

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2014) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks
(EUSPN-2014)

Modeling Context Aware Features for Pervasive Computing

Hamid Mcheick^{a*}*^aProfessor, 555 Boul De l'Universite, Chiuoutimi, G7H-2B1, Canada*

Abstract

For more than two decades, research in the development of context-aware applications has gained significant attention, in which the context aware should be taken into account to adapt them to the requirements of the environment and users. However, advances in application models to support the development of these systems have not kept up. Researchers have been building and deploying context-aware applications with their behaviors often tailored to specific problem domains, by designing the anticipated context and the desired application behavior. Motivated by the raised facts, this paper presents a context aware model with ability to interact and to adapt smartly and dynamically to environment and needs of users. We are revisiting the context life cycle, especially the representation and the modeling of context features, regarding the relations within these features and the systems functionalities. Different kinds of components adaptation are described and scenarios are presented to illustrate these adaptations. As a proof of concept, we have simulated the context model in health care systems and show the results.

© 2014 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of the Program Chairs of EUSPN-2014.

Keywords: Pervasive and Ubiquitous Computing; Context aware Applications; Software adaptation;

1. Introduction

Computer is evolved in many steps starting from microprocessor and personal computer, to networks and communication to World Wide Web, to mobile-internet and so the social networking. Today, the computer is everywhere, connected to everything and embedded in every object. The advance in sensing technologies and the development of data extraction and management capabilities are providing more and more accurate information about the user and application runtime environments. These information provide a detailed view of the system and

* Hamid Mcheick. Tel.: +1-418-545-5011; fax: +1-418-545-5017.

E-mail address: Hamid_Mcheick@uqac.ca

user context. In addition, market research has shown significant growth in the deployment of sophisticated sensors, generally embedded in faster, smaller and globally connected devices.

Context-aware systems are emerging as an important class of applications. Such systems can respond intelligently to their physical world changes, acquired by using sensors, and according to the information about the computational environment and user's needs [1]. However, many of today's context-aware system architectures tend to treat the need of no-human intervention by a static design of artifacts that implement the variations of needs in the software features during the development cycle.

Contemporary systems must follow a flexible and adaptive architecture to perceive, sense, respond and, in consequence, to suit the particularity of their operation environments [29]. A combination of runtime adaptability and context features aware modeling offers best approach to deal with the need of reconfiguration during runtime to satisfy human needs. We believe that to achieve such efficiency in context-awareness, a system must be able to i) add (update) and remove functionalities with (nearly) no-human interference in, ii) offer variety of binding times with the focus on, and iii) fast response to condition changes.

Thus, we are providing a model to allow context-aware systems to be self-adaptive and self-evolving during execution. The approach encourages a high-level of abstraction and supports building context-aware systems by providing modularity and separation of concerns.

To adapt an application to a context, we should take into account many kind of compositions based on reasoning approaches, such as rule based reasoning, fuzzy logic, probabilistic logic, Bayesian system, and ontology based reasoning. If the components are on a mobile device, it is limited by the resources and must take into account the performance, for example. However, the context is often changed dynamically. Therefore, we need to do the linking (binding time) during the runtime phase, which reduces the performance of this unit. The problem is to find a solution to these situations.

The rest of this paper is organized as following: section 2 shows previous and related works. Section 3 describes the detail of our approach. Section 4 presents the details of different kinds of contextual application adaptation. Section 5 presents the simulation of context aware model win healthcare systems. Finally, section 6 concludes and mentions some of the future remarks.

2. Related works

Many researchers have been trying to define the term context in their studies. In this section, we present briefly, in a chronological order of appearance, the different definitions of the term which allow us thereafter to give our own definition. Schilit *et al.* [2] describe the context as a physical environment of users and response to the questions: i) Where are you? ii) Who are you with? iii) What resources are nearby? So according to them, the context is the location of the user, the people and objects in its vicinity. Schilit and Theimer [4] define context as the user's location, description of people and objects nearby and changes in these objects.

Brown *et al.* [8] assimilate context to the location, time, season, temperature, and the identity of the people in the environment of the user. Ryan and Pascoe [9] identify the context as the location and identity of the user as well as the environment and time. Ward and Hopper [10] see context as the possible states of the application environment.

Dourish *et al.* [12] define context as a form of stable information which can be defined. Henricksen *et al.* [13] define context as the situation in which a computing task takes place. For Brezillon [14], a context is "What is not directly involved in solving a problem but forces its resolution". Brezillon concludes after an analysis of the different existing definitions of context that the majority of the latter is only the answer to the following questions: i) Who? Identity of the user and those around him; ii) What? Description of the user's environment; iii) Where? Location of the user; iv) When? Time; v) Why? The reason for being in a situation; vi) How? Description of circumstances of the situation. These responses represent a major information rate, some of which are unusable.

Hoareau and Sato make a state of the arts on the different context templates. In their work [15] the focus was on the representation of contextual data that describes the state of an entity. Also, based on the work of Chen and Kotz [16] and on the evaluation of context templates made by Strang and Linnho - Popien in their work, [17], the authors make a model evaluation context. However, there is a few research done on context to generalize its definition that

would apply to all areas. Especially, when we consider a simple (temperature) and composed (GPS) elements in context, these definitions do not take care of the different elements.

We distinguish six models of context as follows: *<key, value> Model*, *Markup Model*, *The object-oriented model*, *Model based on logic*, *Ontology Model* [18], *Situational logic* [19].

Each model is suitable for a particular application class. In fact, each model is more or less depending on the practical needs of the application. Therefore, context models are developed for an application. In this paper, it is important to consider a template of models from which we can instance a specific model for a specific application.

3. Context Modeling and Lifecycle

3.1. Context definitions

For more than two decades, Context-awareness has existed and been treated as core feature of ubiquitous and pervasive computing systems. This term, also referred to as sentient, was first coined by Schilit and Theimer in 1994 [4]. As stated by Abowd *et al.* [5], definitions are too specific to computer systems and cannot be used to identify whether or not a given system is a context-aware system.

Motivated by the raised questions, we expand the definition of context awareness. Schilit *et al.* [2] illustrate context as physical environment and user's profile characteristics. Beyond definitions that enumerate context as a set of information (location, temperature, user identity ...), it is important to take care of the definition introduced by Zhou *et al.* [3]: "Context is any information characterizing the situation of a task session or interaction between a user and his/her service world. Context is categorized into user context, peer context, process context, physical context and service context." Indeed, in our point of view, context does not encompass only information about the user, but also the conditions under which the device and the service providers operate.

Paccoc generalizes the notion of context by proposing the following definition [27, 28]: the context is a subset of physical and conceptual states which having interest to a particular entity. Then, several clarifications to the definition have been made such as Dey's definition [5, 11, 28]: Any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that may be relevant to the interaction between the user and the application, including the user and the application themselves. This definition is considered as the most complete and adopted by the researchers.

However, Chaari *et al.* [28] consider that this definition does not separate the data belonging to the context of those belonging to the application. According to them, this separation is important in the design of a system responsive to the context. Therefore, they brought more information to the definition of Dey: *The context is the set of external parameters of an application that may affect its behavior by defining new views of its data and services.*

We have found that this last definition is the most appropriate for research in healthcare systems to distinguish between data of context and data of the application and keep them separately.

We define (*initial model*) the context (CT) as a partial order of contextual elements (CE), which are defined by a precedence relation: a context may replace or modify another context. For example, heart attack context CT has higher priority than diabetic context CT which has higher priority than a tourist context CT. A contextual element (CE) is defined as a set of properties and behavioral rules. A property has a name and a type. A behavioral rule is defined as follows: (pattern) if value (name) \in S then call function f with parameters p_1, p_2, \dots, p_n with priority X. The properties of the function f will be considered under this rule when a context applies on an application. f may use a subset of these parameters. We argue that this pattern is applicable to any application using the defined rule. This rule is validated in the rest of this paper (see section 3.3 for more detail).

3.2. Context Lifecycle

A data life cycle shows how data moves from phase to phase in software systems for example in application, middleware. Specifically, it explains where the data is generated and where the data is consumed. In this section we consider movement of context in context-aware systems. Context-awareness is no longer limited to desktop, web, or mobile applications. In other terms, context management has become an essential functionality in software systems.

As depicted in the survey by Hynes et al. [6], there two different categories of context life cycle. The Enterprise Lifecycle Approaches (ELA) which are based on industry standards and Context Lifecycle Approaches (CLA) specialized on context management and lack of standardization and tests [1]. For more detailed information, reader may refer to the survey of Perera *et al.* [1]. An appropriate context lifecycle consists of four phases: Context Acquisition, context Modeling, Context reasoning and Context dissemination. Bernardino *et al.* [7], in the JDL fusion model, identified three phases in a typical context management system: context acquisition, information processing, the last phase corresponds to reasoning and decision.

Context-aware system must take smart decision about when, where and what service to provide to the user? Thus, designers of such systems have to include the ability to dynamically add, change and select the appropriate needed functionality or service. The following figure illustrates the way of the relationship between a context management system and the applications or the service providers.

The next section describes our vision and modeling technique.

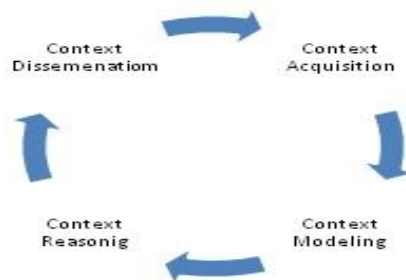


Fig. 1. Simplest form of context life cycle [1].

3.3. Context Modeling

The majority of research papers in this field focus on the context in a particular application or an application category. In this study, we choose to develop the context for applications as a initial generic model. This model is applied to the medical sector in section 5.

To model the context of an application, first of all, one has to look for different elements that affect the application to infer contextual elements. In our work, we seek to meet the needs of a patient who wants to manage his state at any time and location. So, our application must analyze the condition and react according to its context. For example, if the patient has a very high temperature, the application searches for the nearest emergency and is responsible for the contact. Several factors affect the application, to simplify our work we have chosen to divide this set of contextual elements into three subsets:

- The first includes medical variables such as temperature, oxygen levels in the blood, heart rate, ...
- The second contains the environmental variables such as the location of patient, OS, humidity, ...
- And the third is specific to the user, it contains the name, age, medical condition, ...

We note that the first two subsets change often, they represent the dynamic elements of context and they have a great influence on the system. The third subset rarely changes and it has an effect on the application but not as much as the dynamic elements of context. In our application, we must model the context so that it includes all the contextual factors that influence an application. The model that best meets this specification is the XML model. This model helps us to represent an image of context in a specific time. The application must also detect the change in context, a review according to contextual elements should be performed if necessary.

However, the XML model cannot accomplish this task since its use is difficult, especially for dynamic contextual elements. These elements are constantly changing and the application must adapt to their variations. Therefore, we introduced another model: the object that handles the dynamic elements of context-oriented model. Indeed, this model perfectly meets the need of this type of information and also offers the possibility of constructing methods to manage context data.

In this context, we propose to use a combination of two models: The object-oriented to adapt well to dynamic elements change that affect the applications. The XML model for common elements that do not change often, such as elements that represent the profile of a patient.

In Figure 2, we represent the context by a class diagram (a and b). The context is the main class in the model. It has a location, user profile, terminal profile, set of contextual elements and rules. Figure 2.b represents the meta-model of the context which consists of a set of rules and contextual elements. Each element can be a common (Location) or/and specific (temperature) view. In other words, a context could be seen as a set of views. For example, a cardiologist and diabetes doctors could see the context of patient in different views. With this model you can have several applications at the same time taking such an application of health and tourist.

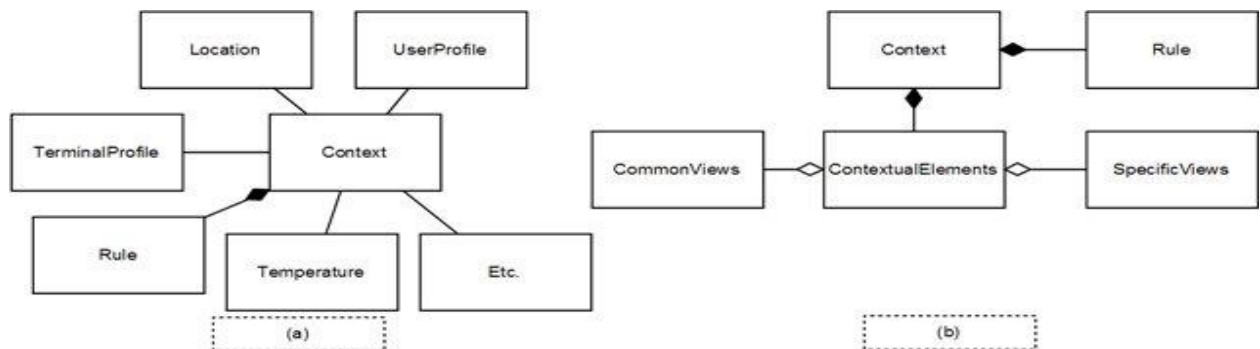


Fig. 2. (a) Context aware Model, (b) Meta model of context

4. Adaptation types of applications depending on state of context

Exploiting the change of contextual properties enables pervasive and autonomous systems to adapt to the context needs. Such systems learn to sense and perceive context varies and then respond by modifying the system's behavior. Different contextual conditions invoke different functionalities in the system. Thus, suitable runtime adaptation and decision inference mechanisms enable a successful handle of the demand and complexity of divers use cases. The intention is to identify the types of adaptation scenarios necessary to take into account the types of context and interactions to modify the behavior of an application. Existing adaptation contextual scenarios of an application vary depending on the field. To better understand the types of changes, consider two examples of context aware application (CA):

a) an application in tourism, which should provide contextual information such as restaurants or hotels closest. This application may undergo modification of the display.

b) a medical application, which should take actions and make decisions in place of the image (e.g. call an ambulance for an elderly person having an accident: high blood pressure). This application can trigger discontinuation underway to launch urgent treatment.

We analyze the various types of changes that an application can undergo in a given context. in our approach, we define an CA types, types of contexts and types of adaptations. We list some adaptation scenarios as follows:

a) adding a new treatment more than the basic one. Each user can have a set of components available dynamically depending on context. For example, after a first treatment (eg surgery), the patient may need to have a new treatment radiotherapy.

b) take a different path in a global predefined process, compared to the normal path. During his usual medical examination by her doctor, a patient may have an urgent hospital treatment. The usual feature of blood at the clinic will be replaced by urgent treatment in the overall process.

c) change the process. A patient may change the process if there is a new context (ex. new disease).

There are different techniques of composition of components and functions that are different depending of the level of composition :

- a) linking (binding time) in compilation or execution time,
- b) flexibility, and
- c) performance.

For example, if the components are on a mobile device, it is limited by the resources and must take account of the performance, for example. However, the context is often changed dynamically. Therefore, we need to do the linking (binding time) during the implementation phase, which reduces the performance of this unit. The problem is to find a solution to these situations.

There are several techniques to adapt applications that we classify into two categories: for static linking (early binding) and dynamic linking (late binding). In the case first case, all applications must be known in advance and changes are made in a fixed manner. In contrast, in the second case, editions links are made dynamically according to the context. But we must take into account the constraints of mobile devices such as performance, security, etc..

There are several mechanisms of implementation techniques of dynamic adaptation in middleware applications that intercept calls and modify as needed. These middleware can use dependency injection in Spring Framework [20], JBoss AOP [21, 22], OSGi [23, 24], and the formalism of connections (I/O) between software components technologies in Service Component Architecture (SCA) and FraSCAti [25]. We have chosen to use this last architecture (FraSCAti) and Spring Framework to configure dynamically the behavior of our application.

A context aware application can be adapted dynamically using the rules of context, rule based reasoning mechanism, to add, remove, activate or deactivate operations of components (figure 3). The adapter of operations uses rule based reasoning to infer which reference should be updated dynamically to an operation.

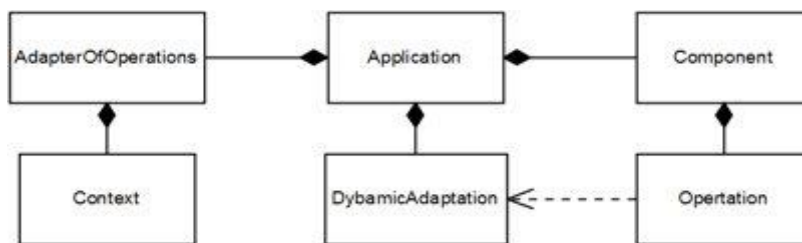


Fig. 3. Context aware Adaptation Model

5. Simulation of Context-aware Model

We decided to build our simulation based on the above context aware definition and model using our context aware architecture [26] and dependency injection (IoC). Context aware model is instantiated for a health care and patient application using a few rules to validate it. The rule pattern is validated in this scenario: We have applied it on temperature and heart rate parameters. The function is called dynamically based on the intervals of context.

This simulation uses publish-subscribe and Observer design patterns. In our case, applications represent observers and agents represent subjects. It is possible for an application to subscribe to a set of heterogeneous agents and to receive their associated notifications. The exchange protocol is located at the Master Agent level that stores subjects-observers couples (applications / agents) also acting as directory agents for applications. Specifically, we used the concept of separation of concerns through AspectJ to model the spread of a notification of an agent subject to observer applications of this agent.

We have created generic interfaces to represent the concepts of agent application and sensors, to create an implementation of these interfaces as necessary. For example, temperature sensors or the heart rate are two different implementations of the sensor interface. Virtual sensors are used for simulating activity and sending data at regular intervals which is associated with a reference to the corresponding agent. And according to the same logic it is possible to create several different implementations of agents for the same sensor. The interest is to easily change the logical processing of the data for a given sensor, such as the activation threshold.

Finally, we integrated inversion of control (IoC) mechanism so as to configure sensors, agents and applications from an XML file. This mechanism is implemented using Spring Framework [20]. It is therefore possible to

determine the parameters related to the processing logic agents and the behavioral characteristics of virtual sensors (amplitude variations, frequency of updating, unit of measure). Before making the whole more consistent and can easily test the interaction between the different layers, we have also developed simple applications that highlight the notification of the master agent through journaling.

We consider two applications: one control the temperature received from sensor temperature agent and another control the agent heart rate agent. These agents monitor the sensors using aspectJ and injection of control (IoC) mechanisms. Figure 4 shows a scenario and results of this simulation.

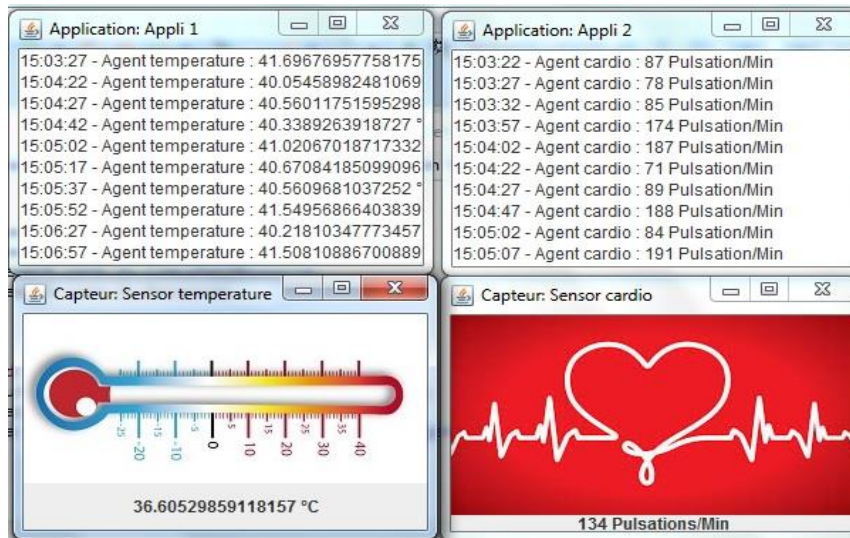


Fig. 4. Context aware Model Result using two applications

6. Conclusion

In this article, we focused on the sensitivity of context aware, in particular, applications in the medical sector. Starting from the definition of the context to modeling. First, we reviewed various definitions of context. We acknowledged that the context must be defined in relation to interaction with the user to make sense. Then, based on existing definitions, we introduced our own definition as a template of context that best suits any application, and could be dedicated to the field of the doctors. Secondly, we discussed the different existing models of context but noticed that none of them satisfies our requirements. So we proposed a model to satisfy the generic applications. Our model is composed of an XML model for static elements that change rarely and an object-oriented model for dynamic elements. We have also developed and grouped elements that may influence the application.

Also, we describe the most important types of adaptation in components related to a context. The scenario is given to validate our contextual model and to illustrate these adaptations. Thus, as a proof of concept, we have simulated the context aware model and we show the result of this model. In the future work, this contextual model will be applied in different scenarios, also extraction and acquisition of context aware and will be done.

Acknowledgements

This work is supported by the Department of Computer Science and Mathematics at the University of Quebec at Chicoutimi, Quebec Canada.

References

1. C. Perera, A. Zaslavsky, P. Christen, and D. Geogakopoulos, "Context Aware Computing for The Internet of Things : A Survey," pp. 1–41, 2013.
2. B. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications," in *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, 1994, pp. 85–90.
3. M. Zhou, R. Zhang, D. Zeng, and W. Qian, "Services in the Cloud Computing era: A survey," in *Universal Communication Symposium (IUCS)*, 2010 4th International, 2010, pp. 40–46.
4. B. N. Schilit and M. M. Theimer, "Disseminating active map information to mobile hosts," *Network*, IEEE, vol. 8, no. 5, pp. 22–32, Sep. 1994.
5. G. Abowd, A. Dey, P. Brown, N. Davies, M. Smith, and P. Steggle, "Towards a Better Understanding of Context and Context-Awareness," in *Handheld and Ubiquitous Computing SE - 29*, vol. 1707, H.-W. Gellersen, Ed. Springer Berlin Heidelberg, 1999, pp. 304–307.
6. G. Hynes, V. Reynolds, and M. Hauswirth, "A Context Lifecycle for Web-Based Context Management Services," in *Smart Sensing and Context SE - 5*, vol. 5741, P. Barnaghi, K. Moessner, M. Presser, and S. Meissner, Eds. Springer Berlin Heidelberg, 2009, pp. 51–65.
7. A. M. Bernardos, P. Tarrio, and J. R. Casar, "A data fusion framework for context-aware mobile services," *Multisensor Fusion and Integration for Intelligent Systems*, 2008. MFI 2008. IEEE International Conference on. pp. 606–613, 2008.
8. Brown, P. J., J. D. Bovey and X. Chen. 1997. « Context-aware applications: from the laboratory to the marketplace », p.58-64 IEEE Personal Communications.
9. Ryan, N., J. Pascoe and D. Morse, 1997. « Enhanced Reality Fieldwork: The Context-aware Archaeological Assistant ». In *Computer Applications in Archaeology*. Oxford, UK.
10. Ward, A. J., and A. Hopper, 1997. « A New Location Technique for the Active Office », p. 42-47. IEEE Personal Communications, vol. 4, no5.
11. Anind K. Dey, 2001. « Understanding and using context ». *Personal and ubiquitous computing*, vol. 5, p. 4-7.
12. Dourish P, 2001 "What we talk about when we talk about context", Submitted to HCI Journal for publication in special issue on context-aware computing.
13. Henricksen, K., J. Indulska and A. Rakotonirainy, 2002. « Modeling Context Information in Pervasive Computing Systems », p. 167-180. In *Proc. of the First International Conference on Pervasive Computing, Pervasive'2002*.
14. Brezillon, P., M. R. Borges, J.A. Pino and J.-Ch. Pomerol, 2004. « Context-Awareness in Group Work: Three Case Studies ». In *IFIP Int. Conf. on Decision Support Systems* .
15. Christian Hoareau and Ichiro Satoh, 2009 «Modeling and processing information for context-aware computing : A survey, *New Generation Computing* 27.
16. Guanling Chen and David Kotz, 2000, «A survey of context-aware mobile computing research, Tech. report, Department of Computer Science Dartmouth College, Hanover, NH, USA.
17. Thomas Strang and Claudia Linnho-Popien, 2004, «A context modeling survey, Workshop on Advanced Context Modelling, Reasoning and Management, UbiCom, The Sixth International Conference on Ubiquitous Computing.
18. Harry Chen, Tim Finin, and Anupam Joshi, 2003 «An ontology for context-aware pervasive computing environments, *Knowl. Eng. Rev.* 18, no. 3.
19. Seng W. Loke, 2004 «Representing and reasoning with situations for context-aware pervasive computing : a logic programming perspective, *Knowl. Eng. Rev.* 19.
20. Spring Framework and IOC, Spring, <http://docs.spring.io/spring/docs/2.0.x/reference/beans.html>
21. JBoss AOP JBoss, <http://docs.jboss.org/jbossas/jboss4guide/r4/html/aop.chapt.html>
22. S. Stark and The JBoss Group. *JBoss Administration and Development*, Edition 2. JBoss Group, 2002.
23. The OSGi Service Platform Release 4 Core Specification Ver 4.1, <http://bundles.osgi.org/browse.php>
24. Binstock, A.: OSGi: Out of the Gates. Dr. Dobb Portal, 2006.
25. Seinturier L., Merle P., Fournier D., & Dolet N., Reconfigurable SCA Applications with the FraSCAti Platform. 6th IEEE International Conference on Service Computing (SCC'09,) pp. 268-275, 2009.
26. Hamid Mcheick, Hassan Sbeity, Hussein Hazimeh, Jihad Naim, and Mohamed Alameh. Context Aware Mobile Application Architecture (CAMA) for Health Care Systems : Standardization and abstraction of context aware layers, 2014 IEEE International Humanitarian Technology Conference (IHTC-IEEE). June 1-4, 2014, Montreal, Canada.
27. Pascoe, M.J. 1998. Adding generic contextual capabilities to wearable computers. In the proceedings of the 2nd IEEE International Symposium on Wearable Computers, pages 92-99. IEEE Computer Society.
28. Chaari, T., Laforest, F. and Flory A. Adaptation des applications au contexte en utilisant les services web. In *Proceedings of the 2th French Speaking Conference on Mobility and Ubiquity Computing*, pp.111-118. 2005. ACM.
29. Eunhoe Kim and Jaeyoung Choi. A Context Management System for Supporting Context-Aware Applications. IEEE/IFIP International Conference on Embedded and Ubiquitous Computing 2008.