# Towards Context Adaptation in Ubiquitous Applications

Mohamed Sbai
Department of Science Computer
Faculty of Sciences
Tunisia
e-mail: mohamed.sbai155@gmail.com

Faouzi Moussa
Department of Science Computer
Faculty of Sciences
Tunisia
e-mail: faouzimoussa@gmail.com

Hajer Taktak
Department of Science Computer
Faculty of Sciences
Tunisia
e-mail: taktakhajer@gmail.com

***Abstract*—Ubiquitous computing is considered one of the most impactful scientific achievements in the last decade. This conception created tremendous revolution in the end-user interactions through the concept of context-awareness. Ubiquitous computing offers a new opportunity to redesign the pattern of conventional solutions where it can easily tailor its processes upon existing contextual situations. Several theoretical architectures have been developed to enable context-awareness computing in pervasive settings. In order to exceed the limits of these related works, we will make a comparative study of these architectures and we will propose our solution. The objective of this article is to propose an adaptation architecture which aims to design and validate a contextual model for ubiquitous systems in order to offer services adapted to the preferences of the user.**

***Keywords- Context; Ontology; Adaptation.***

## I. Introduction

The study of the literature shows that sensitivity to the context has become an essential element for the implementation of adaptive services in ubiquitous interactive applications. The context is no longer a pre-established and predefined model when designing interactive application systems, but rather a dynamic description of the current situations which can be discovered in the context data and which can change dynamically according to changes in requirements and user preferences. Therefore, context information tends to be incorrect because it does not exactly reflect the actual state of the observed entity, incomplete when certain aspects of the context are missing, or even ambiguous if several values are collected and do not entirely correspond to each other [1]. For example, two separate localization devices can provide values corresponding to overlapping regions, having different levels of precision or even being inconsistent if they present contradictory information. When the context information collected is imperfect and uncertain, there is a risk of basing a decision on incorrect information [2]. In addition, reasoning on uncertain information induces very high reasoning costs due to the complexity of the solutions to be implemented [3][4]. Ubiquitous applications must be able to run in different contexts of use depending on the user's environment, their profile, the terminal they are using, or their location.

In order to meet the different requirements for adapting to dynamic changes in contextual situations, we propose in this article our architecture for adapting to context in ubiquitous applications.

In this work, we distinguish four main components: the acquisition of the context, the representation, the reasoning and the application. Context acquisition functions allow interrogation of physical devices to obtain contextual data. Given the various characteristics of contextual information such as heterogeneity, dynamics and imperfections, it is essential to define a model to describe this data. In addition to context information, reasoning schemes are used to develop applications and services for specific needs.

The article is organized in several parts: In Section 2, we present a detailed study of the different approaches to contextual adaptation. In Section 3, we make a comparative study of these approaches. Section 4 positions our proposal in relation to related work. We describe later, in Section 5, our context modeling method. Then, we present in Section 6 our method of adaptation to the context. In Section 7, we present a performance study of our approach. Section 8 concludes the article and presents future work on the subject.

## II. Related work

In recent years, many context-dependent infrastructures have been developed to manage ubiquitous systems. However, these infrastructures differ a lot in their architectures and implementations. They depend on the requirements of the systems and the process of acquiring, transforming and processing context information. These systems are different not only in architecture, which is generally organized in layers, but also in the model of the context adopted.

Pung et al. [5] proposed an architecture that provided context information to context dependent mobile services. This approach allowed applications to integrate several online services for their specific areas of context. It offered the ability to easily integrate and reuse components in the system such as new sensors. It also made it possible to abstract from the heterogeneity of the data sources.

Chen [6] proposed an architecture based on multi-agent systems. Its operation was essentially based on an intelligent agent called "Context Broker" who owned and managed a context model. This agent was composed of four main elements: the context knowledge base, a context reasoning engine, a context acquisition module and a private data management module. The major advantage of this architecture is the use of the ontology which, by definition, allows the sharing of data and the reasoning on its content.

 ISBN: 978-1-61208-761-0 

Rouvoy et al. [7] proposed an architecture supporting self-adaptive mobile and context-aware applications. This architecture could be adapted to the dynamic changes of the environment (e.g., location, network connectivity) in order to satisfy the user requirements and device properties (battery, memory, CPU). The adaptation process defined is based on the principles of planning-based adaptation. This work has not taken into account the multimodal aspects for user-machine interaction and the contextual information that could be gathered by the distributed action mechanism.

More recently, Taing et al. [8] have proposed an architecture based on the Context Toolkit infrastructure; it supported the change of XML files and fire events to an unanticipated adaptation component that could be associated to fully described situations, including time, place and other pieces of context. This work used a transaction mechanism to ensure uniformly-consistent behavior for every smart object executing inside a transaction and supported only a notification as an action type without multimodality aspects that could be triggered as a result of situation identification and smart event detection.

Ghiani et al. [9] proposed an architecture that aimed to provide adaptable interfaces, allowing end users to easily and autonomously customize the behavior of their applications. It provided an environment for users to easily specify rules in the form of event / action pairs by limiting these rules to the contextual elements that are actually possible in the user's situation. However, this approach did not present adaptation rules as such.

Miñón et al. [10] proposed a system called "Adaptation Integration System". This system aimed to integrate accessibility requirements for people with disabilities by including adaptation rules in the development process.

## III. DISCUSSION

After studying the related work, our comparison will be made on two fundamental criteria: the modeling of context elements and context adaptation parameters. At this level, we prove the need to use these two criteria while referring to the different roles of each in the response to the main context modeling objectives when designing a pervasive interactive application.

### *A. The first criterion: The modeling of context elements*

The modeling of contextual elements is one of the important features for fostering and improving context sensitivity in pervasive environments. Indeed, this modeling is considered as an essential step for the design and development of interactive systems.

In Table I, we present a classification of all approaches according to their degree of modeling.

In Table II, we start our study by designating the set of contextual elements that were most often used in the context sensitivity. One element of the context is the one that describes the points of adaptation. It is part of the descriptions of the characteristics of the environment which describes reference parameters or preferences. The first step is to study the sensitivity to the context and to define the constituent elements of the current environment. These elements are used later in the adaptation phase (Section 6).

### *B. The second criterion: Context adaptation parameters*

The design of interactive applications in pervasive environments requires consideration of the set of adaptation parameters (Table III). So, these applications can be used on terminals, by users, in specific environments and locations. In addition, these applications must deal with the dynamic change of context of use to achieve activities and achieve the appropriate objectives of users.

TABLE I. COMPARATIVE STUDY ON THE DEFINITION OF DATA STRUCTURES

| Related Works | Modeling approach | Complex structure definition |
|---|---|---|
| [7][9][10] | - | No |
| [5] | + | Yes |
| [6][8] | ++ | Yes |

TABLE II. COMPARATIVE STUDY OF CONTEXT ELEMENTS

| Related Works | Description of context-sensitive elements | | | | | | |
|---|---|---|---|---|---|---|---|
| | *User* | *Environment* | *Terminal* | *Location* | *Service* | *Activity* | *Time* |
| [5][6][8] | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| [7][9][10] | Yes | Yes | Yes | No | No | No | Yes |

TABLE III. COMPARATIVE STUDY AT THE CONTEXT ADAPTATION PARAMETERS LEVEL

| Related Works | Context adaptation parameters | | | | |
|---|---|---|---|---|---|
| | *Logical reasoning* | *Management* | *Adaptation type* | *Adaptation technique* | *Action Mechanism* |
| [6][8] | Yes | Centralized | Integration | Reasonner | Centralized |
| [5] | No | Distributed | Reaction | Metamodel | None |
| [7][9][10] | Yes | Distributed | Integration | Formel object | Centralized |

 

On the other hand, these approaches have limits in terms of modeling and adapting of the context. In the approach presented in [5], the modeling did not take into account neither the description of the relevant situations nor the adaptation actions. On the other hand, architecture makes it possible to describe the dependencies between the observable contexts through properties or by using the notion of inheritance. The limits of the approach presented in [5] consist of an object-oriented model for context management. This model is not easily extensible and offers limited expressiveness. In addition, access to relational databases takes time, which negatively affects the performance of the infrastructure. The limit of approaches [6][8] is the impossibility of recording the background history, which does not allow the use of learning algorithms to improve the treatment of the context. The model of entity association adopted is less expressive than the ontological model. To overcome the various limitations present in these approaches, we will present, in the following section, our proposed approach.

## IV. CONTRIBUTION

After an in-depth study of the different architectures presented in the previous section, we note that an important challenge in the field of ambient computing concerns the optimization of the use of context management mechanisms and the adaptation of interactive applications to the diversity of ubiquitous environments. Our objective is to propose an adaptation architecture which aims to design and validate a contextual model for ubiquitous applications in order to offer services adapted to the preferences of the user. Our proposed architecture is formed by the following layers (Figure 1):

### *A. Sensor*

Contains all data sources which can provide useful information for the context.

### *B. Context processing*

Used to identify and model the context (User, Environment, Platform) from data of the various sensors.

### *C. Adaptation control*

Allows us to detect adaptations and to implement them, it is made up of two modules: adaptation analysis and adaptation decision. The adaptation approach involves considering the different facets of the data context to adapt them to their needs. This is the basic concept that influences the process of the system.

### *D. Interface generation*

Once the necessary elements of the interface are identified, the next step is to specify the graphical interface in terms of graphical objects and display. Indeed, the last step is devoted to automatic generation of the interface.

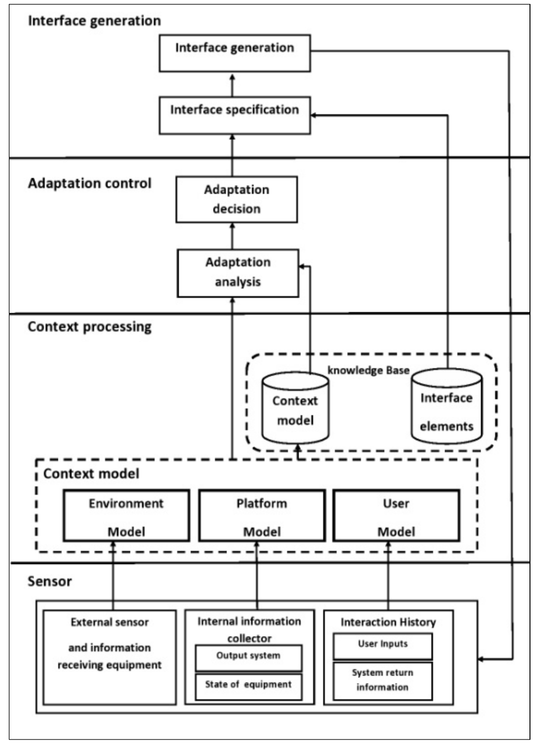


Figure 1. Proposed architecture.

## V. CONTEXT MODELING

In this section, we present our context model based on ontology. To begin, we will present the core model of this ontology from a scenario. Then, we will detail all the ontologies of our domain. The main objective is to build a context model to support the functionalities of our context-aware adaptation platform. A context model for the adaptation platform must contain the usage contextual state: environment, platform, user context. It must be extensible by these applications. Our adaptation platform uses it to identify adaptation situations. It must find knowledge of contextual data, the state of execution of the platform, etc. This knowledge helps deploy the platform and the application.

### *A. Scenario in a context-aware environment*

We consider, for example, users who use computing resources for daily activities in a context-aware environment. Olivier and David are teachers at the university. In the morning before leaving for work, one uses the tablet, the other uses the phone to consult the news and communicate with their family remotely by video chat. When they arrive at the university, they use their PC from their office for their research activities. They have access to some university services to work. In the afternoon, they have a lecture at another university. They take the company car to get there and they use the computer built into the car for navigation.

 ISBN: 978-1-61208-761-0 

This scenario shows that changes in user activity or location will cause context changes. When they move from one place to another, the physical environments around them are different.

The computing resources available to them may change, for example, because of the available network connectivity. As they move from one activity to another, computing resources also change. For example, in the scenario, when they are in their offices, the resources of the university are accessible to them for their professional activities. These hardware or software resources are different from those available to them when they are at home.

## B. Core ontology

In this section, we present our context model design starting with the highest level of abstraction. Our objective is to have a state of the current context of the users in their adaptation domain and to enable the identification of adaptation situations. After developing the generic context model based on the core ontology (Figure 2), we will introduce domain ontologies, their objectives, and describe each domain.

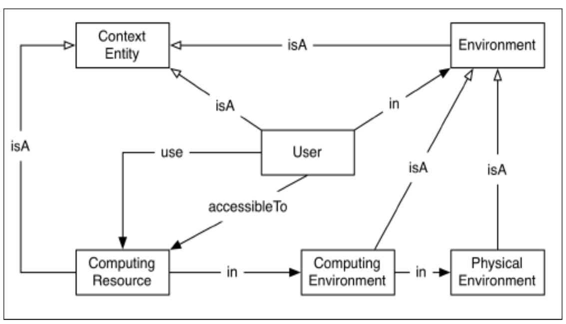


Figure 2. Core ontology: Relationships.

## C. Domain ontologies

### 1) Domain "User"

This part of the ontology (Figure 3) describes the user's current situation and profile. This information is closely linked to information of a space-time nature (place, time, mobility). The adaptation platform needs to know which devices the user is using.

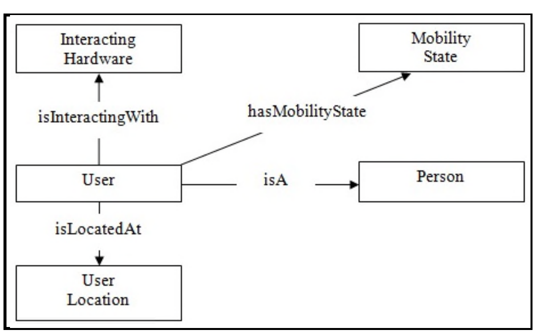


Figure 3. "User" ontology.

### 2) Domain "Environment"

A user of our platform is an individual who is surrounded by a set of physical elements and computer elements. The living environment of a user consists of the physical elements (light, location, etc.) and the computing resources (Smartphone, PC, etc.) which the user can access and which render services in the user's daily activities. The environmental ontology aims to describe these elements and their relationships (Figure 4).

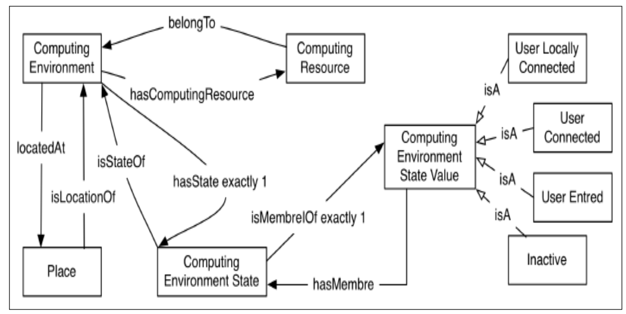


Figure 4. "ComputingEnvironment" ontlology.

### 3) Domain "ComputingResource"

The purpose of this ontology is to have knowledge of all the resources of the adaptation domain, i.e., their static and dynamic descriptions during the execution. We have identified four types of computing resources of interest for software adaptations: "Hardware", "Software", "Power", and "Network" (Figure 5).

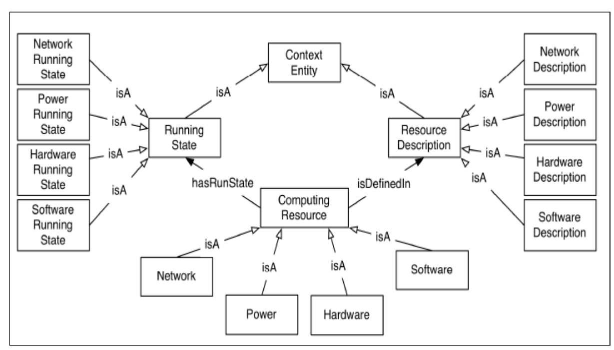


Figure 5. Structure of "ComputingResource" ontology.

# VI. ADAPTATION CONTROL

This layer allows us to detect adaptations and implement them. It is composed of two modules: adaptation analysis and adaptation decision.

## A. Adaptation analysis

This component of our adaptation platform must answer the following question: What is the current situation? This is the essential information we need to know and which will guide us to make the decision of adaptation.

Our platform enables the system to make adaptation decisions. Reasoning about situations is a very broad area of research. In context-aware applications, researchers define a situation as an external semantic interpretation of sensor data [11].

Next, we present our situation model. This situation model is designed to represent adaptation situations. We once again use an ontology to design this model because we want a semantic representation that is easy to extend and also a tool adapted to reasoning. An adaptation situation is either detected by a lack of resource adequacy, identified by the application, or identified by the platform. An adaptation situation is a result of the states of the current context.

Our goal is to design a model to support the identification of the situations defined by the application and those defined by the platform. An adaptation situation in our platform (Figure 6) corresponds to one of three broad categories: "GeneralASituation", "PlatformASituation" and "AppASituation".

The general situations ("GeneralASituation") correspond to an imbalance between the needs of the user and the resources available in the current environment. The platform adaptation situations ("PlatformASituation") are related to the structure and operating logic of the platform itself. The adaptation situations of the application ("AppASituation") are only detected to be transmitted to the application that will process them according to its business logic. The "PlatformASituation" and "AppASituation" categories are extensible. The platform and applications will be able to extend these situations according to their needs. A situation is linked to a description and a cause. The description is either a natural language description text or a context element state (user activity, environment, network presence, etc.). A situation of adaptation can be produced by a chain of reasoning from the state of the resources and the user or by a chain of reasoning from a logical reasoning on the ontology. Both cases correspond to an imbalance between the needs of the user and the resources available in the current environment.

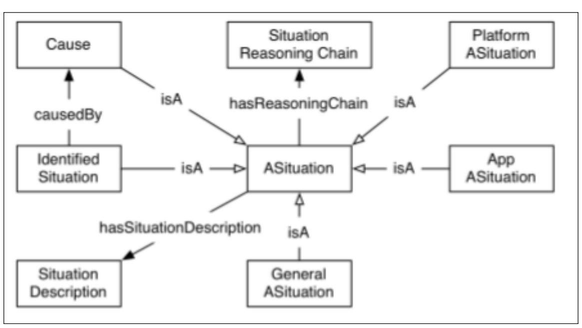


Figure 6. Situation Ontology.

*B. Adaptation decision*

Depending on the different types of adaptation platforms, the definition of the decision-making changes. For example, in Event, Condition, Action (ECA) -based platforms, decision-making means that when the platform captures context-change events, if these events meet the conditions set out in the predefined rules, an adaptation will be made by the defined actions in these same rules. For heuristic platforms like CAMPUS [11], a decision-making is equivalent to a software component parameterization according to the available resources. Our platform is based on semantic situations; this allows us to treat the context with a higher level of vision. The adaptation decision is to find a new application architecture (Figure 7) applicable in the current context to respond to the identified adaptation situation. When an adaptation situation is identified, a "new adaptation situation identified" notification is sent to the "decision makers".

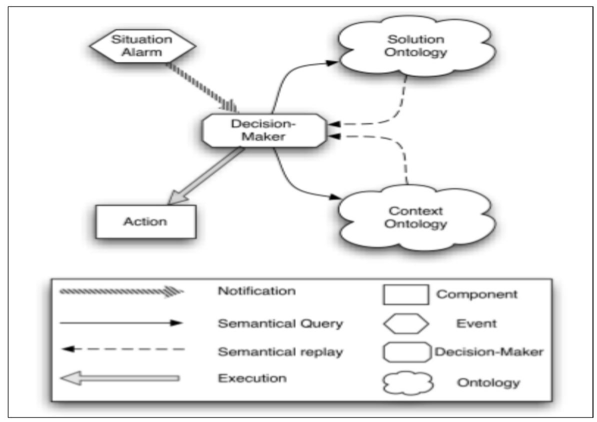


Figure 7. Architecture for Adaptation Decision Making.

The decision makers are notified of the addition of a new alarm "ASituation" in the knowledge base. (ASituation is the concept associated with adaptation situations). They are then in charge of the decision making adaptation for which they use both ontologies (situation and solutions) to reason and find a solution to the notification received.

The solution ontology contains knowledge of solutions related to situations and their causes. The situation ontology contains the situation information. When the "decision makers" have chosen a solution, it will deploy "Actions" to make the adaptation. An adaptation solution contains the processing logic specific to the associated situation.

## VII. PERFORMANCE STUDY

In this section, we present the results of the performance study of our approach. First, in order to estimate the effectiveness of the adaptation, we measured the user satisfaction rate. In a second step, we measured the gain in memory space generated by the adaptation.

*A. User satisfaction rate*

We asked a panel of users to do a few tasks including browsing the Web pages they used to visit. After adapting these pages using our adaptation method, we asked them to locate specific information in the original page and in its adapted version. After completing these tasks, we asked them to respond to a questionnaire. In this questionnaire, the first four questions relate to navigation. The last question concerns the harmony of the structure of the Web page. A score is assigned to each question to assess the level of satisfaction of these users (8 being the highest score and 1 the lowest score). In Figure 8, we present the means of the scores obtained for each question posed to the panel of users. We note that the averages for the adapted version of the page exceed the averages for the original version.

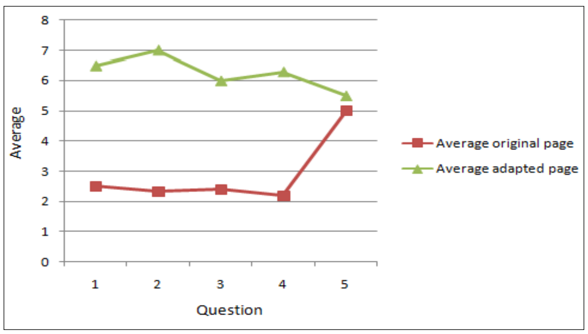


Figure 8. Satisfaction rate of users for the adaptation obtained.

*B. Memory space used*

In Figure 9, we present the results of the study of the memory space occupied before and after adaptation. The results show that adaptation plays an important role in reducing the memory space occupied by media objects.

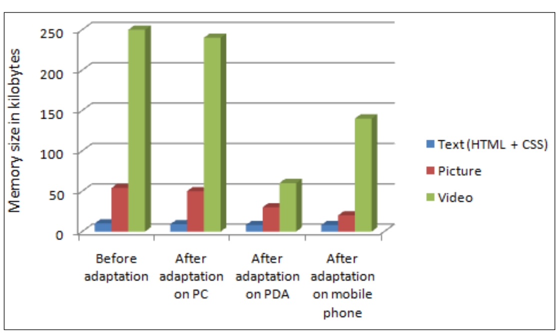


Figure 9. Component memory space before and after adaptation.

## VIII. CONCLUSION AND FUTURE WORK

Building on recent advances in the world of mobile technology, pervasive environments are attracting growing interest. However, limitations linked to the resources of mobile terminals, the heterogeneity of devices and data, the multiplicity of requests and user preferences generate undesirable problems.

For that, the objective of this article is to propose an adaptation architecture which aims to design and validate a contextual model for ubiquitous systems in order to offer services adapted to the preferences of the user. In a mobile environment, decision-making cannot guarantee that the best solution will be chosen for a given adaptation situation during execution.

In future research, in order to choose a better solution, we will consider setting up a passive service responsible for long-term analysis of past (historical) decision-making without intervening directly in the adaptation cycle. This service could then refine decision making by adding adaptive situations in the ontology of situations and by associating them with specific solutions in the ontology of solutions. This new information being taken into account in future decisions, it could allow the learning decision-making system to evolve.

## REFERENCES

[1] W. Dargie and T. Hamann. “A distributed architecture for reasoning about a higher-level context”. In IEEE International Conference on Wireless and Mobile Computing, Networking and Communications. (WiMob 2006), Canada, pp. 268–275, July 2006.

[2] A. Dey, G. Abowd, and D. Salber, “A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications”. Special issue on context-aware computing in the Human-Computer Interaction Journal, April 2001.

[3] K. Henricksen and J. Indulska, “ Modelling and using Imperfect Context Ifnformation”. In Proc. 1st PerCom Workshop CoMoRea, USA, pp. 33–37, March 2004.

[4] A. Ranganathan, J. Al-Muhtadi, and R. Campbell, “Reasoning About Uncertain Contexts in Pervasive Computing Environments”. IEEE Pervasive Computing, vol. 3, pp. 10–18, April 2004.

[5] H. Pung and D. Q. Zhang, “A service-oriented middleware for buildfing context-aware services”, vol. 28, pp. 1-18, January 2005.

[6] H. Chen, "An Intelligent Broker Architecture for Pervasive Context-Aware Systems", PhdThesis, University of Maryland, Baltimore County, December 2004.

[7] R. Rouvoy, P. Barone, Y. Ding and F. Eliassen,” MUSIC: Middleware Support for Self-Adaptation in Ubiquitous and Service-Oriented Environments”. In Software Engineering for Self-Adaptive Systems, Germany, January 2009.

[8] T. Taing, M. Wutzler and T. Spriner, “Consistent Unanticipated Adaptation for Context-Dependent Applications”. In Proceedings of the 8th International Workshop on Context Oriented Programming, Italy, pp. 33-38, July 2016.

[9] G. Ghiani, M. Manca, F. Paterno and C. Santoro, “Personalization of Context-Dependent Applications Through Trigger-Action Rules”. ACM Transactions on Computer-Human Interaction (TOCHI), vol. 24(2), April 2017.

[10] R. Miñón, F. Paternò, M. Arrue and J. Abascal, ”Integrating adaptation rules for people with special needs in model-based UI development process”, Universal Access in the Information Society, vol. 15, pp. 153-168, March 2016.

[11] E. Wei and A. Chan,”Campus : A middleware for automated context-aware adaptation decision making at run time”. Pervasive and Mobile Computing, vol. 9, pp.35-56, February 2013.

 ISBN: 978-1-61208-761-0 