User's Profile Representation for Distributed Systems

Youcef SKLAB[a], Hassina Nacer[b], A.Kamel Tari[c]

[a]LIMED, Laboratory, Computer Science Department, University A/Mira Bejaia
ysklab@hotmail.com
[b]LIMED, Laboratory, Computer Science Department, University A/Mira Bejaia
sino nacer@yahoo.fr
[c]LIMED, Laboratory, Computer Science Department, University A/Mira Bejaia
tarikamel59@gmail.com

Abstract. Nowadays, computing systems are used and integrated in different ways into our daily life. In pervasive environments, they give users the ability to get access to the information anywhere at any time, which leads to a dynamic evolution in the user context. However, those systems should be aware of this evolution and adapt their behavior accordingly, to meet the user needs and preferences. In this paper, we present a hybrid user context model for the pervasive environment following a layered architecture. Each layer describes one functional aspect of context management. We combine between graphical representations, logic expressions and Semantic Web technologies to describe and search for user context. Our model enables model specializations to particular domains and context sharing with external applications. Finally, this paper shows specific prototypes that offer personalized and context-aware information to the user.

Keywords: User Profile, User Context, Knowledge Representation.

1 INTRODUCTION

In recent years, people are increasingly using information systems with a world wide access. Nowadays, in pervasive environments, several Web-enable devices have been massively introduced, such as laptops, PDAs and smartphones. Users have been able to get access to both local and remote resources everywhere at any time and may accomplish several of their tasks using a large variety of devices. However, these new capabilities lead to a highly dynamic changes in the circumstances on which those users are using these systems. Users may have different profiles, with different needs, interests, preferences and activities.

Computing systems in pervasive environments are facing several new challenges on identifying and representing the user context. They may be used by different users in different conditions and/or for different purposes. So, they should be aware of the user context to grant a correct interpretation of his interactions and adapt their behaviors accordingly. Many efforts to consider the user context in several domains and for several purposes were made in the past years. In the literature, some works aimed to propose a context model and others has just integrated some part of it, without following a specific context model. However, in all these efforts and whatever the purpose of considering context, there is a lack of a general model to guide the context-aware systems design. The model must offer the ability to cover large user information and the possibility to instantiate it in various domains.

In this paper, we present a layered user context model, using three context modeling approaches. We use: 1- a graphical representation by UML, increasing the expressiveness of the model. 2- Logic predicate and inference rules, enhancing its formality with formal expressions. 3- And ontologies, to grant semantic description of the user context. Our context model has great advantages in terms of expressiveness and context semantic sharing, which allows to have common interpretation of context information among different context-aware systems. The model considers different functional aspects of context management in a layered architecture, which reduces the complexity of context handling. Furthermore, it enhances the degree of independence between context and its sources and increase the system flexibility.

The rest of this paper is organized as follows: Section 2 presents a short related work in the field of context modeling. Section 3 presents our user context model in pervasive environments, a prototype implementation and a comparison with other works. Finally, Section 4 concludes the article.

2 USER CONTEXT MODELING

Context modeling is a fundamental step for the development of context-aware systems. In this section, we present some user context models in different areas, adopting the major classifications of context modeling approaches (Vanathi & Uthariaraj, 2011).

2.1 Key-Value-Pairs modeling

Earlier context consideration (Dey, Salber, & Abowd, 2001), (Schilit, Adams, & Want, 1994) was by providing the value of a context attribute to an application as an environment variable. Key-Value pairs are the simplest data structures associating context attributes with specific values of contextual information. They are easy to manipulate but not convenient for complicated and sophisticated structuring purposes.

2.2 Markup scheme modeling

The portability of markup tags languages is a key feature when considering context in systems which are distributed and use hybrid technologies. Mark-up scheme models (Mrissa, 2007) provide only syntactic description of context and they are not convenient to describe complex contexts. Furthermore, usually, they are specific to a particular field and/or limited to some context aspects (localization, environment...).

2.3 Graphical modeling

Expressiveness and well structuring are suitable features for context modeling. Several authors (Behlouli, 2006), (Ben-Hamida, 2010), (Chaari, 2007), (Kostadinov, 2008), (Sheng & Benatallah, 2005) have used UML (Unified Modeling Language) to propose context models, because of it provides descriptive and generic structures paired with a strong graphical representation. Furthermore, UML is suitable to have multidimensional representations of context (Kostadinov, 2008) or a hierarchical representation (Ben-Hamida, 2010). Another graphical model is CML (*Context Modeling Language*) (Henricksen, Indulska, & McFadden, 2005), which is proposed as an extension of ORM (*Object Role Modeling*). Graphical models are simple but less formal than the others and, in general, not used on instance level. Moreover, most of the proposed models are limited to a graphical representation and do not handle other aspects of context management.

2.4 Object oriented modeling

There have been also some context models with promoting re-usability and controlled access to contextual information. This is done by adopting the Object oriented modeling approach (Hofer & Schwinger, 2003), (Kirsch, Gensel, & Martin, 2004) which consists of encapsulating contextual information into objects. The information can be reached only through well-defined interfaces, and so, hidden to and from other objects. Object oriented models have a high level of formality because of well-defined interfaces. They simplify their integration within the applications, but they are not adapted for context sharing in open and dynamic environments.

Table 1. User Context Elements

Domains	Works	Context Elements																
		Entity								Activity		Environment		Time			Location	
		Physical						Virtual										
		User			Device													
		Personal Data	Interests Domain	Preferences	Hardware	Software	Physical object	Service	Application	Task	Actions	Physical	Virtual	Absolute value	Duration	Symbolic	Geometric	Symbolic
Pervasive Environments	K. Henricksen, 2005		X		X					X					X		X	
	T. Chaari, 2007		X		X	X	X		X			X	X	X	X		X	X
	A. Ben-Hamida, 2010			X	X	X		X	X						X		X	
	M. Miraoui, 2009				X	X		X				X	X	X		X		X
	D. Cho et al, 2008		X		X					X					X		X	X
Mobile Environments	T. Hofer et al, 2003	X			X									X			X	
	N. Behlouli, 2006			X				X		X		X		X	X		X	X
	M. Kirsch-Pinheiro et al, 2004			X			X	X		X		X	X		X			X
Information Personalization	D. Kostadinov, 2008	X	X	X	X	X						X		X		X	X	X
	K. Abbas,2008	X		X	X		X			X	X	X		X	X		X	X
	R. Hervas et al, 2010	X	X		X			X		X		X		X	X		X	X
Service-oriented Environments	Q. Sheng et al, 2005							X				X	X		X			X
	M. Missa, 2007							X					X					
	S. YU, 2008				X		X	X		X			X	X	X			X

2.5 Logic based modeling

Supporting reasoning about context to deduce new context information or to initiate reactions by applications, is highly recommended when considering context in dynamic environments. The authors in (Bao, Tao, & Deborah, 2010), (Ranganathan & Campbell, 2002), use logic expressions to define conditions on which a concluding expressions may be derived from a set of other expressions. Logic-based models represent a highly formal modeling approach. But they are often based on a centralized context management, a solution which is not convenient for context distribution, and they have low expressiveness.

2.6 Ontology modeling

Semantic description is a strong feature when context is handled. Ontological models use ontologies as a uniform way for specifying their core concepts as well as sub-concepts and facts (Cho & Hong, 2008), (Hervàs, Bravo, & Fontecha, 2010), (Miraoui, 2009). They offer clear advantages regarding support for interoperability, semantic description and context sharing with common interpretation. However, when considering the trade-off between expressiveness and complexity the choice of ontological models may not always be satisfactory. Furthermore, ontological models alone are generally not well suited to recognize and to represent some simpler and/or dynamic context data (Bettini & Brdiczka, 2010). They are not designed for capturing and processing constantly changing information in dynamic environments in a scalable way, such as user's dynamic interests.

The notion of context is modeled in different ways and for different areas and purposes. Several classifications of context information were proposed, most of them made a classification in several categories (Razzaque, Dobson, & Nixon, 2005), (Rosemann, Recker, & Flender, 2008) others have made a classification in two categories only (Behlouli, 2006), (Henricksen, Indulska, & McFadden, 2005), (Sheng & Benatallah, 2005). The Table (*Table 1*) shows how the most common categories were considered in different areas.

In pervasive environments, a user context model should consider the heterogeneity of context information's and their sources. It should support several acquisition techniques to map easily the dynamic changes of context information from real world concepts into modeling constructs. And it should offer a good balance between expressive power and formal representation with efficient reasoning about high-level context abstraction expressed as a form of predicate logic. Moreover, it should allow semantic context sharing among applications, to enhance richness and interoperability. According to the above considerations, none of the viewed approaches can satisfy all the requirements needed for context modeling in such environments. So, it is better to integrate different modeling approaches and reasoning tools with each other, to have a generic and inter-operable model.

2 USER CONTEXT MODEL

Pervasive environments cover a lot of application domains, like medical domain, e-business domain or military domain. In this section we present our user context model following a layered architecture.

3.1 Architecture overview

The layered architecture is one of the easiest architectures to manage and implement. It promotes re-usability and independence. Considering these advantages, we propose a formal layered user context model (Fig. 1), to guide the development of context-aware systems. We consider several functional aspects of context management and melds them into a coherent architecture. Each layer describes one aspect, starting by: Specifying and identifying the context sources, specifying the different capturing methods and mechanisms, interpreting and abstracting events and user interactions to usable context information, inferring new context information, syntactic and semantic description of user context situations, then storing the context description and sharing it as a Web Service.

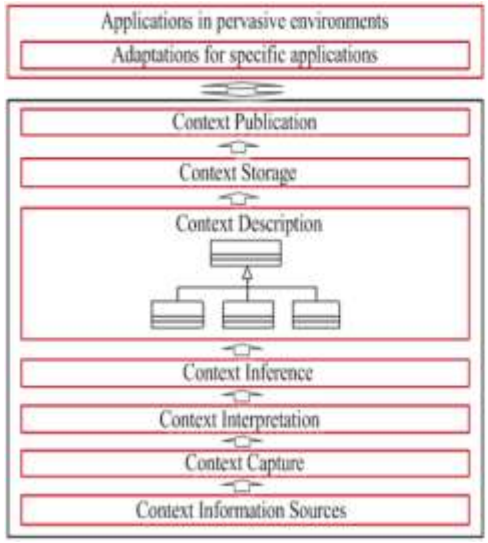

Fig.1. Architecture of the proposed model.

In addition to the layers suggested in (Baldauf & Dustdar, 2007), (Chaari, 2007), our model provides two more layers which aims to support semantic description of the user context and it's sharing between applications.

3.1.1 Context information sources layer

In pervasive environments, context information could be gathered from a variety of sources that differ in the quality of information they provide. Therefore, it is important to know from which source the information is obtained, since they do not all have the same level of accuracy or credibility. Context information sources layer aims to specify and manage the set of entities that provide initial information. This will enable applications to choose which information to consider according to their sources.

3.1.2 Context capture layer

This layer aims to abstract the acquisition mode independently from the context information. Due to the dynamic of pervasive environments, it is convenient to consider different techniques for context information acquisition. Because, one technique that can be effective in a user environment may be less in other environments. At this layer we specify two types of acquisition modes, explicit acquisition and implicit acquisition. Context information is contained in a basic structure called *Context Attribute*, which can be a simple attribute or a complex attribute, composed of several simple attributes.

3.1.3 Context interpretation layer

This layer aims to abstract sensor's events and/or user's interactions. It consists on translating them into a context information in a format usable by applications. To do so, we use interpretation functions to associate for a set of events or users' actions with interpretable context attributes values.

3.1.4 Context inference layer

Since there are static and dynamic conditions in the context of real world, we need inference mechanism to deduce context information according to the dynamic changes in the user situations. Some information can only be inferred by analyzing user's activities over time. The context inference layer leverages propositional inference rules to deduce implicit context information from related explicit or implicit context.

3.1.5 Context description layer

After acquiring the set of user context information's, we use a UML model (Fig. 2) to represent the user context description. This description is supported by formal expressions and associated with a semantic description using ontologies. We associate each user context description with an ontology to describe the semantic of its concepts.

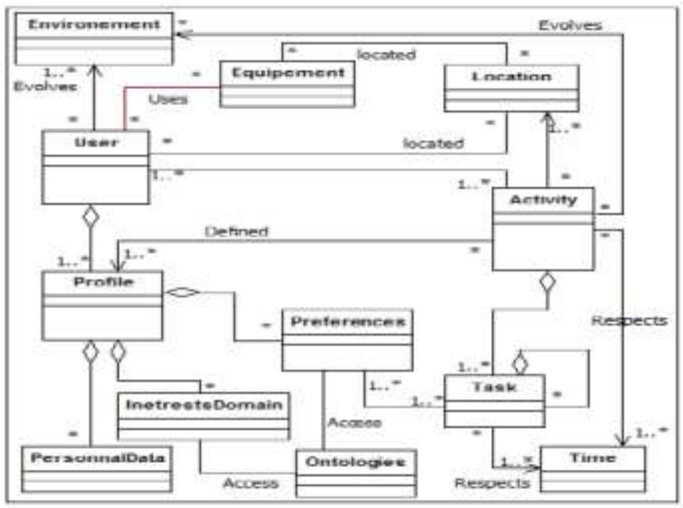

Fig.2. User context Meta-Model

A user context description covers information related to the following main elements: The user, which is described by its profile (Personal data, Preferences, Domain of interest), the activities performed by the user, its environment and devices, time and location.

3.1.6 Context Storage layer

Once we obtained the user context description from the description layer, this layer aims to store it in a semantically annotated XML documents.

3.1.7 Context publication layer

To allow other applications to use obtained context description, we share it by using the Web services technologies. Using Web services, allows to unify and standardize the access to user context information.

3.1.8 Application Layer

Applications which use the context provided by the underneath layers. It has a sub-layer (*Adaptation Layer*) which aims to specify the required adaptations (if needed, like the implementation of context interpretations functions) when using the model in different applications. This layer is not detailed, as it is considered as a part of each application.

3.2 Implementation

The figure (Fig. 3), shows the global architecture we have used to implement our model. We specify two kinds of components: specific and common components. Specific components (Like *Application*: to contain the application treatments, *Local database*, *Local Ontology*) are specific to each application. Common components are common among all applications. This category includes *StorageManagment*: to manage databases, ontologies and XML files, *ContextManagment*: which is the implementation of the proposed model, and *Global Ontology*.

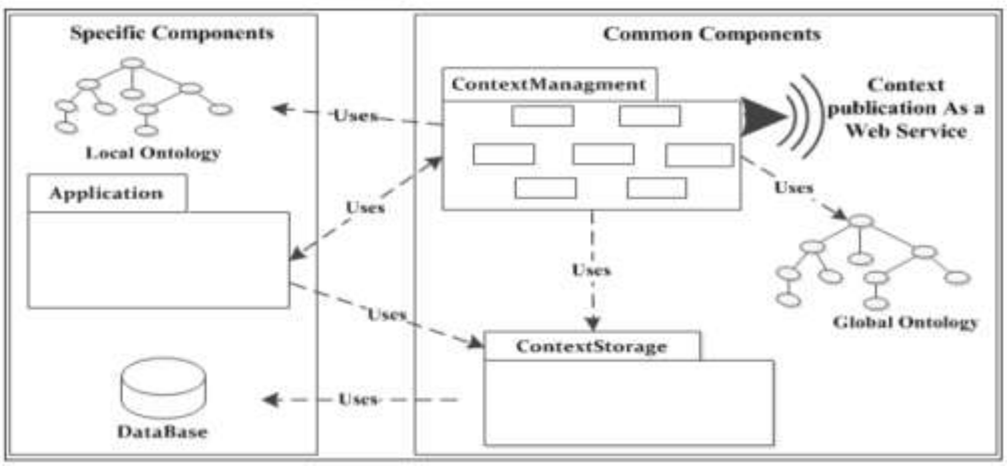

Fig.3. Architecture implementation overview

Instantiating the meta-model (Fig.2) can lead to different specific models with different context elements for each domain. To store the user context we use an XML documents meta-model (Fig. 4). It offers a generic structure which can be instantiated and used to contain the user context instances for different applications. It has the following classes: *contextElement*: which represents a user context element. *profilDimension*: which represents a user profile dimension. *contextCategory*: which allows to have a hierarchical representation, like a hierarchical user profile dimensions and sub-dimensions or activities and their sub-activities. *contextAttribut* which represents a user context attribute. Each one is described by a set of attributes: *name* for the attribute name, *mode* for its acquisition mode, *source* for its source, *structure* for its structure and *uri* for the link to the used ontology for its semantic description.

3.2.1 Instantiation of the proposed model

We have instantiated our user context model in two different domains in pervasive environments. The first is the medical domain, where the challenge is how to keep a clear picture of what happens during the presence of a patient in a hospital to allow optimum use of available resources (human and / or material) and to better take care of patients and their health. The second is the e-business domain, where the challenge is how to get maximum information about the context of customers and how to detect and identify their needs.

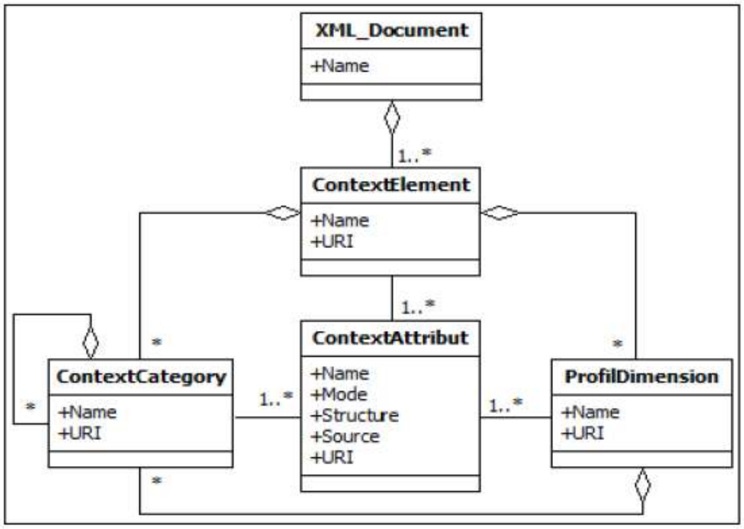

Fig. 4. XML documents Meta Model.

The first prototype is an application for monitoring patients in a hospital. It is used by physicians and / or nurses to record patient's data, like new observations on visits and prescribed treatments and their follow-up (Fig. 5). The application detects and captures patients who form the interests domain of physicians and according to their preferences and activities, it send them notifications about new prescribed treatments (by other doctors). The second prototype is a website for online selling. This application uses the user context to better respond to his requests about products. The used user context description is that shared and obtained from the first prototype. The figure (Fig. 6) shows an excerpt of the obtained document after identifying matches between the used concepts in published (by first prototype) and local context document (second prototype).

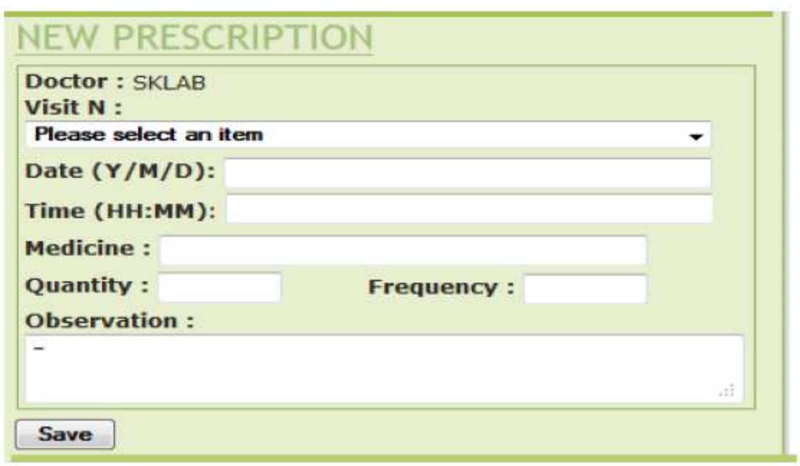

Fig. 5. Creating a new treatment interface.

```xml
<contextElement name="user" uri="http://www.LocaleOnto.com/GlobalOnto.owl#user">
<profilDimension name="PersonalData" uri="http://www.LocalOnto.com/LocalOnto.owl#PersonalData">
    <contextAttribut mode="profiled" name="function" source="U17" structure="String"
        uri="http://www.LocaleOnto.com/LocalOnto.owl#function">doctor</contextAttribut>
    <contextAttribut mode="profiled" name="speciality" source="U17" structure="String"
        uri="http://www.LocaleOnto.com/LocalOnto.owl#speciality">cardiology</contextAttribut>
    <contextAttribut mode="profiled" name="e-mail" source="U17" structure="String"
        uri="http://www.LocaleOnto.com/LocalOnto.owl#email">ysklab@hotmail.com</contextAttribut>
    ...
</profilDimension>
<profilDimension name="preferences" uri="http://www.LocaleOnto.com/LocalOnto.owl#preferences">
    <contextCategory name="Display" uri="http://www.LocaleOnto.com/LocalOnto.owl#display">
        <contextAttribut mode="profiled" name="language" operator="=" source="U17" structure="String"
          uri="http://www.LocaleOnto.com/LocalOnto.owl#language">FR</contextAttribut>
    </contextCategory>
</profilDimension>
<profilDimension name="espaceInterets" uri="http://www.LocaleOnto.com/LocalOnto.owl#espaceInterets">
    <contextAttribut mode="captured" name="Patient 10" source="Application" structure="int"
        uri="">12</contextAttribut>
    ...
</profilDimension>
</contextElement>
<contextElement name="activity" uri="http://www.LocaleOnto.com/LocalOnto.owl#activity">
    <contextAttribut mode="infered" name="NameTask" source="Application" structure="String"
    uri="http://www.LocaleOnto.com/LocalOnto.owl#NameTask">ClinicalExam</contextAttribut>
    <contextAttribut mode="static" name="StartDate" source="Application" structure="DateTime"
     uri="http://www.LocaleOnto.com/LocalOnto.owl#dateDebut">12/08/2012/08:00</contextAttribut>
    ...
```

Fig.6. Extract of the shared user context.

3.3 Discussion

We proposed a user context model applicable in different application domains in pervasive environments, unlike works like (Hervàs, Bravo, & Fontecha, 2010), (Mrissa, 2007, (Ranganathan & Campbell, 2002). Our model is supported by its hybridization, integrating several modeling approaches and benefiting from their advantages such as expressiveness, abstraction, formality and semantic richness. It considers different functional aspects of context management, unlike works (Bao, Tao, & Deborah, 2010), (Hervàs, Bravo, & Fontecha, 2010), (Kirsch, Gensel, & Martin, 2004) which are limited only to the representation aspect. Furthermore, it support semantic description of the user context, like (Cho & Hong, 2008), (Hervàs, Bravo, & Fontecha, 2010), (Miraoui, 2009), (YU, 2008). This is enhanced by a context sharing mechanism, which is not supported in works like (Bao, Tao, & Deborah, 2010), (Behlouli, 2006), (Ben-Hamida, 2010), (Henricksen, Indulska, & McFadden, 2005), (Hofer & Schwinger, 2003), (Kirsch, Gensel, & Martin, 2004), (Schilit, Adams, & Want, 1994), (Sheng & Benatallah, 2005), allowing to share and use context descriptions that are already available on other applications.

Alongside these positive features, our model has some limitations. Like considering context information quality and the credibility of their sources, which will allow applications to better choose which information to consider (Henricksen, Indulska, & McFadden, 2005). Other limitation is modeling relationships among contextual situations (Bao, Tao, & Deborah, 2010).

4 CONCLUSION

Considering context by computing systems promises to offer a better future for everyday computation. The application areas are variable, where users can interact with each other and traditional systems using their hand-held devices. In this paper, context is managed in a layered architecture and represented in a hybrid model using UML, Predicate logic and ontologies with supporting sharing by means of Web services technology. We developed a prototype of our model and instantiated it in two different domains.

References

ABBAS, K. (December 2008). *Système d'accès personnalisés à l'information: application au domaine médical*. PhD thesis, Institut National des sciences Appliquées de Lyon, France.

Baldauf, M. & Dustdar, S. (2007). A survey on context-aware system. *International Journal of Ad Hoc and Ubiquitous Computing*, 2:263–277.

Bao, J., Tao, J. & Deborah, L. (April 2010). Context representation for the semantic web. In *Web Science Conference*, Raleigh, North Carolina, USA.

Behlouli, N. B. (November 2006). *Ajout de mécanismes de réactivité au contexte dans les intergiciels pour composants dans le cadre d'utilisateurs nomades*. PhD thesis, Institut National des Télécommunications, France.

Ben-Hamida, A. (February 2010). *AxSel : un intergiciel pour le déploiement contextuel et autonome de services dans les environnements pervasifs*. PhD thesis, L'institut national des sciences appliquées (INSA) (Université de Lyon), France.

Bettini, C., Brdiczka, O., & K. H. et al (April 2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6:161–180.

Chaari, T. (September 2007). *Adaptation d'applications pervasives dans des environnements multi-contextes*. PhD thesis, Laboratoire d'Informatique en Image et Systèmes d'information (LIRIS).

Cho, D. J., & Hong, M. W. (2008). A design of ontology context model in ubiquitous learning environments. In *Proceedings of the 12th WSEAS international conference on Computers*, pages 844–848, Wisconsin, USA.

Dey, A. K. & Salber, D., & Abowd, G. D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2-4):97–166.

Henricksen, K., Indulska, J., & McFadden, T. (2005). Modelling context information with ORM. In *Proceedings of the OTM Confederated international conference on the Move to Meaningful Internet Systems*, pages 626–635, Berlin, Germany.

Hervàs, R., Bravo, J., & Fontecha, J. (2010). A context model based on ontological languages: a proposal for information visualization. *Journal of Universal Computer Science*, 16:1539–1555.

Hofer, T., Schwinger, W., & M. P. et al (January 2003). Context-awareness on mobile devices - the hydrogen approach. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, volume 09, pages292–302, Hawaii, USA.

Kirsch-Pinheiro, M., Gensel, J., & Martin, H. (2004). Representing context for an adaptative awareness mechanism. In *Proceedings of the International Workshop on Groupware: Design, Implementation and Use*, volume3198, pages 339–348, San Carlos, Costa Rica.

Kostadinov, D. (September 2008). *Personnalisation de l'information : une approche de gestion de profils et de reformulation de requetes*. PhD thesis, L'université de Versailles Saint-Quentin-en-Yvelines, France.

Miraoui, M. (July 2009). *Architecture logicielle pour l'informatique diffuse : Modélisation du contexte et adaptation dynamique des services.* PhD thesis, Ecole de technologies supérieure Université du Québec, Canada.

Mrissa, M. (November 2007). *Médiation Sémantique Orientée Contexte pour la Composition de Services Web*. PhD thesis, Université Claude Bernard Lyon 1, France.

Ranganathan, Z., Campbell, R., & A. R. et al (2002). Conchat: A context-aware chat program. *IEEE Pervasive Computing*, 1(3):51–57.

Razzaque, M., Dobson, S., & Nixon, P. (July 2005). Categorization and modeling of quality in context information. In *Proceedings of the IJCAI Workshop on AI and Autonomic Communications*, Edinburgh, Scotland.

Rosemann, M., Recker, J., & Flender, C. (2008). Contextualisation of business processes. *International Journal of Business Process Integration and Management*, 3:47–60.

Schilit, B. N., Adams, N., & Want, R. (1994). Context-aware computing applications. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, pages 85–90, Santa Cruz, CA, US.

Sheng, Q. Z., & Benatallah, B. (2005). Contextuml: A uml-based modeling language for model-driven development of context-aware web services. In *Proceedings of the International Conference on Mobile Business*, pages206–212, Washington, USA.

Vanathi, B. V. & Uthariaraj, R. (2011). Context aware system for smart consignment tracking. *European Journal of Scientific Research*, 50:586–596.

YU, S. (February 2008). *Contextualized and Personalized Location-based Services*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, France.