iML: extension of UML for designing Context-Aware Services

Mostapha Bouterfass[1], Abdelhaq Errami[2]and Alami Semma[3]

[1] Department of Mathematics and Computer Science, Faculty of Science and Techniques, Hassan 1 University, Settat, Morocco

[2] Department of Mathematics and Computer Science, Faculty of Science and Techniques, Hassan 1 University, Settat, Morocco

[3] Department of Mathematics and Computer Science, Faculty of Science and Techniques, Hassan 1 University, Settat, Morocco

Abstract

In the last decade, the IT industry has been oriented and deeply influenced by the prospecting of both mobile and ubiquitous computing trends. PDAs, smart phones and other devices have already freed us from the technological shackles of workstations and desktops.

Since the use of context as a concept, a new generation of information systems was born, named context-sensitive services (CAS), and became an essential technology for the creation of context-aware applications. Unfortunately, the multiplicity of execution services and the difficulty to model the context makes the CAS quite challenging to adopt.

In this document, we will start by presenting our meta-model of context based on an accurate specification of searching e-deals. Then, based on this meta-model, we will propose a detailed specification of Context Aware Service and meta-model.

Keywords: Context-Aware Service; Context meta-model.

1. Introduction

The high use of mobile devices and infrastructures (IPhone, PDA, HTC,..) allowed smoothly the access to end users to the services from any location, and at any time. However, the services are still mostly unaware of the end user context. The service remains unchanged regardless the changes in the user's environment or his information needs that would be very apparent to a human observer.

Taking into account the variety of mobile device and telecommunication infrastructure, applications gets faster, and consequently the challenge of developing them grows and takes a large scale.

To meet this requirement, the service needs to be personalized according to each category of end users, and adapted to different views, mobile devices and infrastructures, and different contexts of use. This is a key factor in order to establish a relation between the service and the end user and consequently improve the flexibility of the service. A Context-Aware Services (CASs) responds to this requirement, since it does automatically adapt services to the environment of a user, for instance to the location, time or physical condition (emotion, intentions…) (see Figure1).

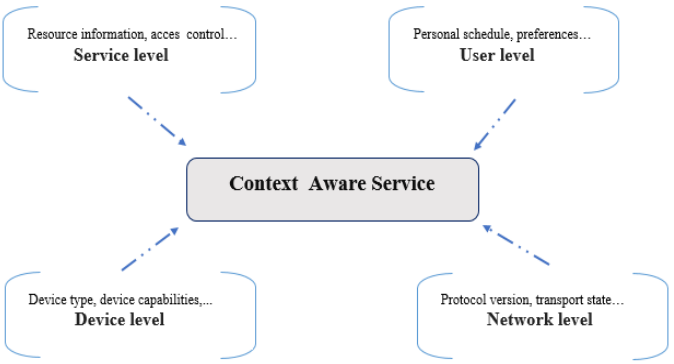

Fig. 1. CASs as group of many technological domains

Based on the current example of a Deal Searching service, who returns to end users a series of results according to their locations (coordinate, user profile etc.), preferences, taking into account the capabilities of the device used and its limitations over time. Generally, this information is qualified context. Nowadays, designing systems based on CASs must implement every single element in interaction with the context and information related to the use of it. For this reason, it is difficult to model the CAS given its complexity, which makes the CAS's activity lasts consuming time. As a matter of fact, the complexity of the use of context's information and the different possible cases are the key words related to challenge of implementing CAS. This capability is particularly critical in ubiquitous environments, where context is the main element of mobile systems [1].

This article focuses on introducing an extending UML (Unified Modeling Language) to support context design

for modeling context concept and CASs in order to support development on a specific application like (i.e. E-Deal) independently of the mobile devices and infrastructure. The proposed approach "iML" in this article provide the approach to construct the context, and model the dependency between context and CASs models. Finally, context and CASs implementations in development can be generated automatically from meta-model. This approach allows achieving the objectives in term of reuse, evolution, integration and maintenance.

This paper is organized as follows: In next section, we present a scenario concerning an E-Deal system, which will be used in subsequent sections as an illustrating example. In section 3, we introduce and detail our context meta-model based on E-Deal. Then, the section 4 will give a description of our CASs meta-model. In section 5, we introduce the extension iML of UML, and demonstrate how this extension can be applied to CASs design by projecting this approach in the E-Deal system. Finally, the section 6 will sump the topic of this article.

2. Deal scenario Deal searching service

For instance, let's imagine that a user wants to consult the daily deal, to do this, he will need to connect through his mobile device (Samsung, Nokia, Sony....) to an E-deal application to visualize a list of adequate deals. He uses the corresponding service, sends his request and then the application returns the following answers:

1. No possibility to call the service due to the dependence of the service with the equipment used, for the case of mobile use there may be a service failure.
2. On the other hand, the service returns an inadequate answer for user expectations because the system does not include all parameters like kind of user terminal, user position, user language and his preferences (food preferences, restaurants prices).

According to this case, it is imperative that our E-Deal system must be context-aware to use as input the context of the user and remains adaptable to any changes in the context through time. In fact, irrespective of the system, the design must be Context Aware, as matter of fact, as the user connected to the service, the system will fetch automatically (time is taken into account: it's midday for example) a list of deals well presented (for display adaptation the system will consider the terminal type), about its location (system will consider localization of user), the system displays the information in the user's language (taken into consideration the user's language) and taking into account his preferences (food preferences for instance). Another point to note, is that this system must provide a list of results allowing users a dynamic pagination concept for easy navigation through the list of results (take into consideration the technical requirements of the terminal in question, memory in this case like RAM) to identify the low-memory problem or any problem with the memory that can block the system. Also it must be able to detect any change from the context of the user (e.g., insufficient light, need more charge in the battery - minimum threshold), and be able to adapt its behavior to the context of change in order to optimize the end-use.

3. Context

Context is the information that characterizes the interactions between humans, applications, and the environment [14]. Information characterizing the context is dependent on system's domain, as a type of information might be considered as context information in one domain but not in another one. So, several context definitions were proposed in the literature [6, 15] serving various domains, however the context definition, given by Dey and Abowd, remains the most generic. Indeed, these authors have defined context as "any information that can be used to characterize the situation of an entity. An entity is a person, place or an object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" [16]. As mentioned, We define any context parameter is a new information used to extend the modules of the system, use of parameters improves the behavior of a service and makes it even more addable system to the different cases. Without these parameters, the service should be consumable as a simple service, but with context information such parameters, we deduce that the service works as a generic service [10]. The CAs environments are characterized by the fact that the context contains any information useful and responsible in the use of the service to ease the execution. Some of the examples including context are listed below:

1. Context depends on a request made by the user requesting a call from a given Service (the client uses a specific service), taking into account the information which is the subject of the application: ie the current position of the user's site, his choices and his favorite recreation and other information (agenda, calendar)
2. Context depends on a web service, as a service that returns the root of a requested path, location service, and traffic service.

3. Other contexts like time (weekend, night, vacancy) and physical condition (emotion, intentions…).

It is noted that there is a segment of contexts which is very specific to a particular applications: Like Calculating the availability of a hotel's rooms during a certain period. This service can be called through a vacation planning managing service, not in a traffic calculation service.

Sometimes it is easy to detect some context information (e.g. location). While we have context linked to other contexts. It is important to say that most providers advertise their services — called context information services—over the Web that can be integrate easily into CASs. [11, 4, 9].

4. Context Aware Service

In 1994 appears the first uses of concept of context-aware [13]. We can say a service is context-aware, when the service is ready to be adaptable to user behavior despite the multitude of their contexts and the dependence between these contexts [16, 17, 12]. A service is defined as a roping independently calculated, and this is the machine that takes care of quickly at low cost and composition of the coupling and distributed software application [8]. For development CASs, we must implement all that is related to the management of the context and its dependencies, also including the use of parameters used in the information in order to identify the context. CASs developers need to know the type of information to be used to implement the system. Due to the heterogeneity of context providers, the quality of the parameters used in the context of information and environmental diversity using information from the context [11].

The second issue for developing to adapt the behavior of the CAS is to use specific mechanism taking into account the context information without explicitly intervene in the user intervention. For this purpose the great problematic is how to use the information context to have a context awareness of services. To add that, the concept of abstraction in sensitive contexts mechanism makes the development and maintenance of CASs flexible and easy over time. In this case it is preferable to use the concept of the legacy web service to ensure flexibility in the development of CAS and not impact the implementations of web services. [3].

5. Interactive Modeling Language Extension

This section focuses on introducing the profile « iML » (Interactive Modeling Language). In this section, we start by presenting the basic principles of « iML », then we will move to describe the concept of context-sensitive class and associated mechanisms, and finally we will illustrate the different concepts by excerpts from the case study «E-Deal» used as a reference for the illustrations.

5.1 Origin

« **iML** » is a UML profile based on the modeling approach by context integration. This profile provides a formalism that extends UML. « **iML** » based explicitly on the UML standard. It introduces a set of concepts and mechanisms to manage access rights to sensitive classes to the context of use, specialize sensitive class context, specify dependencies between contexts, ensure model consistency in case of updates, and administer views at runtime.

Informally, the key concepts of « iML » are defined as follows:

1. Actor: human or external entity that interacts with the system.
2. Scenario execution: the vision system in a given context (or part of this system). A single scenario is associated with a user.
3. Context: entity modeling (static). It is the application of a point of view on a given (class, and generalized the whole system) entity. For simplicity of language, we say that a view is associated with a player taking as implicit entity on which the perspective of the actor applies

5.2 Basic Concepts: Context Class

A context class is defined as a unit of abstraction and encapsulation to store and release information based on the context of use. This class is composed of a base (stereotype «base») that has the same name as the corresponding UML class, and context (stereotype « context ») which are connected to the base via a dependency called « contextExtension ». The relationship contextExtension is not an inheritance relationship: the contexts depend on the base in the sense that the attributes and methods of the base are implicitly shared by the contexts of the context class. One characteristic of a context can redefine a feature of the database. Figure 2 below shows the static structure of a class contextual.

Figure 3, illustrates the structure of a context class through the simplified example of contextual Deal class, consisting of a base and four contexts corresponding to the context locations (coordinate etc.), preferences, even the used device capabilities and finally the time (week end, night, vacancy …).

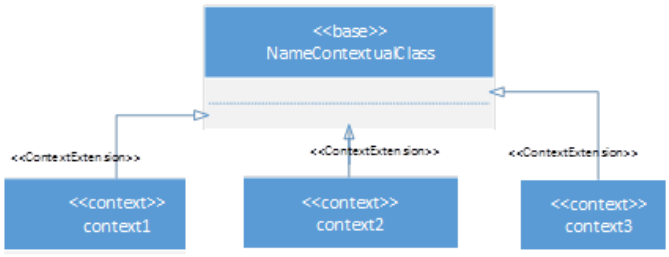

Fig. 2. Static structure of a context class

The database contains the features (attributes and methods) considered irreducible, that is to say independent of a particular execution context:

Localization class that describes the location, while class Device describes the type of service usage as mobile devices and infrastructure (smart phone, tablet, PDA, PC…). And prior to the last there is also class Preference that describes the preferences of user (shopping, game, travel...), a finally we have class Time that describes context of time.

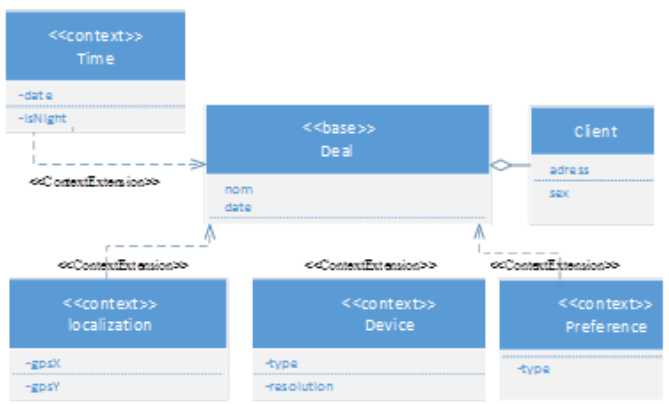

Fig. 3. Simplified example of multicontext class Deal

In some particular cases, there might be a dependence between the contexts. If we have for instance many contexts for a multicontext class, they could naturally be dependent. In other words, changes in attribute's values from one context can affect the values of attributes in other contexts. It is therefore necessary to maintain the internal consistency of such a class. Consequently, management is usually part of the implementation phase in the sense that the code must be entered to make the necessary updates. However, it is preferable that these dependencies between contexts are made explicit as early as possible in development, that is to say, during the design phase; why a relation of dependency stereotyped by « context Dependency » was introduced using the UML notes or OCL (Object Constraint Language) [OMG-OCL] to specify constraints.

Figure 4 shows an abstract example of a dependency between two contexts of a class multicontext. Dependence "contextDependency" between contexts context1 and context2 indicates that data context1 context (source of addiction) depend on some data context2 context (target dependency).

The relationship between these data should be described in OCL on the note associated with addiction. Note that there are several types of dependencies between contexts: Inclusion of data dependence, dependence data and equal functional dependency.

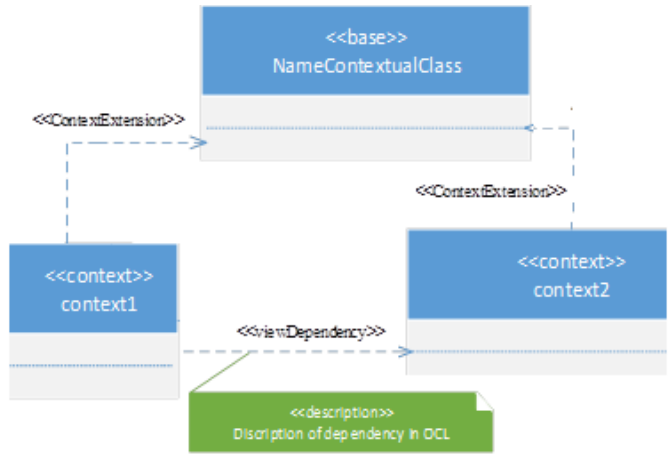

Fig. 4. Abstract illustration of a dependency between two contexts

5.3 Metamodel of iML

Figure 5 provides an overview of the iML meta-model, added to the UML elements, which are marked by dark color.

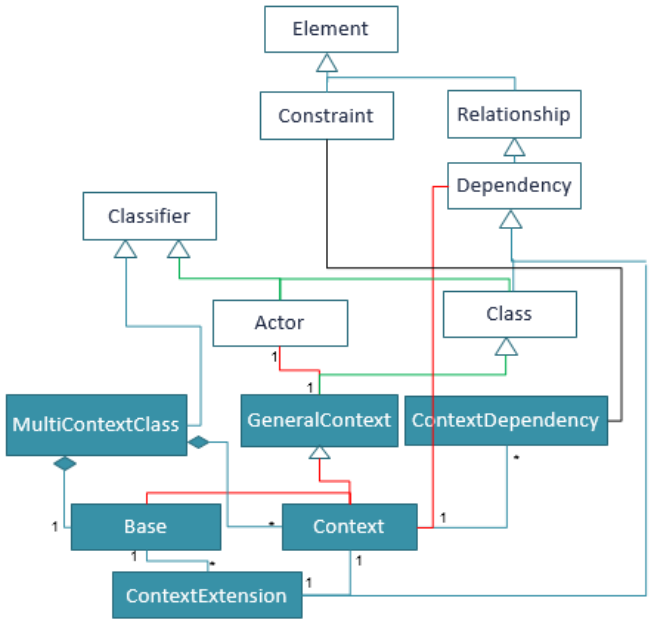

Fig. 5. Metamodel associated with the profile iML

Here we only give the informal semantics associated with iML. The informal semantics associated with elements of the meta-model iML is:

1. **Base**: Element specializing meta-class Class, which describes the common system actor's structural and behavioral characteristics. Base element can be associated with dependency relationships contextExtension that can connect to the database contexts.
2. **Context**: Element that allows to model structural characteristics and specific behavioral to an actor. It is connected with the base via relationship contextExtension. It can also be a source or a target of one or more relationships contextDependency.
3. **MultiContextualClass**: Element specializing meta-class Classifier. It consists of a base and a list of context connected to the base via contextExtension.
4. **ContextExtension**: Element specializing meta-class Dependency. It is an addiction as having a context source and a target base. The basic settings depend on the sense in which the attributes and methods of the base are implicitly shared contexts multiContexte the class.
5. **ContextDependency**: Element for modeling the dependency relationships between contexts. ContextDependency each can be associated with one or more constraints that can be expressed in natural language or formal language such OCL.

5.4 Case Study

To illustrate our diagram iML, let's project it on the case of the E-Deal system presented in this section (See Figure. 6):

6. Conclusion

In this article, we've presented a meta-model for the modeling of context-aware services CASs following an "iML" approach by context integration independently of the mobile device, its infrastructure and the other application concerned by the same domain.. We have presented techniques for development of CASs, in particular we defined the meta-model of CASs form as ContextUML with UML. And then we treated deeper the modeling of context information, CASs specifications by proposing an extension of interactive modeling language in order to mature the reflection of this topic.

It is difficult to develop the CAS given its complexity which makes the CAS's activity consumes a lot of time during its execution.

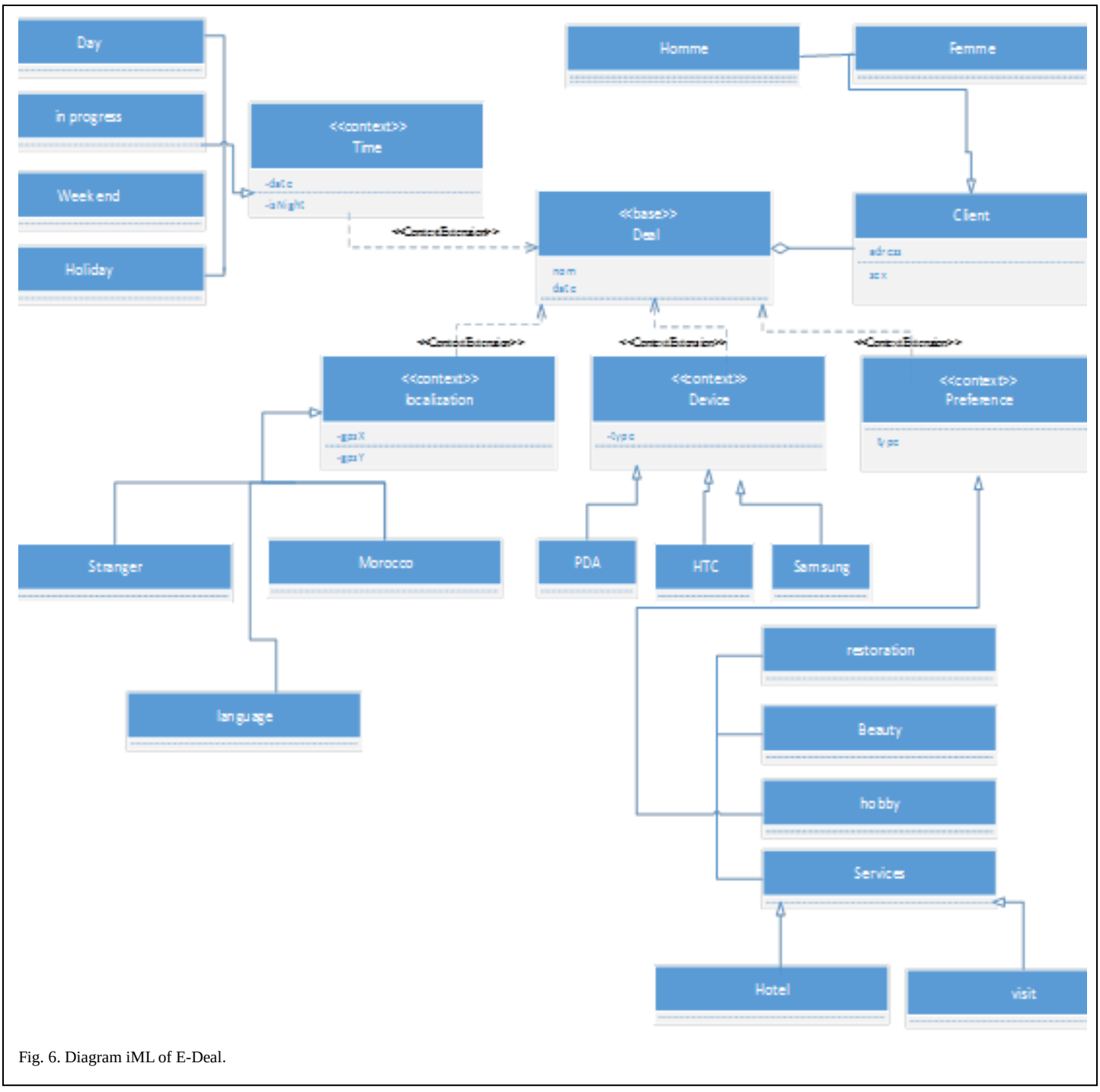

Fig. 6. Diagram iML of E-Deal.

References

[1] Cohen, N.H, Black, J., Castro, P., Ebling, M. Leiba, B., Misra, A., Segmuller, W. "Building Context-Aware Applications with Context Weaver". IBM Research Report RC 23388, October 22, 2004.
[2] Enabling Context-Aware Web Services: Methods, Architectures, and Technologies Chapman and Hall/CRC (Ed.) (2009), pp. 113-135.
[3] Enabling Context-Aware Web Services: Methods, Architectures, and Technologies Chapman and Hall/CRC (Ed.) (2009), pp. 45-46
[4] Buchholz. T, Krause, M., Linnhoff-Popien, C.,Schiffers, M.: CoCo: Dynamic Composition of Context Information. In Proc. of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), Boston, Massachusets, USA, August 2004.
[5] Enabling Context-Aware Web Services: Methods, Architectures, and Technologies Chapman and Hall/CRC (Ed.) (2009), pp. 44
[6] Chen, G., Kotz, D. A Survey of Context-Aware Mobile Computing Research. Technical Report, Dartmouth College. 2000.
[7] Frankel, D.S., Model Driven ArchitectureTM: Applying MDATM to Enterprise Computing. John Wiley & Sons, 2003.
[8] Papazoglou M. P. Service Oriented Computing: Concepts, Characteristics and Directions. Actes de la 4° conférence WISE`03, Rome, 10-12décembre 2003, Italie, IEEE Computer Society, p. 3-12.
[9] Henricksen, K., Indulska, J., A Software Engineering Framework for Context-Aware Pervasive Computing. In Proc. of the Second IEEE Annual Conference on Pervasive Computing and Communications (PerCom'04), Orlando, Florida, USA, March 2004.
[10] Truong, H.L., Dustdar, S., Context Coupling Techniques for Context-aware Web Service Systems -An Overview, November 2009.
[11] Salber, D., Dey, A.K., Abowd, G.D., The Context Toolkit: Aiding the Development of Context-Enabled Applications. InProc. of the Conference on Human Factors in Computing Systems (CHI'99), Pittsburgh, PA, USA, May 1999.
[12] Schilit, B., N. Adams, Want, R., Context-Aware Computing Applications. In Proc. of the 1st International Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, USA, December 1994
[13] Schilit, B., Theimer, M., Disseminating Active Map Information to Mobile Hosts. IEEE Network, 8(5):22–32, 1994.
[14] Brézillon, P. Focusing on context in human-centered computing. IEEE Intelligent Systems 18, 3 (May/June 2003).
[15] Schmidt, A., Beigl, M., Gellersen, H.-W. There is more to context than location. Computers and Graphics 23 (6), 893–901, 1999
[16] Dey, A.K., Abowd, G.D., Towards a Better Understanding of Context and Context-Awareness. Technical Report GIT-GVU-99-22, GVU Center, Georgia Institute of Technology, June 1999.
[17] Pascoe, J., Adding Generic Contextual Capabilities to Wearable Computers. In Proc. of the 2nd International Symposium on Wearable Computers, Pittsburgh, USA, October 1998.