DEVELOPMENT OF CONTEXT-AWARE APPLICATIONS IN UBIQUITOUS INFORMATION SYSTEMS

Benselim Mohamed Salah
LabSTIC Laboratory, University of 08 Mai 45, Guelma, Algeria

Seridi-Bouchelaghem Hassina
LabGed Laboratory, University of Badji Mokhtar, Annaba, Algeria

Keywords: Ubiquitous computing, Context of use, Concerns separation, MDA, Context modeling, UML.

Abstract: Nowadays, software engineering is moving towards the development of ubiquitous and distributed applications. This tendency is constrained by parameters such as mobility and heterogeneity that characterize the current situation of a user. Each new application will be able to adapt its services with the change of context of use and satisfy all user's preferences. The aim of this paper is to propose a new development approach that can take into account the change in context of use during the application development process. It permits us to develop contextual aspects of a system in a separate way and independently from the business aspects of this system and from the technological constraints of the chosen platform. Our proposal, based on the principles of MDA (Model Driven Architecture), is defined by three steps. First, the separation of contextual aspects by introducing the 3TUP process (3 Track Unified Process) and the development process as PSI "Ψ". Second, the context modeling using UML (Unified Modeling Language) and conform to a proposed context metamodel. And thirdly, the integration of the contextual model in MDA process using model merging operation.

1 INTRODUCTION

In a pervasive (or ubiquitous) environment, the great challenge of designers will be how to take into account all continuous changes of user's current situation. So, ubiquitous information systems are influenced by several factors and features like mobility, heterogenity and distribution. Thus, new application will be able to adapt its services with the change of context of use and satisfy all user's preferences. Our study aims to provide an approach for developing context-aware applications that can guarantee the contextualization of future applications. Our proposal uses the principles of MDA (OMG, 2003), and it is based on the concept of aspects separation and particularly the separation of contextual constraints. This separation allows us to study all contextual information in an independent way and without remaking the entire development process because of any changes in the context of use. Then we will use a graphical approach to represent the elements of context in order to obtain a contextual model. This one will be built according to a proposed contextual metamodel. The contextual model will be integrated into the process of MDA by model merging operation. Merging is to combine business model elements with those of the contextual model to form a merged model which will be transformed into a specific model according to the specifications of a chosen platform.

2 RELATED WORKS

Several studies and researchs have been made in the field of context and especially for taking into account of contextual changes that surround the execution of an application and introducing user's desired preferences. There are few works that are oriented towards using the MDA approach to design adapted applications and obtain personnalized information. Vale and Hammoudi showed in (Vale, 2008) the importance of MDA in context modeling and context-aware applications development, and for

Mohamed Salah B. and Hassina S..
DEVELOPMENT OF CONTEXT-AWARE APPLICATIONS IN UBIQUITOUS INFORMATION SYSTEMS.
DOI: 10.5220/0003488802230228
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 223-228
ISBN: 978-989-8425-55-3

that they designed the CSOA (Context-aware Service Oriented architecture) based on ECA viewpoints (Enterprise Collaboration Architecture). In the study (Ou, 2006), the authors have designed a contextual model using ontologies in order to represent context information, then they proposed a new architecture MDIA (Model Driven Integration Architecture) dedicated to the implementation of context-aware applications. Dey present in (Dey, 2001) a toolbox named "context toolkit" that can serve as support to adapt applications to context. Another study presented by Chen (Chen, 2000) consists to capture various contextual information that will be treated and modeled (location model, data structures). The project PUMAS (Peer Ubiquitous Multi-Agent System) is a framework based on agents. PUMAS selects the desired information from one or more sources, and adapts it with user's characteristics and with the mobile device features (Chaari, 2005). The project SECAS (Simple Environment for Context Aware Systems) is interested in adapting applications to use context (user's preferences, environment, device used, ...). SECAS aims to make adapted applications to different context of use on three components: data, services and presentation. To ensure such an adaptation, this project uses Web services (Chaari, 2005).

Our study aims to demonstrate the importance of separation between three aspects, namely: business aspects, contextual aspects and technical aspects. Our goal is to develop and design a new technique (extended MDA) by isolation of contextual constraints in a separate branch. This technique is defined by three steps: contextual aspects separation, context modeling and integration of the contextual model.

3 EXTENDED MDA

Future applications have to provide exact results and personnalized information and they must be adequate with every variation of user's current situation. Ubiquitous information systems are characterized by many features like mobility, heterogenity and distribution. These constraints represent the context of user's current situation and are not introduced explicitly in MDA life cycle. MDA is based on models and model transformation and its advantage is the concerns separation (only two branches : business and technical). But this approach does not take into account, in clear way, the continuous changes and variation of context especially with modern information systems operating in pervasive environment.

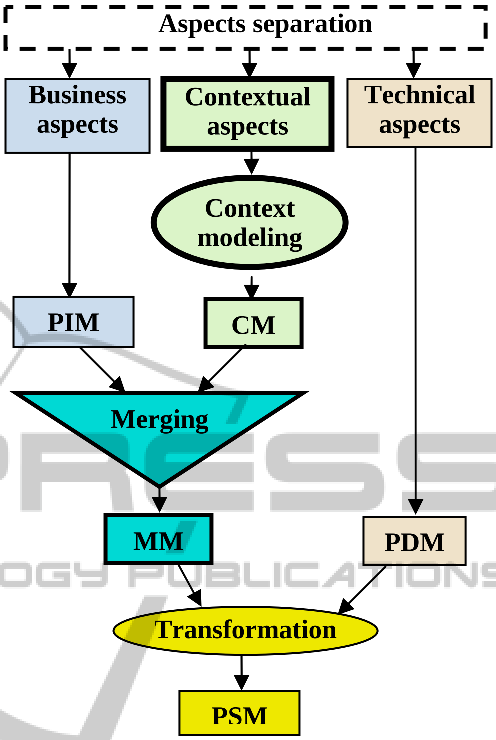

Figure 1: Architecture of extended MDA.

To extend this limit, we propose two solutions that can introduce contextual information in MDA life cycle. This information can be represented as rules or models as the preferred option. So, the first solution consists in creating "contextual rules" that represent the contextual elements of the current situation of a user and then to introduce these "contextual rules" in a mapping of the PIM-PSM transformation. The second solution uses a "contextual model" that represents all contextual aspects of a system and which will be integrated in the process of MDA using a model merging operation. Among these proposed solutions, we choose the second one to develop a new vision of MDA architecture. This new approach is defined by three steps: separation of contextual aspects, context modeling and integration of the contextual model in MDA process.

Figure 1 presents an extension of MDA approach. This new vision aims to distinguish between the contextual constraints and the two types of classical constraints (business and technical) in MDA. Context separation allows us to study all the environmental changes and user's preferences in an independent single branch. This will minimize the risk of remaking all the application development process because of these changes. Once the contextual aspects are isolated and separated, we proceed to model them in order to build a contextual model (CM) based on separated contextual information. The contextual model thus obtained is combined with the business model (PIM: Platform Independant Model). So, we obtain a merged model (MM). This one will be transformed into a specific model (PSM: Platform Specific Model) using the specifications of a chosen platform stocked in the Platform Description Model (PDM).

3.1 Contextual Aspects Separation

Why we separate the contextual aspects? (Figure 2) Indeed to answer this question, we must demonstrate that each of the contextual elements can not be handled by the business branch neither by the technical branch. For example, in ubiquitous computing, time and space are very important and they cover the concept of ubiquitous information systems (anytime, anywhere). Also, any changes of the surrounding objects can directly affect the state of the current situation of a user. On the other hand, each user may have particular preferences related to content, presentation or information display.

The concept of "context" has been defined in (Dey, 2001) as follow: *“Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves”*.

From this definition, we can extract the basic elements that constitute the main components of context of use. These elements are: the user, location, time and environment. In the field of ubiquitous computing, these elements have a common feature such as change and variation.

Ubiquitous computing offers many opportunities for access to information like random use, in time and space, of mobile devices by varying kinds of users including distant persons and nomads. Because of user mobility, not only the time and location are constantly changing but also the surrounding objects (including nearby people) and environmental factors. So, the user is facing a different situation that he is in a new context of use and with which he must be adapted. Modification or change in one element of the context of use involves the transition of this context toward another state of the user's current situation. Because of these changes, we must not remake the whole application development process, but only the part that deals with such constraints.

This brings us to distribute all constraints of an ubiquitous information system into three distinct types: business, contextual and technical. Hence the name "3-Track" which means "three ways" or "three paths" to introduce the new concept of: 3TUP process (3 Track Unified Process).

This separation permits us to develop contextual aspects of a system in a separate way and independently firstly, from the business aspects of this system and secondly, from the technological constraints of the chosen platform. Hence the need to consolidate all the contextual specifications that represent the context of use in a new independent branch named "contextual branch". It is to say that the development process can be done on three independent ways. And here we introduce a new concept of development process as PSI "Ψ" (Figure 3). Any application development by using PSI process is based on four branches (three parallel branches and one common branch).

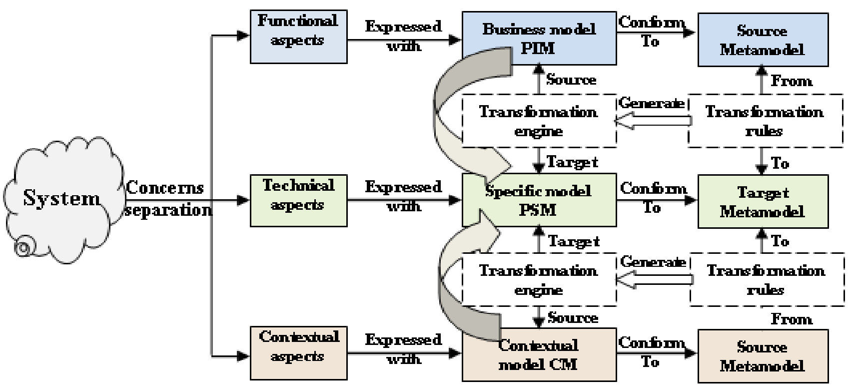

Figure 2: Contextual aspects separation.

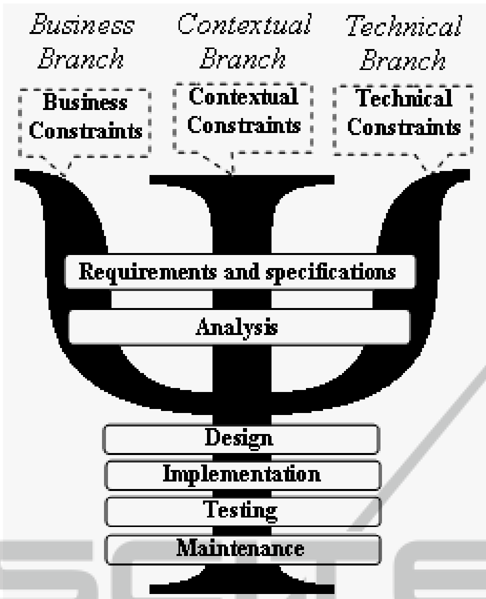

Figure 3: Development process as PSI "Ψ".

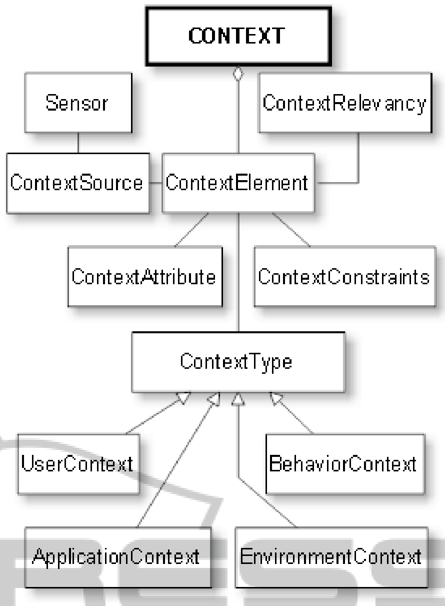

Figure 4: The context metamodel.

The three parallel branches are:

- Business Branch: regroups all basic functionalities of the system,
- Contextual Branch: studies and represents only the contextual constraints,
- Technical Branch: describes the specific chosen platform.

All these branches nearly have the same steps (Requirements, Specifications, Analysis) but according to each kind of constraints. They are independent and they can be done simultaneously.

The common branch corresponds to the phase of realization. This phase begins as soon as the previous three branches are complete and it consists on several steps like: Design, implementation, coding and maintenance.

3.2 Context Modeling

Context modeling approaches are as follows (Strang, 2004) : KeyValue Models, Markup Scheme Models, Graphical Models, Object Oriented Models, Logic Based Models and Ontology Based Models. These approaches are classified by the scheme of data structures which are used to exchange contextual information in the respective system.

Before building the contextual model we are going to present our context metamodel (Figure 4).

Our context metamodel is made by using UML (Unified Modeling Language) notation. The proposed metamodel allows designers to produce more consistent and more homogeneous contextual models. This metamodel is described as follows:

The context of use "*CONTEXT*"of a user (or a situation) can be composed by several contextual elements "*ContextElement*". Each of these elements is defined by: the type, the attributes, the constraints, the relevance and the source (sensor).

The type of a contextual element "*ContextType*" is provided by one of the following entities: the user, the application, the environment or the behavior. So each contextual element has one of the following types:

- "*UserContext*" : when information is related to the user,
- "*ApplicationContext*" : if information comes from the application means,
- "*EnvironmentContext*" : if information is provided by the surrounding objects (other than the user and the application),
- "*BehaviorContext*" : its provided by all behavioral iteractions between the three entities (user, application, environment).

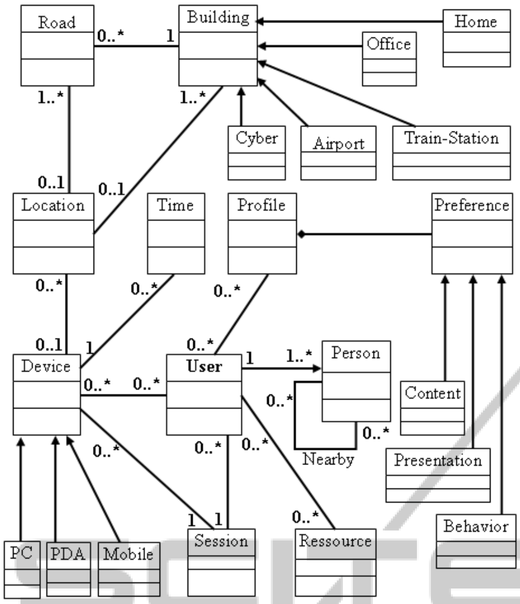

Figure 5: The contextual model.

The attributes "*ContextAttribute*" permit to represent the properties of a contextual element class and define all values that can be attached to instances of this class or an interface.

Constraints "*ContextConstraint*" are defined by a restriction or condition on an element class and they are used to model complex and important information such as domain, table …

The source of the contextual element "*ContextSource*" allows to represent the specific hardware or software used for contextual information sensing. "*ContextSource*" assures the processing of all information captured by the "*Sensor*".

"*ContextRelevancy*" indicates the relevance quantification of an element. It represents the level of importance that each contextual element may have.

Our contextual model (Figure 5) is built using a graphical approach based on UML and it is conform to the context metamodel. This one contains main elements representing the context of use, namely: location, time, device, user's preferences, and surrounding objects.

The current situation of a user may be influenced by several factors and constraints as follows:

- Constraints related to the user himself (Identity, Profile, Behavior, Preferences).

- Constraints related to the application (Software, Hardware, Networks).

- Constraints related to the environment (time, Location, weather, nearby persons, surrounding objects and available resources).

3.3 Integration of the Contextual Model

The integration process of the contextual model in MDA approach is essentially based on merging models operation. This operation combines the elements of business model (PIM) with those of contextual model (CM). Merging operation can be decomposed into four distinct phases: comparison, conformance checking, merging and reconciliation (or restructuring) (Kolovos, 2006). The result of this operation will be a merged model (MM) which will be transformed to a specific model (PSM) according the technical specifications of a chosen platform.
The integration process has five phases: comparison, conformance checking, merging, restructuring and transformation.

In comparison phase all correspondences between equivalent elements of the source models (PIM+CM) are identified. It is a phase of "matching" elements in pairs. In EML (Epsilon Merging Language) (Kolovos, 2008), match-rules are used in order to compare elemets and to decide if they match or no. The decision has a boolean type and it will be used in merging phase. This will eliminate duplicate elements in the merged model.

In conformance checking phase all identified elements during the comparison phase (matching) are tested and verified. They are examined for conformance with each other. This phase is also performed using match-rules that decide whether compared elements are conform or not.

The merging phase is the phase of union and combination of source elements for creating target elements according to the two decisions (match and conform) from previous phases. Here, there are two possible approaches, and this, according to the result of matching. Indeed, if the compared elements are "full correspondent" then they are unified as a single element in the merged model by using merge-rules. And if these elements are considered "not correspondent" then they are propaged distincly in the merged model by using transform-rules.

The target model, obtained after the merging phase, may contain inconsistencies or anomalies. These ones must be cleaned or fixed in the restructuring phase to make a valid model.

The last step in the integration process of contextual model is to transform the merged model to a specific model according to the technical specifications of the chosen platform.

4 CONCLUSIONS

In this study, we presented a new vision of MDA approach which is able to introduce the context of use in its life cycle and, consequently, produce context-aware applications in pervasive environment characterized by ubiquitous information systems. Our proposal goes through three steps. The first is to separate contextual aspects of a system from other ones (business and technical). This separation is based on a new introduced concept, namely the 3TUP process. The second step aims to represent contextual information as a model. To construct the contextual model, we used a graphical approach based on UML and we proposed a context metamodel which can be used by other researchers. The final step proposes a method to integrate the contextual model in the development process of MDA approach, and this, with merging models. At the end we have introduced a new idea for creating an extended UML or an UML profile especially for the domain of context of use.

As perspective we try to present an UML extension for representing the context of use. An extension of UML can be described by an extensisbility mechanism that should contain: Stereotypes, tagged values, constraints and interface. Our future goal consists on creating a stereotype for representing the relevancy of a contextual element. This stereotype will be described by several tagged values (destined for the measurement of relevancy) and some constraints (destined to specify the conditions of using). This extension is based on UML notation and it permits obtaining a specific graphic representation of a contextual situation. We have to extend UML so that it permits us to represent all of the conceptual elements that can influence the current situation of a user. Each contextual element should be able to be represented by thus extension of UML. As perspectives, we expect to continue this study by constructing a contextual mapping. This will include new transformation rules to describe the context of use and that will guide the transition from PIM (including context) to PSM. Our vision extends to the implementation of information systems that will be "contextually parameterized ". Applications must include variables representing all possible contextual information that will be introduced by constraints and profiles of user's current situation.

REFERENCES

Chaari, T., Laforest, F., Flory, A., 2005. «Adaptation des applications au contexte en utilisant les services web». *Proceedings of the 2nd French-speaking conference on mobility and ubiquity computing, ACM International Conference Proceeding Series, Vol. 120 pp 111-118, Grenoble, France*.

Chen, G., Kotz, D., 2000. «A survey of context-aware mobile computing research». *Dartmouth Computer Science. Technical Report TR2000-381*.

Dey, A.K., Abowd, G.D., Salber, D., 2001. «A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications». *Human-Computer Interaction vol.16, 2001, pp 97-166*.

Kolovos, D.S, Paige, R.F., Polack, F.A.C., 2006. «Merging Models with the Epsilon Merging Language (EML) ». *In Proc. ACM/IEEE 9th International Conference on MDE Languages and Systems (Models/UML 2006), Italy*.

Kolovos, D.S, Paig,e R.F., Rose, L.M., Polack, F.A.C., 2008. «Epsilon». *Epsilon Book From Department of Computer Science, University of York, UK*.

OMG., 2003. «MDA Guide V1.0.1». www.omg.org

OMG ., 2007. «Unified Modeling Language (OMG UML), Superstructure, V2.1.2». www.omg.org/spec/UML/2.1.2/Superstructure/PDF

Ou, S., Georgalas, N., Azmoodeh, M., Yang, K., Sun, X., 2006. «A Model Driven Integration Architecture for Ontology-Based Context Modelling and CAA Development» . *Rensink and Warmer (Eds.): ECMDA-FA 2006, LNCS 4066, pp. 188 – 197*.

Strang, T., Linnhoff-Popien, C., 2004. «A context modelling survey». *The 6th international conference on ubiquitous computing, workshop on advanced context modeling, reasoning and management as part of ubicomp 2004, , pp 33-40*.

Vale, S., Hammoudi, S., 2008. «Towards Context Independence in Distributed context-aware applications by the Model Driven Approach». *ACM SIPE'08, Sorrento, Italy*.