The Thirteenth International Conference on Advances in Computer-Human Interactions (ACHI 2020)

Towards Context Adaptation in Ubiquitous Applications

Presenter:

Mohamed SBAI

PLAN

1. Introduction

- 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 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.

2. 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.

2. Related work

- 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.

2. Related work

- 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.

2. Related work

- 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.

2. Related work

- 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.

3. Discussion

After studying the related work, our comparison will be made on two fundamental criteria:

a) 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.

3. Discussion

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

3. Discussion

b) 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 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

4. Contribution

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):

4.1. Sensor

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

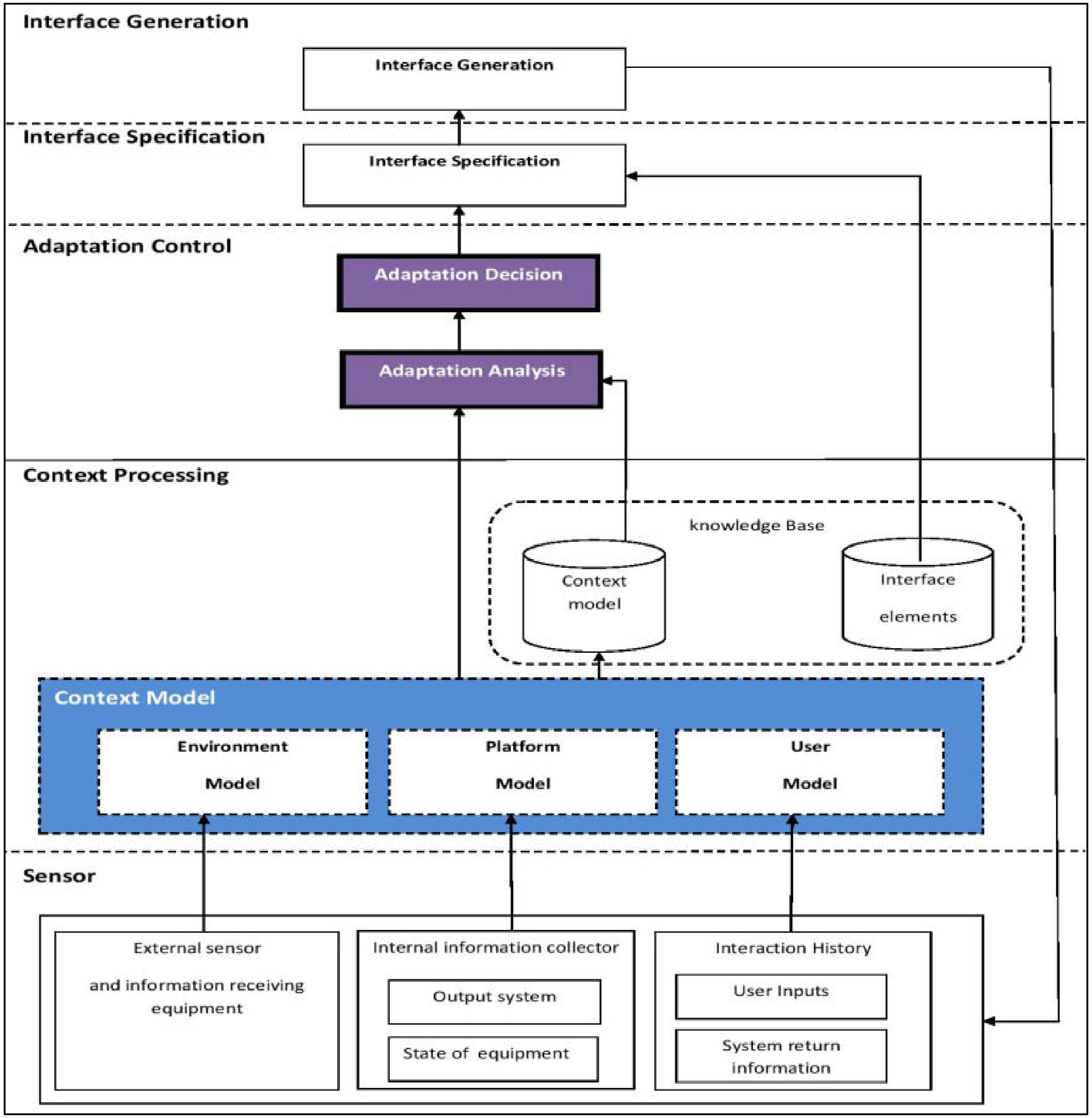

Figure 1. Proposed architecture.

4. Contribution

4.2. Context processing

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

4.2.1 Context modeling

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

4. Contribution

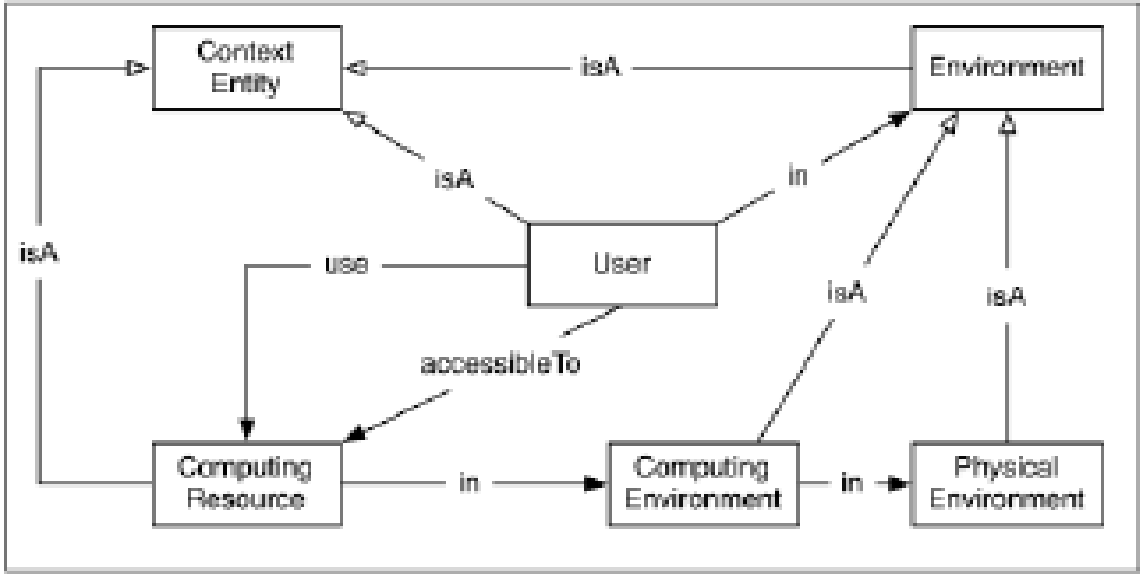

Figure 2. Core ontology: Relationships.

4. Contribution

a) 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.

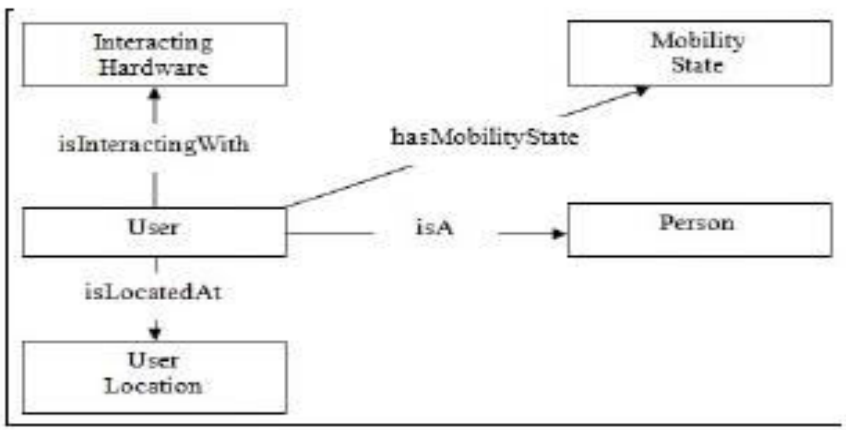

Figure 3. "User" ontology.

4. Contribution

b) 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).

4. Contribution

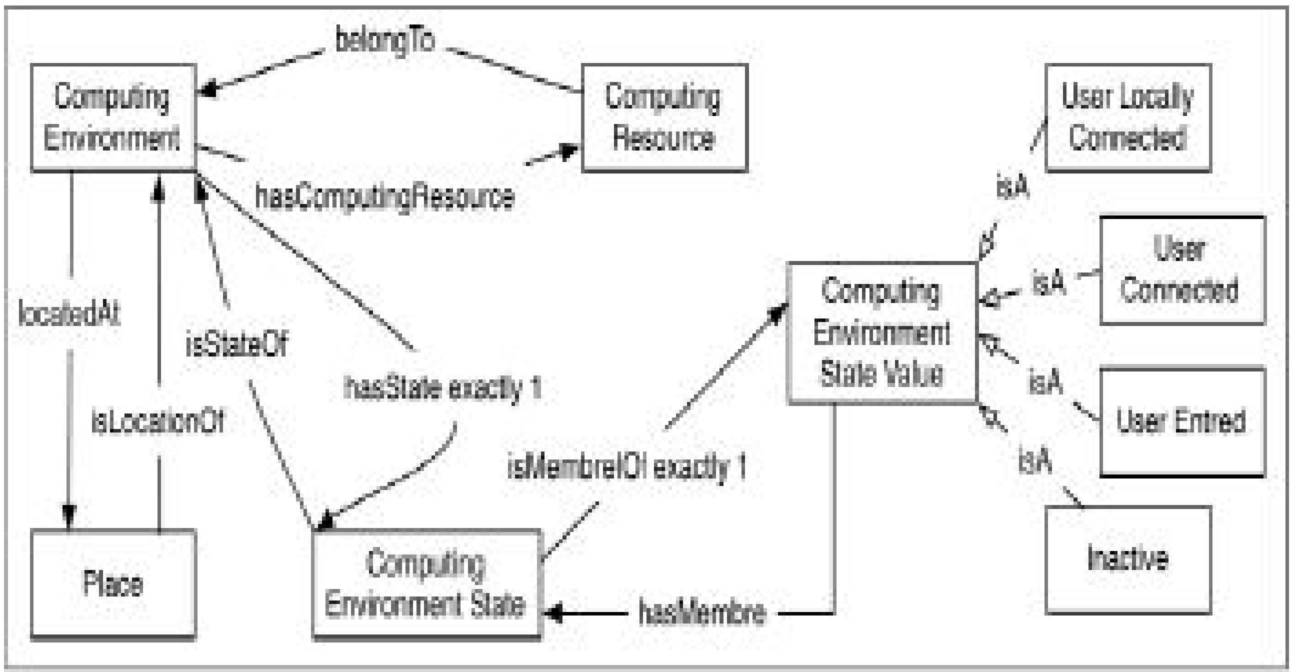

Figure 4. "ComputingEnvironment" ontlology

4. Contribution

c) 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).

4. Contribution

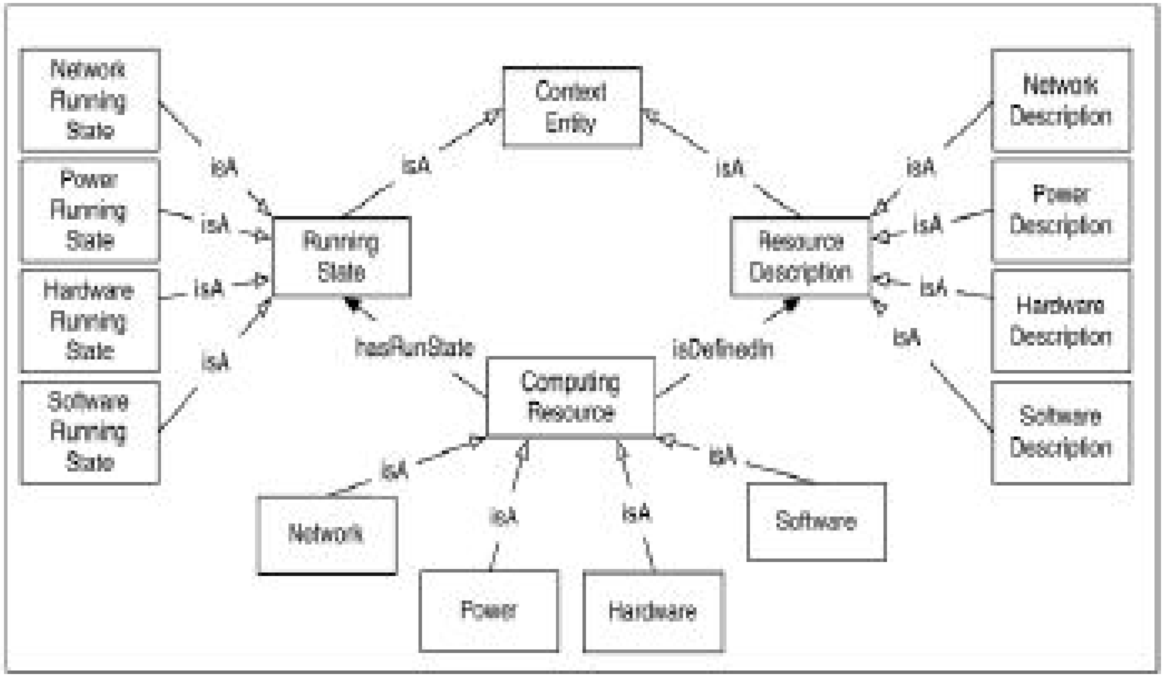

Figure 5. Structure of "ComputingResource" ontology.

4. Contribution

4.3 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].

4. Contribution

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".

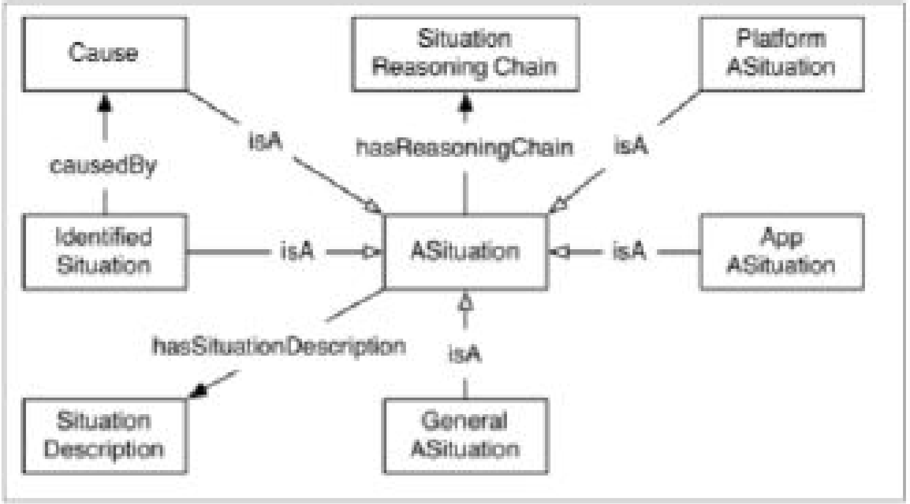

Figure 6. Situation Ontology

4. Contribution

b) Adaptation decision

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".

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.

4. Contribution

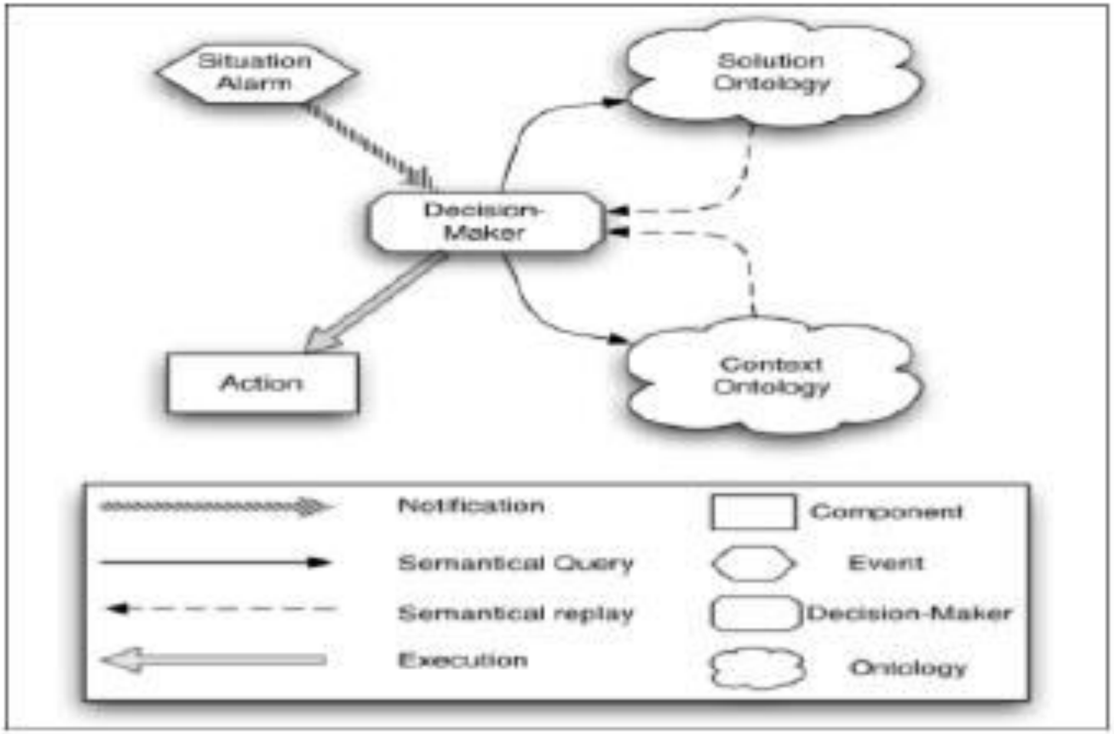

Figure 7. Architecture for Adaptation Decision Making.

5. Performance study

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). averages for the original version.

5. Performance study

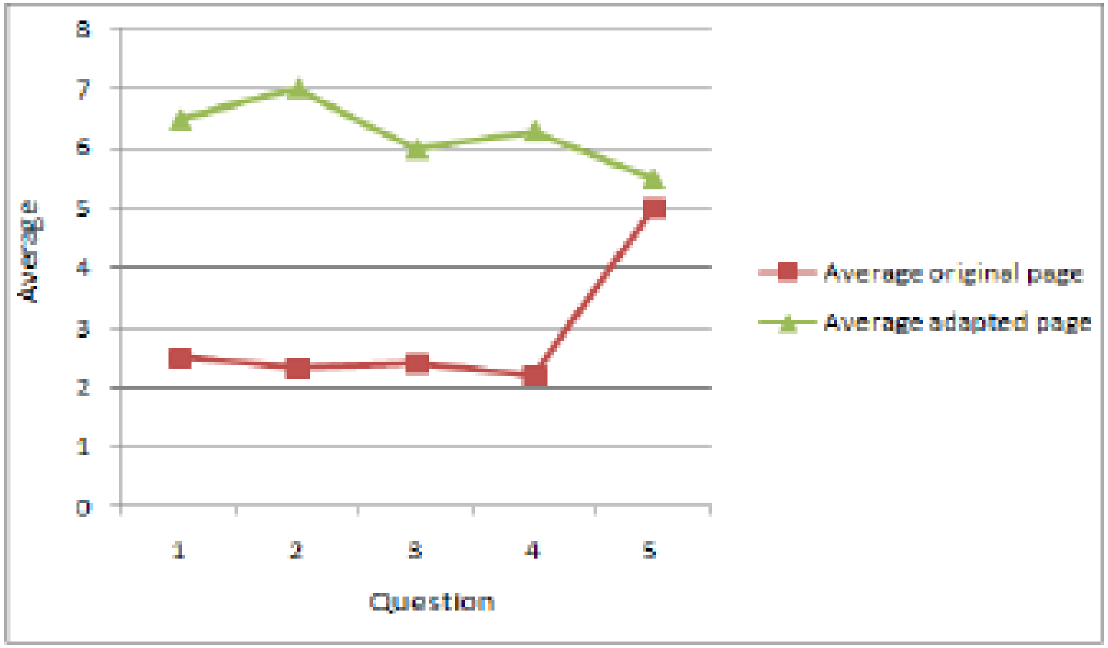

Figure 8. Satisfaction rate of users for the adaptation obtained.

5. Performance study

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.

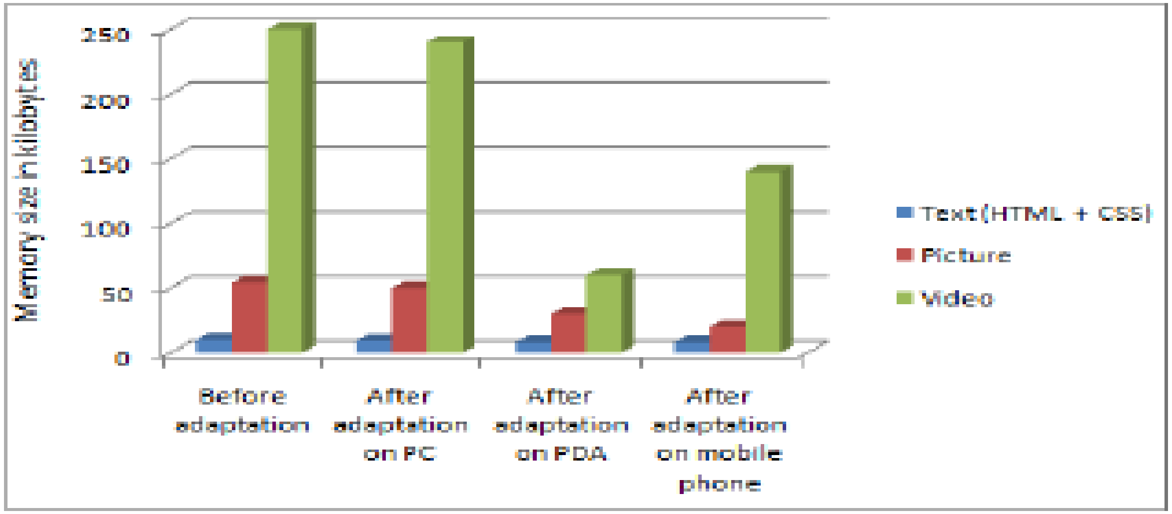

Figure 9. Component memory space before and after adaptation.

6. 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.

6. Conclusion and future work

- 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 contextaware applications". Special issue on context-aware omputing 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.