



HAL
open science

Towards an Adaptive Web Information System Based on Web Services

Bouchra Soukkarieh, Florence Sèdes

► **To cite this version:**

Bouchra Soukkarieh, Florence Sèdes. Towards an Adaptive Web Information System Based on Web Services. 4th International Conference on Autonomic and Autonomous Systems (ICAS 2008), Mar 2008, Gosier, Guadeloupe. pp.272-277, 10.1109/ICAS.2008.28 . hal-03771528

HAL Id: hal-03771528

<https://hal.science/hal-03771528v1>

Submitted on 9 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards an Adaptive Web Information System Based on Web Services

B. Soukkarieh, F. Sedes, *Member, IEEE*

Abstract— The appearance of wireless networks and mobile computing requires building context-aware Information Systems. These systems are more and more based on the usage of Web Services, which are designed to support interoperable application-to-application interaction over the web.

Our aim is to build an Adaptive Web Information System based on Web Services. Our contribution aims at proposing a new architecture of Web Information Systems, supporting the adaptation process to the user context and returning to the user a list of Web Services adapted to his context. This architecture is based on an extension of AHA! architecture adding an adaptation layer containing various components dedicated to the context adaptation.

Index Terms—Adaptation, Web Service, Web Information System, Context.

I. INTRODUCTION

Nowadays, the propagation of heterogeneous information resources and the diversity of devices used by the user to access these resources (laptops, mobile devices, etc) is creating a new requirement of Web Information Systems (WIS). This need is presented by the system capacity to deliver to the user information that is not only adapted to his request, but also to his context. Such systems are characterized by their architectures that can be distributed or centralized.

In our work, we are interested in Distributed Information Systems. Several architectures have been proposed, such as “Adaptive Hypermedia Architecture (AHA!)” that is regarded as a Web Information System (WIS) [15]. However, AHA! does not support the user’s mobility; it only takes into account his characteristics (preferences, interests, etc.).

Indeed, WIS are more and more based on the usage of Web Services (WS). But, the classical architecture of Web Services does not take context adaptation into account.

In order to integrate the concept of adaptation in WIS and provide to the user a list of services adapted to his context, we propose a simple and general architecture extending the AHA!’s one. In this architecture, we integrate the classical Web Services architecture with the AHA! architecture adding to these architectures an adaptation layer containing various components dedicated to the context adaptation. So, our aim

is to build an Adaptive Web Information System based on Web Services. The rest of this paper is as follows. In section 2, we present a state of the art. This state of the art is organized in three paragraphs; first we define the context by showing how this concept is evaluated in recent years especially in the field of context-aware applications. Then, we present the definition of Web Information System, taking the AHA! system as an example. Finally, we present the definition of Web Services. In section 3, we detail our contribution which aims to propose a new architecture of Web Information System. This new architecture can, on one hand, support the adaptation process to the user context and on the other hand, return to the user Web Services Adapted to his context. We conclude, in section 4, by presenting some prospects.

II. STATE OF THE ART

A. Context-awareness

Various areas of research have been interested in the concept of context. Each of these areas proposes one or more definitions of this concept. With regard to computer science areas, the need for indicating the context was underlined in three principal areas of research i.e. ubiquitous and mobile computing, context-aware applications, and pervasive computing [8].

In *ubiquitous and mobile applications*, users can access data and information from a variety of devices and applications with different capabilities and computational resources; the context, in this case, is used to deliver the appropriate content using the presentation that is best-suited for the user characteristics. Therefore, in order to define the context information, such as the user’s location, environmental conditions and device characteristics are needed. This information is, in most cases, implicit and gathered from sensors and plugins that are deployed in the environment surrounding the user.

Context-aware applications use context to provide relevant information and services to the user [2]. The problem, in this case, is not only related to adapt content to the user computational capabilities. Other elements of the context, such as user’s behavior or preferences, have to be considered in order to choose the most relevant information and service for the user.

In *pervasive computing*, the context is not only used to

identify the environment and user characteristics for executing tasks, but it is also exploited to infer new knowledge about users to help them in a transparent and proactive way. Pervasive computing underlines the need to provide formal context descriptions and rules by which new context features can be extracted.

Our work is based on context-aware systems. Indeed, the first work in this area is interested only in the definition of the context, through a particularly limited vision. Schilit and Theimer [4], for example, consider only the localization of the user, the people who accompany him, the objects which encounter him, and the changes in these elements. Although the context is regarded as true centre of interest for research works of context-aware systems, we note that the first work remains rather vague on the definition of context.

Recent work in this area starts to define the context in a more general and clearer way. According to [12], for example, the context is considered as the user preferences and which brings additional information that would help to understand his preferences. Other authors preferred a more pragmatic approach, proposing more suitable definitions for the design context-aware systems.

Lemlouma [18] defines the context as “the set of all information of the environment that can influence the adaptation process and the transmission of contents towards the end-user”.

Chari [17] defines the context as “the set of the external parameters that can influence the behavior of the application by defining new views on its data and its available services”.

A generalized notion of context has been proposed by Dey: “context is any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups, computational and physical objects.” [3].

B. Web Information System (WIS)

The Information System (IS), which represents set of elements participating in the management, storage, treatment, transport and diffusion of information within an organization, is characterized by its architecture [13] which is:

- 1) **Distributed**: data are localized in physically different places (e.g.: AHA! architecture) [15], or
- 2) **Centralized**: the totality of data is localized in the same data warehouse (e.g.: AHAM architecture [14]).

We are interested in the case of Distributed Information System. Among various suggested architectures, we base on “Adaptive Hypermedia Architecture” (AHA!) which is regarded as a Web Information System (WIS). Indeed, a WIS is initially a traditional IS, and then a system based on the Web. WIS can be regarded as a Web site containing data of complex type and offers services of complex type.

Indeed, the utilisability, the performance and the maintenance are principal measurements of the quality of WIS. Nowadays, users can reach these WIS through various devices

(laptop, mobile device, etc). This heterogeneous, mobile and changing environment requires that the information and services, sent by the server, are adapted to precise conditions of their use. Therefore, the context seems a promising idea in order to increase the utilisability of the WIS.

The first WIS taking into account the context are applications of tourist guide. These applications aim to deliver to the user (a tourist who visits a city, a museum, etc) adapted contents to his localization and to the tourist activity. For example, the GUIDES application proposed by Cheverest [11] is a system designed to provide to visitors of a city, armed with mobile devices, tourist information sensitive to the context. The main goal, in these applications, is to deliver automatically information to the user. The problems dealt by these applications are in particular discovery of the localization and the adaptation of the contents.

Other research work, like that of Schilit and that of Lemlouma, rather concerns the adaptation of the presentation of these contents delivered to the user according to physical capacities of the user device. Schilit underlines the use of various techniques for the adaptation of the presentation of the Web pages.

Indeed, in spite of the importance of concept of context, the majority of context-awareness systems have a limited use of this concept. According to [17] several works are limited to analyzing small quantities of contextual information and presents solutions to very specific needs. The first work, such as [5] considered only the user localization and, still today, several systems are interested only in this aspect of the context [10] or in the aspects related to the use of the mobile devices [18].

The first objective of our work, therefore, aims to build an adaptive WIS that allows integrating a generic model of context in its structured.

AHA! is a WIS, which supports mobility of the user neither the user localization nor the aspects related to the use of the mobile devices, but takes into account only user characteristics. The architecture of this system is detailed in the following paragraph.

Adaptive Hypermedia Architecture (AHA!)

AHA! is an Adaptive Hypermedia System based on the Web, it is inspired by the AHAM model. AHA! system is divided into components (Fig. 1) [14]:

- 1) **Java servlets** are Java applications that allow dynamically to generate the pages from the local file system or from external http servers.

- 2) The **User Model (UM)** contains the user characteristics (preferences, centre of interest, etc.).

- 3) The **Domain Model (DM)** helps to describe the hypermedia content and its organization.

- 4) The **Adaptation Model (AM)** describes how the Adaptive Hypermedia must make the personalization that represents the core of this system. The Adaptation Model is represented by a set of the adaptation rules that form the connection between

the User Model, the Domain Model and the presentation to generate.

5) *Authors* typically create the domain/adaptation model through an authoring tool.

6) the *Manager* who configures AHA!, chooses the installation directory, path names, etc., and who creates accounts for authors. The configuration is entirely done through a web based forms interface.

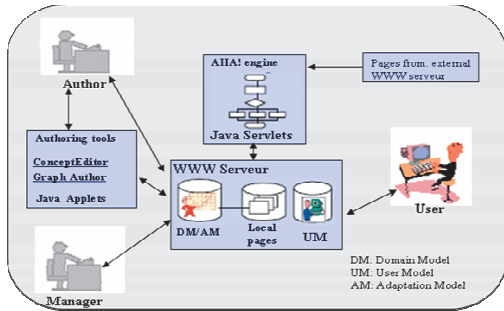


Fig. 1. Architecture of AHA!

This system is characterized by its simplicity. It carries out the adaptation of the presentation, the navigation and the content according to user's characteristics and preferences. But, since the user is not fixed in his office, it is obviously necessary to regard the user's context as a principal element for the adaptation process. This context is presented by the user's dynamic and static characteristics and the characteristics of his environment (terminal, location, session, etc.). But, as we said, AHA! does not support the user's context. For that, in our work, we will allow this system to carry out the adaptation according to user's context while trying to use the context generally.

C. Web services (WS)

A WS is an application put to provision on Internet by a provider of service allowing interoperability between the users of applications through the Web.

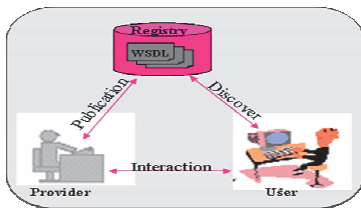


Fig. 2. Classical architecture of WS

Traditionally, the architecture of WS (Fig. 2) is composed of three entities (provider, user, and registry) and is based on three standards (*WSDL* (Web Service Description Language) [22], *UDDI* (Universal Description, Discovery and integration) [21] and *SOAP* (Simple Object Access Protocol) [16]). The service provider builds the service and publishes its description in a registry. The user needs are translated into requests that are carried on by the WS registry. Once the

service is found, the user will obtain direct interaction with the service [19].

Although the context becomes the ear and the eye of WIS and the use of WS within WIS is increasingly frequent, the classical architecture of WS does not take context adaptation into account. Indeed, some research works are interested in the possibilities of applying the context adaptation on WS as [6] and [7]. However, we can say that a generic model of the context is still absent.

In order to integrate the adaptation concept in the WIS and restore to the user a list of adapted services to his context, we will integrate the classical architecture of WS with the architecture of AHA!, adding to these architectures an adaptation layer containing various components dedicated to the context adaptation.

III. NEW VISION OF WEB INFORMATION SYSTEM

We note that the majority of proposed works in the area of adaptive WIS aim to build a WIS that is able to provide the user with relevant information to his context. However, these works are limited in analyzing small quantities of contextual information and present solutions at very specific needs. In addition, none of these works may return to the user the most adapted WS to his context.

The objective of our work is, therefore, to build an Adaptive Web Information System based on Web Services. On order to achieve this objective, we propose to extend the AHA! architecture in order to achieve a general architecture of Adaptive Web Information Systems, where this system is based on the Web Service. This system allows, on one hand, to integrate a generic context model in AHA!, and on the other hand, to deliver to the user a list of the most adapted WS to his context.

In this part, we illustrate our contribution, which aims to support the context adaptation and WS in WIS. We look, first, our architecture that combines the two architectures AHA! and the classical architecture of WS detailing the role of each component. Then, we show how the context is defined and represented in our architecture. Next, we detail the adaptation process. Finally, we will show our work through an example.

A. Our architecture

In our research, we are interested in the architecture of a Distributed Information System providing a list of adapted WS to user context. We will extend the architecture of AHA! combining it with the classical architecture of WS and adding to these architectures an adaptation layer containing various components dedicated to context adaptation.

The next figure illustrates our proposal. In this architecture, the role of some components of AHA! does not change. For each component of the architecture AHA! we define a new component that supports its role in our architecture.

The *user* does not change. But in our architecture, the user can launch his request using not only a PC but can use another device types such as a mobile device or PDA.

The User Model (UM) is replaced, in our architecture, by

the *User Context*. This context includes information from the user's environment, which may influence the adaptation process.

The Domain Model (DM) is replaced, in our architecture, by the *Web Services context*. This context includes information on the environment service, for example, with what kind of terminal we can display services.

The author plays the role of *service provider* that builds and publishes his description in the Registry and which will be responsible for providing the context of each WS and stores its in DM.

The role of (AM) is played by the *Answer Generator* which manages the adaptation process. This process aims to deliver to the user a list of the most adapted WS to his context.

The local and external pages is replaced by the *WS* provided by several *WIS*.

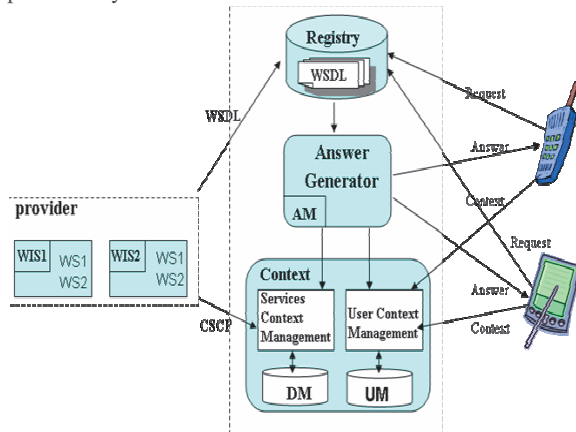


Fig. 3. Architecture of WIS with context and WS

So our architecture is composed of three layers depending on each other (Fig. 3):

1) *The Registry layer* that corresponds to a registry of service description offering facilities that publish services for providers and that searches services for users. The provider was presented by the *WIS* where each *WIS* has a *WS*. The user will launch his request to the Registry; a list of adapted services to his request will be forwarded to the second layer.

2) *The Answer Generator layer* is responsible for managing the adaptation process and generates the final answer (the list of *WS* adapted to the user context) to the user.

3) *The Context layer* is responsible for capturing and managing of user and services context.

The *Fundamental layer* in our architecture is the Answer Generator which is responsible for managing the adaptation process. It is capable of:

- Receiving a list of adapted services to the user request of the first layer.
- Asking the third layer, the user context and services context returned in the list of the first layer.
- Comparing the two contexts (user and services).
- Delivering to the user the list of the most adapted services

to his context.

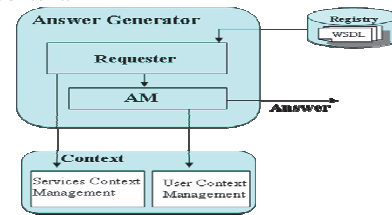


Fig. 4. Answer Generator

This layer is composed of two components (Fig. 4).

1) The first "*Requester*" asks the Registry the list of adapted services to the user request. Then he contacted the Services Context Management for request services contexts found in the list returned to the Registry.

2) The second "*Adaptation Model*" (AM) is responsible for the adaptation process. AM performs a process of "matching" between the user context and the context of each service by comparing the attributes of their metadata. The list of *WS* returned to the user will be sorted according to the degree of similarity calculated in the matching.

To restore the list of the most adapted services to context, we need to define the necessary means to capture and store the user's context and the services context. The Context layer is responsible for managing this process. It is composed of module of the User Context Management and the Services Context Management.

1) The first one called "*User Context Management*" is responsible for capturing the user context and storing it in a database (UM). It is broken down into the three following stages (Fig. 5).

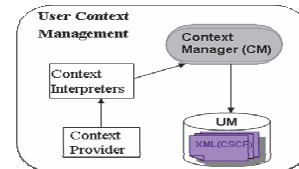


Fig. 5. User Context Management

▪ *Capturing User Context* is carried out using physical sensors which generate raw data that can't be directly exploitable by the application. In our architecture we defined the entity "Context Provider" that represents a context capturing system.

▪ *Interpreting Context*: this step aims to transform raw contextual data into meaningful information easily used by the application. In our architecture we define the "Context Interpreters" to represent a context interpretation system.

▪ *Context Storage*: the Context Manager element CM stores implicit and explicit contextual information in an XML database [23].

The user context is obtained in an explicit and implicit way so when the user launches his request, he can express explicitly his static characteristics and his preferences but remaining characteristics (localization, network, etc) are obtained implicitly by the context sensor as we show next.

2) The second called “*Services Context Management*” is responsible for the extraction of contexts Web Services, the storage of these contexts in a database (Domain Model (DM)), and restitution to The Answer Generator contexts of services that meet the needs of the user (Fig. 6).

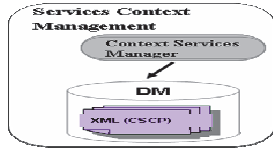


Fig. 6. Services Context management

In this component, the service context is expressed directly by the service provider.

B. The context representation

In this field, researchers did not agree yet on a definition of the context concept which is generic and pragmatic. Indeed, proposed definitions are very specific to a particular domain. These definitions make the formalization of context difficult due to a lack of precision or of generality. The definition of Dey is the most widespread and most accepted by the majority of researchers. But Winograd [20] and Chaari consider that this definition may be a source of conflict because it does not help in separating the contextual data from the application data. To provide more precision in Dey’s definition, Chaari proposed the following definition: the set of the external parameters that can influence the behavior of the application by defining new views on its data and its available services. These parameters may be dynamic and may change during the execution.

We think that this definition is more useful for the context exploitation in applications. Therefore we will use this definition to define four principal elements of the context: the *user characteristics* which are composed of *static characteristics* (langue, first name, etc) and of *evolutive characteristics* defined by his environment (his localization, time, etc) and his preferences, the *device characteristics* (material, software), the *network characteristics* (type, bandwidth) and the *session characteristics* (date, hour, state, etc).

In order to return to the user the list of the most relevant services to his context, we must take into account not only the user context but also the services context. In our architecture, we define the service context by the four elements identified before to the user.

To represent the contextual information, the majority of researchers used the format CC/PP [9]. CC/PP is a decomposable, uniform and extensible standard but we did not use it to represent our context because of its lack of structured functionalities where its two leveled strict hierarchy is not appropriate to capture structures of complex profiles [1].

We use the CSCP format to overcome the structure deficits of CC/PP. CSCP provide a multileveled structure which allows the representation of all the types of contextual information.

So, we stock all the contextual information in an XML document based on this CSCP model (Fig. 8). Each element of this document presents a context element.

Fig. 7 presents the general structure of our context model. For each user session we associate a profile "Context-Profile" which describes the context "Context-Element": the user profile, the session profile, the terminal profile and the network profile. Each "Context-Element" may contain a "Context-Sub-Element". Each "Context-Sub-Element" may contain another "Context-Sub-Element" or a "Context-Attribute". The contextual situation is defined by the values associated with Context-Sub-Element of the context.

The modification of one of these values corresponds to a transition to another contextual situation. Each Sub-Element of the context can be *static* if it does not change in a user session or *dynamic* if its value can vary in the same user session. That's why we associate for each Sub-element, an attribute "Type" to differentiate dynamic elements from static ones. This differentiation is very important for the adaptation process. Indeed, the application must be notified immediately when changing content of a dynamic component while the static elements are consulted when necessary.

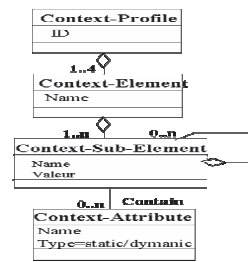


Fig.7. General structure of our context model

C. The adaptation process

As we mentioned in our architecture, the Answer Generator is responsible for managing the adaptation process. This process aims to restore the user the list of the most adapted services to context.

To achieve the adaptation process, AM performs a matching process between the two contexts (user and services) by comparing the contents of the context elements. In order to carry out this process we use a mechanism provided by the CSCP format. This mechanism allows attaching priority to the attributes. When the user launches his request, it must express explicitly not only his preferences but also priorities for each attribute of his preferences (and his context). We associate for each preference an attribute "weight" in order to be able to prioritize the criteria for adaptation. The user can assign each element a real number between 0 and 1: 1 being the criteria for adaptation priority and 0 as the default, designating a criterion of adaptation without special hierarchy.

So, AM compares current user context (D_i) with the context of each service rendered in the adapted list to the needs of the user ($D_j, j = 1 \dots n$) calculating the degree of similarity (DS). The list of Web Services returned to the user by the Answer

Generator will be sorted according to the degree of similarity which was calculated in the matching. So the most appropriate Web Service will be found at the top of the list

D. An example

A user wants to travel from Paris to Bordeaux. He wants to know price of the train ticket going from the nearest station to his location. The user also prefers that the service would answer in English and be able to adapt to his device.

The user will launch his request to the Registry asking a service for reserving the cheapest train ticket from Paris to Bordeaux. At that time, User Context Manager captures explicitly his preferences via an interface that allows him to express his preferences and the priorities associated with them. The service will answer the user request in English and would provide him with the nearest station of his departure. The user precise the weight (0.3) for the first attribute "Language" and the weight (0.7) for the second attribute "place". The User Context Manager captures also the information concerning the device type and the network implicitly. Then, it stores all this information in an XML document in the database UM (Fig.8).

```

<?xml:version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cscp="http://context-aware.org/CSCP/CSCPProfileSyntax#"
...
<ContextProfile rdf:ID="Context">
<User>
<use :UserProfile>
<use :CharStatic>
<use :Languages>French</use :Languages>...
</use :CharStatic>
<use :CharDynamic>
<use :Environment>
<use :Location>Near the care station </use :Location>
<use :Time>8 h</use :Time>
</use :Environment>
<use :Preference>
<use :Language>
<ref:Bag>
<use:li weight="0.3" > English</use:li>
</ref:Bag>
</use :Language>
</use :Preference>
</use :CharDynamic>
</use :UserProfile>
</User>
...
</ContextProfile >
</rdf:RDF>

```

Fig.8. An example of CSCP profile

In order to make the filtering process the Answer Generator calculates a score of correspondence between the descriptions of user context and Services Context by taking into account the priorities expressed by the user. In this example, the Generator gives priority to the attribute place because it has a higher weight. The list of WS returned to the user by the Answer Generator will be sorted according to the degree of similarity calculated in the matching. Where the most adapted WS will be placed at the top of the list of services.

IV. CONCLUSION

Nowadays, with the great diffusion of mobile technology, ubiquitous systems, and distributed systems, the context becomes the ear and the eye of the information system.

In this article, we proposed a simple and general architecture that allows, on one hand, to support the adaptation process according to the user context and on the other hand, to return to the user a list of adapted Web Services to his context.

So, our aim is to build an Adaptive Web Information System based on Web Services where the user can reach the list of the most relevant service for his request user and his Context.

REFERENCES

- [1] A. Held, S. Bouchholz, and A. Shill, "Modeling of Context for Pervasive Computing Application," In Proceedings of the 16th world Multiconference on Systemics, Cybernetics and Informatics (SCI). Orlando, FL, USA, 2002.
- [2] A. Jameson, "Modeling both the context and the user," Personal and Ubiquitous Computing Journal, 2001, pp. 29-33.
- [3] A.K. Dey, "Providing Architectural Support for Building Context-Aware Applications," PhD Thesis, Georgia Institute of Technology, 2000.
- [4] B.N. Schilit, and M.M. Theimer, "Disseminating active map information to mobile hosts," in IEEE Network, vol. 8, n°5, 1994, pp. 22-32.
- [5] B.N. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications," Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, USA, December 1994, pp. 85-90.
- [6] B. Soukkarieh, and F. Sèdes, "Integrating a Context Model in Web Services," in : IEEE International Conference on Web Services (ICWS 2007), Utah, USA, 09/07/07-13/07/07, Juliet 2007, pp. 1195-1196.
- [7] B. Soukkarieh, and F. Sèdes, "TOWARDS CONTEXT-AWARE WEB SERVICES," in IADIS International Conference e-Society (e-Society 2007), Lisbon, Portugal, 03/07/07-06/07/07, Juliet 2007.
- [8] C. Cappiello, M. Comuzzi, E. Mussi, and B. Pernici, "Context Management for Adaptive Information Systems, " Electr. Notes Theor. Comput. Sci. 146(1): (2006), pp. 69-84.
- [9] CC/PP, W3C, Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation 15 January 2004. The last version: <http://www.w3.org/TR/CCPP-struct-vocab/> (Janvier 2007).
- [10] H.K. Rubinsztein, M.Endler, V. Sacramento, K. Gonçalves, and F. Nascimento, "Support for context-aware collaboration," in First International Workshop on Mobility Aware Technologies and Applications - MATA 2004, Florianopolis, Brazil, Springer-Verlag, 2004, pp.37-47.
- [11] K. Cheverest, K. Mitchell, and N. Davies, "The role of adaptive hypermedia in a context-aware tourist guide," in Communication of ACM, vol. 45, n° 5, Mars 2002, ACM Press, pp. 47-51.
- [12] K. Mostéfaoui, J. Pasquier-Rocha, and P. Brézillon, "Context-aware computing: a guide for the pervasive computing community," Proceedings of the IEEE/ACS International Conference on Pervasive Services (IPCS'04), IEEE Computer Society, 2004, pp. 39-48.
- [13] M. Villanova-Oliver, "Adaptabilité dans les systèmes d'information sur le web : Modélisation et mise en oeuvre de l'accès progressif," PHD, Institut National Polytechnique de Grenoble, 18 Décembre 2002.
- [14] P. De Bra, G. J. Houben, and H. Wu, "Aham : A dexter-based reference model for adaptive hypermedia," in tenth ACM conference on hypertext and hypermedia, 1999, pp. 147-156.
- [15] P. De Bra, A. Aerts, B. Berden, B. De Lange, B. Rousseau, T. Santic, D. Smits, and N. Stash, "AHA ! The Adaptive Hypermedia Architecture," HT'03, Nottingham, United Kingdom, August 26-30, 2003, pp 81-84.
- [16] SOAP, W3C, Recommendation W3C 24 Juin 2003. The last version: <http://www.w3.org/TR/soap12-part0/> (December 2006).
- [17] T. Chaari, F. Laforest, and A. Flory, "Adaptation des applications au contexte en utilisant les services Web, " Second Francophone workshop: Mobility and Ubiquity, Grenoble, France, 31 May- 3 June 2005.
- [18] T. Lemlouma, "Architecture de négociation et d'adaptation de Services Multimédia dans des Environnements Hétérogènes," PHD, Institut National Polytechnique de Grenoble, Grenoble, France, April 2004.
- [19] T. Melliti, "Interopérabilité des Services Web complexes. Application aux systems multi-agents," PHD, Paris IX Dauphine University, 2004.
- [20] T. Winograd, "Architectures for Context," in Human-Computer Interaction Journal, 2001, Vol. 6, N°2-3, pp. 401-419.
- [21] UDDI, W3Schools, http://www.w3schools.com/wsd/wsd1_uddi.asp (December 2006).
- [22] WSDL, W3C, Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001, the last version: <http://www.w3.org/TR/wSDL>. (December 2006).
- [23] XML, W3C, XML Base W3C Recommendation 27 Juin 2001, the last version : <http://www.w3.org/TR/xmlbase> (December 2006).