment of users' needs to use information in every place and at any time. Users have to deal with diverse devices (mobile or fixed) supporting diverse interfaces and used in diverse environments. New interaction capabilities are progressively enhanced with the emergence of new plateform (PC, PDA, Smartphone, phones, interactive kiosks, etc.).

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. In this way, many research has emerged recently which focus on how to improve accessibility in web using personalisation according to guidelines. Others have focus on how to generate interface using adaptation mechanisms for generation of user interface based on adaptation rules.

Adaptation process must consider multiple configurations of the context that includes user profiles, hardware and software capabilities of devices and as well the physical environment (light, noise, location, etc.) [3].

Adaptation of interface is the capacity of personalization that takes into account all contextual information from different entities as environment attributes and user characteristics. New approach of adaptive user interface development is emerged in order to adapt the user interface to the current context of use. Some problems have occurred and motivated new recently researches to define features for interface's adaptation according to the context change. The complexity is how the application must be able to both detect the current state of context and the new context in an ambient environment and to determine what actions to take based on the contextual information collected. Using the contextual information captured, it is necessary to know the user current situation as in smart environments.

Building context-aware applications is one of the solutions adopted but also still a complicated task due to the lack of an adequate and generic infrastructure that support in pervasive computing environments [5]. Some issues have to be clarified and solved: how to give the device the ability to recognize the situation, and how to interpret the surrounding environment.

The goal of this paper is to survey the most relevant literature in the area of context-sensitive systems in order to capture the different challenges and solutions to be considered for user interface adaptation in order to ensure accessibility to disable people.

The reminder of the paper is structured as follows. Section 2 summarizes the state of the art on related works of adaptation process. In section 3, we look at various definitions of context and context-awareness and we presents some existing context-aware systems in order to determine the architecture of context-awareness. In section 4, we discuss the infrastructure that supports context-aware systems. Finally, we concludes and shows some future lines of work.

## 2. Related Work

Within model-based UI research [24], there has been some interest in the generation of user interfaces using transformation rules. These model based environment aims at providing a systematic approach to specify the UI by means of models. These models will be translated into the final code executed by the user. Nevertheless, a set of rules is required to generate the final code for the UI adapted to every target supported context of use [24]. The design of computing system must be generic in order to give the possibility to personalization for other specific domain and for specific user's interest (e.g. handicap). Our main goal is interested to solve the problem of how to create accessible user interfaces that allow access to services provided by local devices (e.g. ATMs, vending machines, information kiosks, smart home, smart vehicle, etc.). One of the key issues in achieving full accessibility is the instantaneous generation of an adapted accessible interface suited to specific user that requests the service.

There are many approaches for adaptive UIs, taking a model-based approach more or less into account [24]. Some approach handle adaptation at design time. This is the most common way of handling adaptation in model-based UI development environments. The UIs are adapted to different platforms as part of a UI generation process [24] [11]. A second approach is to provide some kind of transformation mechanism at run time [28], i.e., a mechanism that transforms an UI designed for one platform to fit to a different one.

Adaptation in user interface software is typically served by either adaptivity [11] and/or adaptability [8]. An adaptive user interface changes dynamically in response to its experience with users [9] [14] [10]. The system can recognize the situation (including multiple triggers set by the designer) and adopts reaction (recommendations) also set by the designer for the new situation [11]. An adaptable user interface, however, provides tools, which allow the user to tailor certain aspects of the interaction, while using an interactive system [9] [14] [10]. The system is customizable on explicit intervention of the user who can act on parameters set by the designer [23].

Multiple targets can be determined in adaptation process. In [12] [23], authors have defined the target as an entity that is intended for adaptation, which can be:

- Adaptation to user.
- Adaptation to environment.
- Adaptation to the physical characteristic of the system.

Adaptive and adaptable interactions techniques are increasingly emerged in recent research. Different terminologies are used like universal access [9], design for all [8], and interface for all [10]. 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. The AVANTI project [9] is one of the works in this ways, which is based in "*design of all* " concept. The group has developed some tools which are used in adaptation process**.** USE-IT is a knowledge-based tool for automating the design of interactions at the physical level, so as to ensure accessibility of the target user interface by different user groups, including people with disabilities. And a Unified User Interface Design Method which is a design method that facilitates the design of interfaces that exhibit automatic adaptation behavior based on user and context related diversity factors.

Dey and al. [4] propose architecture called Context Toolkit whose main target is to provide a reusable solution to facilitate implementation and deployment of context-sensitive interactive applications. They specify three steps to adapt to the context: firstly, the capture of context, then the interpretation of context in order to pass a representation of high level usable for the application, and finally, it is necessary to provide information interpreted in the application. The authors of [23] summarize different type of user interface adaptation configuration based on four levels:

- Initiation (I) decision of an actor to perform an adaptation.
- Proposal (P) represents the suggestions or recommendations.
- Decision (D) is the process of choosing among proposals reaction.
- Execution (E) consists of the implementation of the chosen reaction.

Dieterich's taxonomy is considered as a seminal reference for classifying adaptation techniques of user interface [23]. A process of adaptation can take different settings by combining the characteristics of adaptability (control from the user) and adaptivity (control from the system). With the combination of the adaptivity and adaptability using the four levels indicated by Dey et al [4], different configurations of adaptation process are possible [23] [9]. The user or the system can control the actions which are the steps of the adaptation process. Four types of adaptations are proposed:

- Selecting immediate technique to highlight some choices for user (bold, underline, text bigger or smaller).

- Automatic reconfiguration of the platform context is the principle of power to add or remove components in the environment (a program module, a display, etc.).

- Information and contextual commands: user actions can often be predicted depending on the situation.

There are expected actions that users perform in specific places such as a library, a kitchen or office.

- Triggers actions are simple reactions to events (If / Then) used to specify how context can adapt.

Other technique of adaptation, plasticity [15] is the capacity of an interactive system to withstand to context variations while preserving usability. In order to support the end-user preferences, adaptations rules can be changed according to user's order [14] [11]. It results from a *Situation →Reaction* process where the situation denotes a context change that needs a reaction, and reaction denotes the procedures that the system and/or the user executes to preserve usability.

## 3. Context-Awareness

### 3.1 Context Definition and properties

The most commonly used definition of context is that of Dey [1] [7]. 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 [6], Brown et al. [17], Ryan and Pascoe [13].

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.

Based on this definition, we can extract what contextual information is. It 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 [5], an example of context is given. A context may include person (name, role, etc.), location context (coordinate, temperature, etc.), computational entity contexts (device, network, application, etc.) and activity contexts (scheduled activities, etc.).

Rey et al. [29] has defined four properties of context:

- **There is no context without context:** we must set a goal on which defines a context. In our case, we have to adapt application based on user preference and environmental physical conditions.
- **Context is an information area that serves the interpretation:** All captured information must serve the goal fixed previously. The interpretation surely is the task of the system.
- **The context is an infinite and evolutionary space:** there is no absolute context. It is constructed during time.
- **Context is shared by several actors:** Two principal actors are the user and the system.

Based on Dey et al. [2] [1] definition and Rey [29] [26] properties, we can affirm that an information system's context is an infinite set of variables attributes describing any object of interaction (environment, user, device) that evolve.

Context has different aspects. Schmidt et al. [4] has focus on physical parameters and information provided by some mobile devices like PDA or mobile phone. Dey [8] explains that context parameters can be implicit or explicit. These parameters can be provided directly by sensors or by the user from the environment of the user and the application.

So, the context can integrate all factors that can describe the user finalities and his perceptions to his environment. These contextual factors cover various aspects such as environmental facors (light, noise…) or spatial-temporal (location, time, direction…).

### 3.2 Context-awareness Applications

There has been an emergence of infrastructure for the realization of context-aware systems.

#### 3.2.1 Definition and properties

Today, the huge technological developments in computerized communication systems has encouraged to integrate mobile devices into their applications in order to create new information systems called pervasive or ubiquitous, but these systems suffer of the lack of programming support for the design and development of context-aware applications [18]. Agoston [25] has defined a pervasive system as a system that makes the information available everywhere and at all time. Their characteristic is his ability to perceive the surrounding environment.

In [16], Context-awareness is defined as the ability of systems, devices or software, to 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. Context awareness and mobility are two core concepts in the vision of ubiquitous computing [18].

Four core features that constitute a context awareness application [7]:

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.

These infrastructures require the following supports [5]:

- A common context model that can be shared by all devices and services.
- A set of services that perform context acquisition, context discovery, context interpretation and context dissemination.

#### 3.2.2 Frameworks that support context-aware process

Several researchers have tried to categorize context-aware applications and features, including contextual sensing, contextual adaptation, contextual resource discovery and contextual augmentation [21] [7]. In table 1, we have presented several implementations that are considered as reference of context-aware platforms as CoBrA [19], Context-Toolkit [7], SOCAM [5], and CORTEX [18]. We analyze the approach addressed by each platform in order to deduct the principal layers of a context sensitive system.

| Framework | Description | Basic Component | Architecture | Communication Model |
|---|---|---|---|---|
| Context Toolkit [7] | It is a tool for the development and depolyment of context-aware applications. | Widget | The Infrastructure has a process- centered style. It is presented in the form of a composition of process. Five Software components:<br>- Context-Widgets<br>- Context-Aggregators<br>- Discoverer<br>- Context-Interpreter<br>-Services through the Actuator | Peer-to-peer (P2P) |
| Context Broker Architecture (CoBrA) [27] [20] [19] | It supports the development of context-aware applications in an intelligent space | Agent | The Principals components of a Context Broker are:<br>- CoBrA Ontology (COBRA-ONT)<br>- Context Reasoning Engine (CoRE)<br>- Module for Privacy Protection (MoPP) | Client/Server |
| Service Oriented Context-Aware Middleware (SOCAM) [5] | It is a platform for supporting the creation of context-aware services for building and rapid prototyping of context-aware mobile service in an intelligent car | service component | The Principals components are:<br>- Context Provider<br>- Context Interpreter<br>- Context Knowledge And Context Reasoner<br>- Service Locating Service,<br>- Context-Aware Mobile Service<br>- Context Database | Client/Server |
| CORTEX [18] | It supports the development of context-aware applications in mobile ad-hoc environments. | Sentient object | Sentient objects are composed of three major internal components:<br>- Sensory capture and context fusion<br>- Context hierarchy<br>- An inference engine which specifies the applications behavior to a given context and uses the execuation model event-condition-action | P2P |

Table 1. Existing Framework of context sensitive systems

### 3.2.3 Context-Sensitive Architecture Analysis and comparison

We are interested in this section to compare two architectures of some frameworks quoted before: Context Toolkit and CoBrA. This comparison is based on the fact that they are based on different based component and used different mode of communication and components. Let's begin by presenting their architecture.

#### 3.2.3.1 Context-Toolkit Architecture Context

Toolkit, based on widget component, is a set of toolkit composed of the four building blocks described below (see figure 3):

**-** The *Context-Widgets* acquire information from the environment through the use of software and/or hardware-based sensors.

**-** The *Context-Interpreter* gives meaning to the captured context.

**-** The *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.

**-** The *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.

Dey [7] has used a basic component "*context-widget*" which is responsible for retrieving state information about the environment from a sensor. In [7], 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.

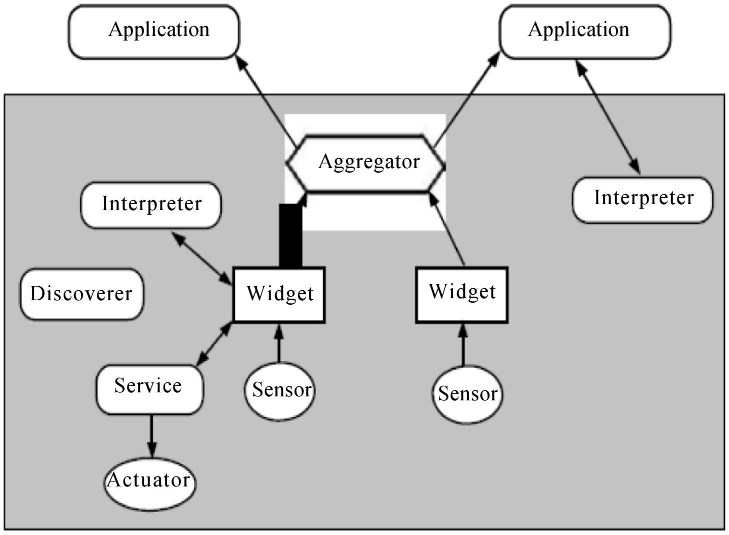

Figure 1. Architecture of Context Toolkit (Taking fom [22])

### 3.2.3.2 Context Broker Architecture (CoBrA)

Its architecture is agent oriented of context-sensitive systems in smart environments (e.g smart homes, smart vehicles) [20]. 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 (see figure 4).

The broker agent is composed of four layers of components: context knowledge, context reasonner engine, context acquisition module and privacy management module. In fact, the broker agent has the role of collecting context from different parts: devices, sensors of its surrounding environment and from others agents [20]. 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.

CoBrA allows, also, the users to control the sharing and their situational information throw privacy policy.

In other systems, the computing entities are usually free to share any acquired situational information of any user.

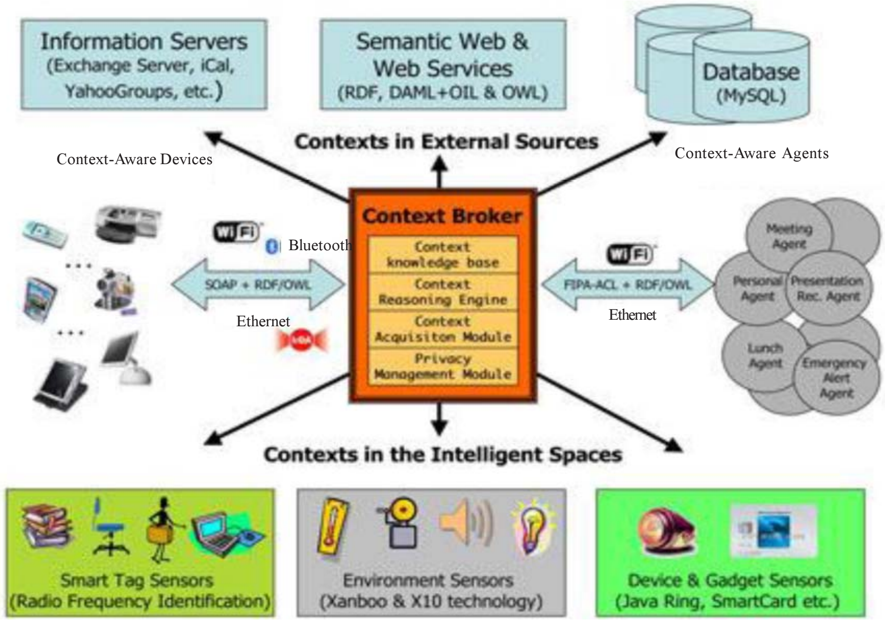

Figure 2. Context-Broker Architecture [13]

### 3.2.3.3 Discussion

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 is at the extensibility which is limited when we increase the number of devices.

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 [20]. 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 [21] [19]. 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 [13]. 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.

## 4. 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.

We note, also, that a context- aware system must to be based on ontology for reasoning process and knowledge sharing. It is considered as the best description of context. We propose the use of the Web Ontology Language OWL, an ontology-based language in order to explicitly define contexts in a semantic way.

A survey made by [13] 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.

## 5. Future Work and Contribution

After looking on infrastructure of context aware system, we conclude that its layered architecture is a good one to discover and take advantage of contextual information. This new technology has demonstrated its usefulness but till now not widely available to everyday users because they still interested on specific domain applications. Context represents the kernel of our problematic. In fact, it gathers characteristics about human, hardware, software, physical environment, sometimes quality of services delivered. We can release different challenges in this emerging research field:

- Ability to enhance the behaviour of any application by informing it of the context of its use.

- Mobility of such devices increases the complexity of determination of dynamic contextual information.

- Heterogeneity of devices needed in some distributed applications.

- The user interface facets including content, structure, navigation, presentation and style have to be modified based on the context.

Adaptation process (see figure 3) is based on *adaptation rule* (figure 3) which is specified according to the arrival of context events (context change) that trigger the adaptation rule, the sensors that produce those events, the data the rule accesses (read/write), the transformations of the UI needed in order to apply the rule, and the context precondition.

| Group | Context Events | Context adaptation operation |
|---|---|---|
| user | Visual Impairments | Prioritising audio context, text-to-speech transmoding |
| | Auditory Impairments | Prioritising video context, audio-to-text transmoding, subtitle presentation.<br>Prioritising a selected area of visual content in a scene (e.g. focus on lecturer lips) |
| | Preferences: if a user wants to watch highlights | Summarising the session |

Table 2. Illustration of possible content adaptation operations according to user context (extract from [7])

The adaptation process is based on rules that correspond to the application a set of principles according to the reception of any contextual information, about (a) User characteristics (see table1) (b) Environment characteristics and (c) Devices characteristics. The adaptation rule is triggered by context events. These context events represent the different changes in the context detected by more than one sensor (hardware or software). The transformation is applied on UI after application of adaption rules. We can use at this level the approach of Model Driven Architecture MDA [30] which is based on a set of transformation between the the Platform Independent Model (PIM) of the proposed MDA approach and the Plateform Specific Model (PSM).

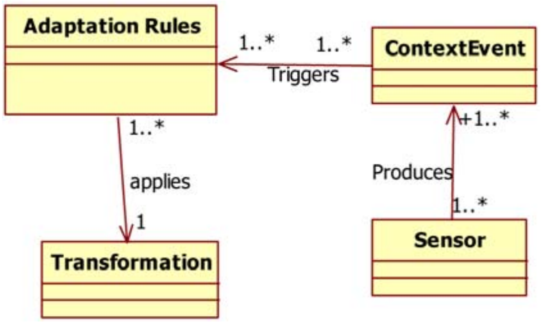

Figure 3. Metamodel for Adaptation Rules (extract from [28])

There is the need to consider the user as a central and fundamental part on context-aware applications. Through this consideration of user, it will be possible to adapt content according to user preferences or physical characteristics as auditory or visual impairments. Users differ in a wide range of variables; (a) profiles including identity, demographic characteristics, ability/disability (deaf, visual impaired person, etc.), and education, cognitive skills; (b) preferences; (c) location where executing task. Useful user contextual information belongs to user info (name, gender, age, and role), user preferences (audiovisual, presentation, etc.), usage history and also content adaptation authorizing information. We propose the use of ontology in order to support context modelling and to provide a formal semantics to context knowledge. In the developed ontology, we must to distinct between user preferences, user profile information (name, genre, age, etc.) and also user impairments information. User Profile is static information but a preference is dynamic information which dependent on the current situation or event [7]. Figure 4 depicts the user context as a central core of our context-aware system which is associated to relevant user concept elements such as Impairment, profile, preference, activities, history, etc.

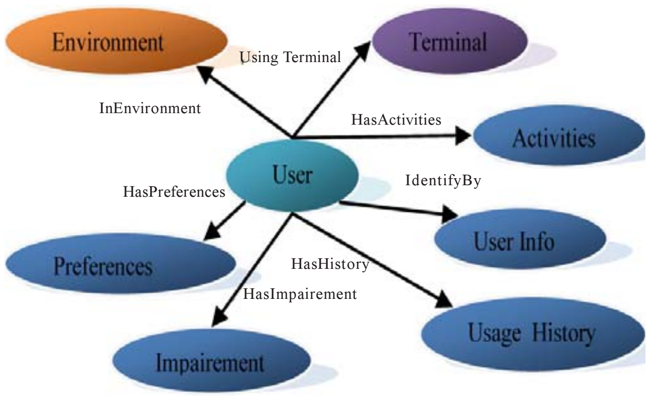

Figure 4. User Context-aware (inspired from [7])

## 6. Conclusion

In this article, we have tried to survey some of relevant context-aware architectures that were proposed in others works to support context awareness applications. Context sensitive process is very crucial feature for ubiquitous computing applications. The specification of model-based adaptation based on adaptation rules is a great interest in our contribution since there are more and more approaches supporting the model-based approach and also, there are many approach of context-awareness system. In this paper, we have contributed to a global solution of our problematic which is adaptation of user interface according to context change in order to ensure accessibility and a satisfaction of disable people.

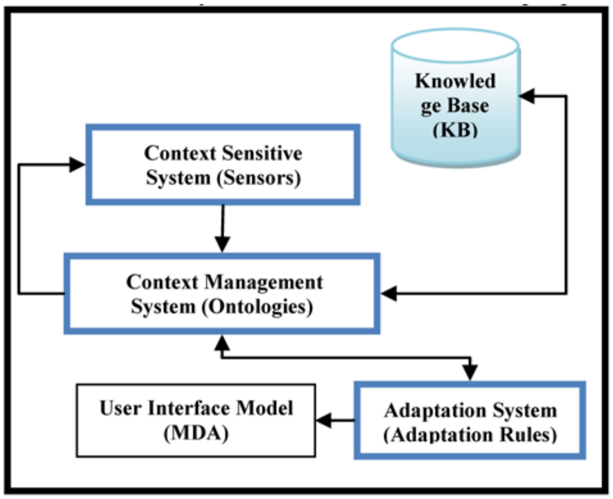
Knowled ge Base (KB)
Context Sensitive System (Sensors)
Context Management System (Ontologies)
User Interface Model (MDA)
Adaptation System (Adaptation Rules)

Figure 5. A Very Generic Structure of Our Approach

## References

[1] Dey, A. K., Abowd, D. G., Salber, D. (1999). A Context-Based Infrastructure for Smart Environments, *In*: Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE '99), Dublin, Ireland, December 13-14, p.114-128.

[2] Schmidt, A., Aidoo, K. A., Takaluoma, A., Tuomela, U.,Van Laerhoven, K.,Velde, W. V. (1999). Advanced Interaction in Context, *In*: Proceedings of HUC'99, Germany.

[3] Soylu, A., Causmaecker, P., Desmet, P. (2009). Context and Adaptivity in Pervasive Computing Environments: Links with Software Engineering and Ontological Engineering, *Journal of Software*, 4 (9) 992-1013.

[4] Dey, A. K., Abowd, D. G., Salber, D. (2001).A Conceptual Framework And A Toolkit For Supporting The Rapid Prototyping of Context-Aware Applications, *Human-Computer Interaction*.

[5] Dey, A. K. (2000). Providing Architectural Support for Building Context-Aware Applications, Phd Thesis, Georgia Institute of Technology.

[6] Schilit, B., Adams, N., Want, R. (1994). Context-Aware Computing Applications, *In*: Workshop on Mobile Computing Systems and Applications WMCSA, p. 85–90.

[7] Vitor Hugo, B. (2008). Ontologies for Context-Aware Applications, Porto, March.

[8] Stephanidis, C. (2000). From User Interfaces for All to an Information Society for All: Recent Achievements and Future Challenges, 6th ERCIM Workshop, User Interfaces For All, Florence, Italy.

[9] Stephanidis, C., Paramythis, A., Sfyrakis, M., Stergiou, A., Maou, N., Leventis, A., Paparoulis, G., Karagiannidis, C. (1998). 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, p. 153-166.

[10] Akoumianakis, D., Savidis, A., Stephanidis, C. (2000). Encapsulating Intelligent Interactive Behaviour in Unified User Interface Artefacts, Interacting With Computers, 12 (4) 383-408.

[11] Thevenin, D. (2001). Adaptation En Interaction Homme-Machine: Le Cas De La Plasticité, Phd Thesis, University of Joseph Fourier, Grenoble1.

[12] Thevenin, D., Coutaz, J. (1999). Plasticity of User Interfaces: Framework and Research Agenda, *In*: Proceedings of Interact'99.

[13] Biegel, G., Cahill, V. (2004). A framework for developing mobile, context-aware applications, *In*: Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communication, p. 361 365.

[14] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Souchon, N., Bouillon, L., Florins, M., Vanderdonckt, J. (2002). Plasticity Of User Interfaces: A Revised Reference Framework, *In*: Proceedings of the first international workshop on Task Models And Diagrams For User Interface Design, p. 127–134.

[15] Calvary, G., Coutaz, J., Thevenin, D. (2001). Supporting Context Changes For Plastic User Interfaces: A Process And Mechanism, *In*: Proceedings of AFIHM-BCS Conference on Human-Computer Interaction IHM-HCI.

[16] Rey, G., Coutaz, J. (2004). Le contexteur: Capture et distribution dynamique d'information contextuelles, Mobilité et Ubiquité (Ubimob04), ACM. Publication, p. 131-138, Nice, France.

[17] Tao, G., Hung, K. P., Da, Q. Z. (2005). A Service-Oriented Middleware For Building Context-Aware Services, *Journal of Network and Computer Applications*, 28 (1), January.

[18] Gaëtan Rey. (2005). Contexte en Interaction Homme-Machine: le contexteur, Phd Thesis, Joseph-Fourier-Grenoble I University, September.

[19] Chen, H., Finin, T., Joshi, A. (2003). Semantic Web in a Pervasive ContextAware Architecture, *In*: Artificial Intelligence In Mobile System.

[20] Chen, H., Finin, T., Joshi, A. (2004). A Context Broker for Building Smart Meeting Rooms, *In*: International Conference on Mobile and Ubiquitous Systems: Networking and Services.

[21] Harry Chen. (2003). An Intelligent Broker Architecture for Context-Aware Systems, Phd Thesis, University of Maryland Baltimore County, January.

[22] Harry Chenn. (2004). http://cobra.umbc.edu/about.html.

[23] Lopez, J. V., Montero, F., Fermandez, C. A., Lozano, M. (2003). Towards Adaptive User Interface Generation: One Step Closer to People, *In*: Proceedings of ICEIS (3), p. 97-103.

[24] Victor, L. J., Francisco, M. S., Fernando, R. (2009). Designing user interface adaptation rules with T: XML, IUI, p. 383-388, ACM.

[25] Miraoui, M., Tadj, C., Ben Amar, C. (2008). Architectural Survey of Context-Aware Systems in Pervasive Computing Environment, *Ubiquitous Computing and Communication Journal* (*UBICC*), 3 (3).

[26] Ryan, N., Pascoe, J., Morse, D. (1998). 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.

[27] Brown, P. J. (1998). Triggering information by context, *Personal Technologies* 2 (1) 1-9.

[28] Limbourg, Q., Souchon, N., Vanderdonckt, J. (2002). Task Modelling In Multiple Contexts Of Use, Mars.

[29] Tarek Chaari. (2007). Adaptation d'applications pervasives dans des environnements multi-contextes, Phd Thesis, The National Institute of Applied Sciences in Lyon, September.

[30] Hlaoui Ben Daly, Y., Jemni Ben Ayed, L. (2009). Patterns for Modeling and Composing Workflows from Grid Services, The 11th International Conference on Enterprise Information Systems (ICEIS 2009), LNBIP, 24, 615-626, Springer..