International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:2, 2014

Proposal for a Generic Context Metamodel

Jaouadi Imen, Ben Djemaa Raoudha, Ben Abdallah Hanene

***Abstract*—**The access to relevant information that is adapted to user's needs, preferences and environment is a challenge in many applications running. That causes an appearance of context-aware systems. To facilitate the development of this class of applications, it is necessary that these applications share a common context metamodel. In this article, we will present our context metamodel that is defined using the OMG Meta Object facility (MOF).This metamodel is based on the analysis and synthesis of context concepts proposed in literature**.**

***Keywords*—**Context, metamodel, Meta Object Facility (MOF), awareness system.

I. Introduction

NOWADAYS new needs in information systems have appeared. The user of application wishes to have information whenever and wherever he is located. This has prompted developers to integrate mobile devices into their applications, creating new information systems called pervasive or ubiquitous. In such systems, an adaptation of the application to a set of parameters (a type of terminal, connection status,..) must be provided to ensure a comfortable use. All these parameters are a particular context of use. In different contexts, users access the same data and the same services but can receive different responses. These systems, said context-sensitive, are able to, on the one hand, provide the personalized and relevant information and, on the other hand, adapt themselves to the variation of the conditions' execution descended from ubiquitous computing.

The design and development of context-aware applications is particularly complex. The context acquisition is not an easy process. Context information which can be acquired from heterogeneous and distributed sources (sensors, files, applications) may be dynamic and may require an additional interpretation in order to be meaningful for an application. So, to facilitate the development of such applications, it is necessary to provide a metamodel of a generic context that is dynamic, manageable by different applications.

The rest of the paper will be structured as follow: the Section II will present context metamodels proposed in the literature. In Section III, we discuss theses metamodels. Then, we propose our context metamodel in Section IV. Section V illustrates its usage through a case study. Finally, we conclude and outline our future works.

II. State of Art

A context metamodel is defined as the semantics of the key concepts that can be used to define the context. This section includes an overview of the main metamodels of contexts as proposed by researchers in the literature. A challenge, in the context modeling, is to identify what are the concepts that must be considered and how they are connected. We examine these metamodels based on the following characteristics :<entity/property/association>. We use "-" to denote a feature that was not supported by a metamodel.

Henricksen and Indulska in [6] propose a graphical modeling notation called Context Modeling Language (CML), conceived as an extension to the Object Role modeling. In their work the formality of models is considered, diverse context sources are addressed, validity and quality of context entities are provided, and dependencies on context fact types are regarded (Table I). The approach is slightly hindered by the absence of a context modeling editor. Moreover, CML isn't widely used by developers.

Farias and Leite [4] propose a context metamodel formally described using the specification Meta Object Facility (MOF) [8] to allow a precise syntax and an abstract representation common to all the models that are developed. The authors provide a metamodel independent of the domain application and mainly based on the work of Henricksen [6] (Table II). However, it does not provide height level abstraction elements to express conceptual characteristics. For instance, it does not provide concepts such as task, focus, quality, etc.

The UML-based Context Modeling Profile (CMP) as defined by Simons and Wirtz [9] allows modeling the context for mobile distributed systems. UML stereotypes have been defined for the context modeling domain and Object Constraint Language (OCL) constraints are enforced to ensure the validity of the models. For a further definition of the types of information, the author has added modeling access rights of the context elements and the degree of validity of context information (Table III). However, Simons is only interested in the context of the user. In addition, he proposed a profile that is not generic and that responds to the needs of case study systems meeting. The approach benefits from the use of the widely accepted UML, since the CMP can be used in various UML tools. Despite that fact, these tools do not provide a standard way to access model stereotypes and enforce constraints.

In [1], the authors propose an approach MDD (Model Driven Development) to model context-aware applications independent of the platform. However, it is focused on the contextual elements that allow the collection of context information and the identification of the context states that are relevant to the adaptation of a given application. Thus, it is not based on another work (Table IV). It proposes a new model and not an extension of existing models. For example, it did not provide concepts such as temporal aspects of context [6], [4], access rights [9].

Jaouadi Imen is with Faculté des Sciences Economiques et de Gestion, Tunisia (e-mail: jaouadiiimen@gmail.com)

TABLE I
APPROACH PROPOSED BY HENRICKSEN [6]

Approach	Entity type	Property of entity	Association type	Property of association
Henricksen	generic	generic	• static • dynamic (derived, profiled, sensed)	• multiplicity (simple, collection, alternative) • temporal constraint • dependency constraint • quality

TABLE II
APPROACH PROPOSED BY FARIAS [4]

Approach	Entity type	Property of entity	Association type	Property of association
Farias	generic	history of values	-static -dynamic (profiled, sensed, derived)	-multiplicity (simple, collection, alternative) -temporal constraint (relative interval, fixed interval) -equivalence group -dependency constraint

TABLE III
APPROACH PROPOSED BY SIMONS [9]

Approach	Entity type	Property of entity	Association type	Property of association
Simons	generic	-	-sensed, -derive -user provided	-access association: (owner, restricted, group, all) -validity : (permanent, infrequent, frequent, volatile) -derivation rules

TABLE IV
APPROACH PROPOSED BY AYED [1]

Approach	Entity type	Property of entity	Association type	Property of association
Ayed	generic	-collection process -quality -context state <context type, operator, value	-collection -alternative	-

TABLE V
APPROACH PROPOSED BY VIEIRA [10]

Approach	Entity type	Property of entity	Association type	Property of association
Vieira	-who (identity) -what (activity) -where (location) -when (time) -why (motivation)	focus <agent, task>	generic	-acquisition type: (sensed, profiled, user defined, derived, queried) -validity (permanent, infrequent, frequent, volatile) -transformation process -relevance constraint <conditions, actions>

TABLE VI
APPROACH PROPOSED BY MORFEO PROJECT [7]

Approach	Entity type	Property of entity	Association type	Property of association
Morfeo project	• atomic • composite	• source • quality • validity • measurement unit • timestamp • dependency constraints	-	-

TABLE VII
APPROACH PROPOSED BY HACHANI [5]

Approach	Entity type	Property of entity	Association type	Property of association
Hachani	• device / user / environnement • atomic/ composite	generic	-	-

TABLE VIII
EXAMPLE RULES

Rule number	Conditions	Actions
Rule1	if value activity=driving	send SMS "please stop the car as soon as possible you may have an epileptic seizure
Rule 2	if heart rate > threshold	lunched alarm
Rule 3	if (caregiver status="on call"or caregiver statut = "emergency only")	situation of caregiver is available
Rule 4	If distance (location caregiver coordinates, location patient coordinates) <100	situation caregiver= within range

In [7], the authors have proposed a metamodel of context which is based mainly on the definition of Dey [3]. A further definition of the types of information, the quality, the validity and the right access to information are provided in this metamodel (Table VI). However, the proposed model is generic but is not completed. It has not represented equally important concepts to build more adaptive systems as the focus and the types of association which are defined by the work discussed above.

Vieira et al. [10] present a domain-independent context metamodel, which guides a context modeling in different applications. This metamodel has added new concepts such as Focus and the level of relevance that are very important to build more systems that are adaptive, usable and friendly (Table V). However, this model does not include concepts already proposed in the literature such as access rights [9], quality of context information [1], [6]. In addition, this metamodel does not conform to MOF.

In 2009, Hachani [5] proposed a context metamodel that respond only to the needs related to the language and interaction device. Thus, in this metamodel, the authors define only the context information types and modeled the context properties in a generic manner (Table VII).

III. Synthesis

Focusing on state of art (Section II), we can conclude that context metamodel must consist of one or more entities that represent elements of the contexts that are considered relevant to the interaction between the user and the application. In the literature, all authors have presented in their models the concept of entity (Entity), but they used the different terminology as ContextItem in [9]. Hachani et al. [5] have used the concept ElementContext to describe the entity. Each entity must be described by properties (Property). Most metamodels proposed in the literature have considered this concept in their metamodels, but some of them presented a generic property [5], [6] and others have considered entities properties in their context metamodel. The entities can be atomic or composite. For example, the position can be characterized by three properties width, height, and length. Therefore, several authors [5], [7], [10] have distinguished between two types of entities in their metamodels: atomic or composite.

The context is dependent of the application. Indeed, information can be considered as part of the context in one area and not in another. In literature, several metamodels are restricted to narrow classes of context. In particular, there are authors who represent only the sensed context information and its derivation [1]. Thus, there are authors who proposed metamodels that respond only to specific needs. In [9], Simons proposed a metamodel that only responds to the meeting system requirements. Hachani et al. [5] have proposed a metamodel that meets only to language and interaction device. Therefore, a generic context metamodel is necessary to captures various types of context information and to be used by different applications in different fields. For this reason, we suggest to construct a generic context metamodel formed by a generic entity characterized by generic properties.

A context metamodel can be used on different platforms using different technologies. For this, we must have a context metamodel compliant to MOF (Meta Object Facility) to ensure the coherence between the different representations of context used by applications. According to the study that we did, several authors have proposed context metamodels not conform to MOF [1], [10] and others have proposed only the concepts of context and not graphical models [6].

Each model is formed by associations that connect entities together. Since a generic context, metamodel must be able to express the concept of associations. However, in the literature there are authors who didn't represent any association in their metamodels [5]. The context information can be characterized as static or dynamic. The static context information describes aspects of a pervasive system that are invariant; for example, date of birth. Dynamic information describes the information that changes over time. Therefore, the association links between entities can be of two types either static or dynamic. Most studies have distinguished between these two types of associations in their metamodels [4], [6]. Others have represented the dynamic associations describing only the period of validity of association [7], [9]. Several distinguished the subtypes of dynamic information. In [4], [6], [10], the authors have distinguished three dynamic associations classes: sensed, profiled and derived that represent the source of context information. Other authors have described a generic information source [7]. In fact, a global context metamodel must present all these types (profiled, user defined, sensed and derived) as subclasses of the dynamic association representing the source of context information.

In addition, each association must be described by a cardinality that represents the occurrences number of the entity participation that owns the association. Based on the study of metamodels proposed in the literature (section II), we find that the authors [1], [4], [6] agreed on three types of the association end: simple, collection et al.ternative. We suggest representing these three types of multiplicity in our context metamodel to describe the end of the association.

In pervasive information systems, a context can be dynamically changed if other information change over time i.e a change in an association may cause a change in other associations. However, the dependence on associations can exist independently of derived associations. Although this concept is important in context-aware systems for updated and relevant information, this dependency constraint on the association is proposed only in the literature by Henricksen [6]. For this reason, we need to model in our model the dependence on the associations.

Derived information can be inferred from the stories of other context elements. For example, the activity of a person can be inferred from the location of a person and the stories of his previous activities in the past. Therefore, a context metamodel should be able to represent the historical values of each context element. In the literature, only Farias et al. [4] proposed a solution in their metamodel for historical contexts.

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:2, 2014

In our metamodel, we then represent the historical properties values of an entity specifying for each historical value the time in which the context information is acquired (Timestamp).

From the study of the proposed context metamodel, we also find that most authors have presented the temporal aspect of context. Some are presented by defining temporal constraints associations [4], [6]. These constraints indicate a valid interval of time for the use of relevant contextual information. Other authors indicating the validity period of contextual information value [7], [9], [10]. They distinguished four types of validity information: permanent, frequent, infrequent and volatile. Only in [4], the authors have distinguished two types of temporal constraints in their metamodel: Temporal relative Interval and Temporal Fixed Interval. However, the authors in [4] have associated the temporal constraints at the ends of associations (AssociationEnd) whereas the temporal aspect represents the entire dynamic association and not just the cardinality. Indeed, in our metamodel we choose temporal constraints to represent the temporal aspect of contextual information. Each dynamic association may have one or several constraints. Thus, we also represent the two types of temporal constraints that are added by Farias et al. (FixedInterval and RelativeInterval).

In addition, it is essential that applications have a context-sensitive means by which to judge the reliability of information. For this reason, we need to incorporate certain measures of the quality of sensed information in our context metamodel. Types of parameters are dependent on the nature of the association. For example, the quality of information of the user's location can be characterized by its accuracy as measured by the standard error of the system of location. In our metamodel, sensed association may be annotated with one or more quality parameters. Some authors add the quality of contextual information in the literature [1], [6], [7].

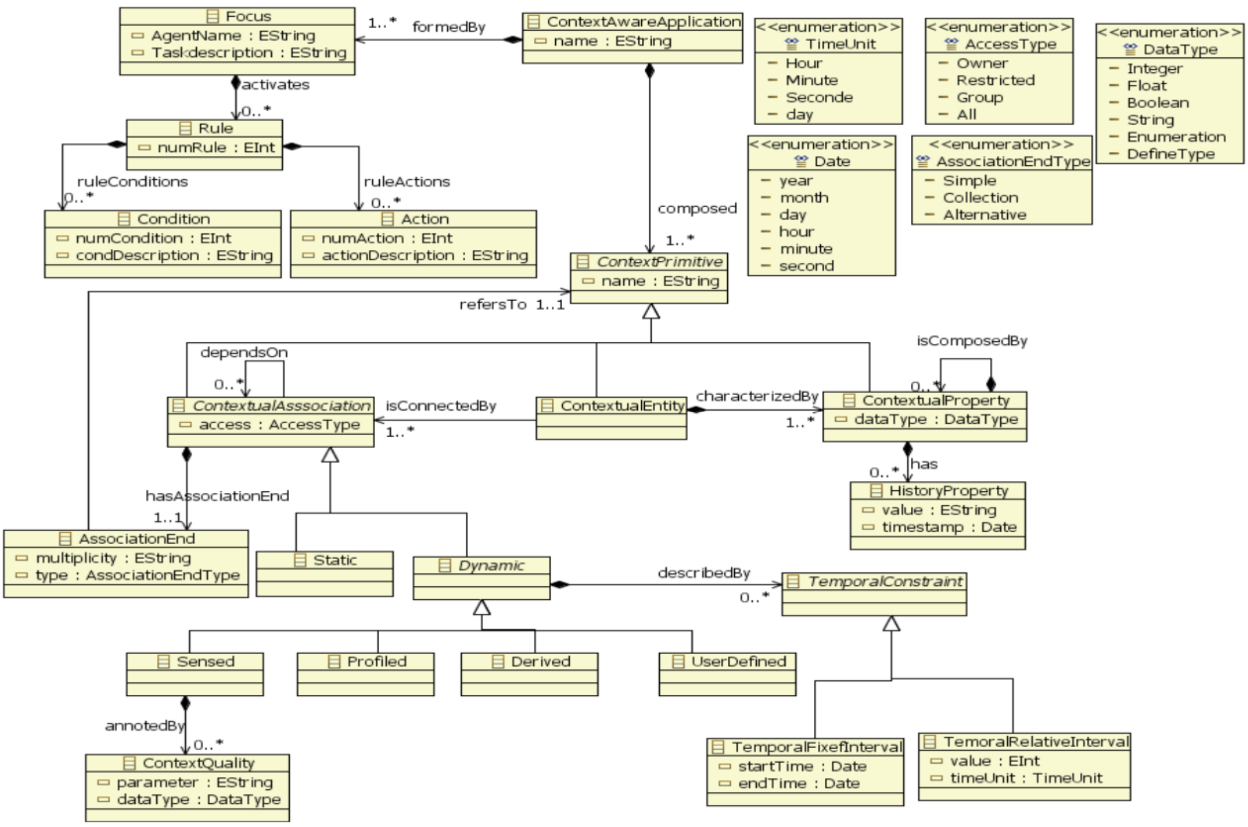

Fig. 1 Our context metamodel designed by eclipse EMF editor

The user context can contain personal information; hence privacy issues have to be regarded. So a context metamodel should allow modeling the access rights to contextual elements. This concept is added only by Simons [9] where he presented four types of access to contextual information: owner, restricted, et al. groups. These types must be represented in our metamodel by an attribute that describes the rights access for associations.

Modeling the focus of context is very important in the context-sensitive applications. It determines which primitive contexts are being considered when the current context occurs dynamically. In fact, the context based on the Focus changes dynamically. However, this concept is indicated only in [10]. In our metamodel we represent the context which is composed of a set of focus. Each Focus is composed of a set of rules.

IV. Our Context Metamodel

Based on the requirements mentioned in Section III, we propose a metamodel compliant to MOF shown in Fig. 1.

A context-aware application is formed by one or more *Focus.* It describe by an agent who interact with the application and the task to execute. A set of rules defines a *Focu*s. A *Rule* consists of a set of conditions and actions. In addition, a context-aware application composed of several context primitives that represent a super class of Contextual Entity, contextual property, and contextual association. *ContextualEntity* correspond to physical or conceptual objects from which contextual information is captured such as: person, device and places. A contextualEntity is characterized by at least one *ContextualProperty*. Each property is described by a *datatype*. The reflexive association *isComposedBy* represents an attribute that can be atomic or composed of other properties. The class *HistoryProperty* represents the historical values of a property that can assume during its lifetime. The attribute *Timestamp* indicates the time during which the value has been stored. The *ContextualAssociation* is used to define the relationships between entities and properties and the relationships among entities themselves. Each association is described by a name and type of access that can be: *owner*, *restricted, group* or *all*. *Owner* is used to model the elements of the private context such as the information on the credit card. The associations applied with a *restricted* access type indicate a user-dependent access. While *Group* denotes the access to members of a group; *All* attributes indicates an unrestricted access. The reflexive relationship *dependsOn* indicates that an association may be dependent on one or more associations. The class *AssociationEnd* represents the cardinality of information. It has three types. An association is *simple* if each entity does not participate more than once as an owner of the association such as the name of a person. The *collection* association can represent an entity that may be simultaneously associated with multiple attribute values and/or other entities; for example one person can work with several others. The alternative association which indicates a collection of mutually exclusive values for example; a channel requires only one device associated therewith. Each *associationEnd* refers to only one *ContextPrimitive.* The associations are classified into two groups: *static* and *dynamic. Static* associations are relationships that remain fixed throughout the life of the entity that has such as date of birth. *Dynamic* are all associations that are not static. They are classified into four types: *sensed* associations represent information about entities obtained through sensors, *derived* associations depend upon one or more associations, *profiled associations* represent information provided by application users by means configurable parameters and *userDefined* association is directly informed by the agent through a dialog interface. A *Sensed* association can be described by parameters of context quality. *TempralConstraint* is an abstract metaclass can be used to define temporal constraints in an association. The metaclass *RelativeInterval* is used to define a time interval based on current time, and the metaClass *FixedInterval* is used to explicit define the valid interval.

V. Case Study

The application of our context metamodel is illustrated using the healthcare epilepsy system has been mentioned in [2]:

> "*Mr. Janssen is an epileptic patient and despite his medications, he still suffers from seizures. Because of his medical condition, Mr. Janssen is unemployed, homebound, and his situation requires constant vigilance to make sure healthcare professionals are alerted of a severe seizure.*
>
> *Recently, Mr. Janssen has been provided with a tele monitoring context-aware application capable of monitoring epileptic patients and providing medical assistance moments before and during an epileptic seizure. Measuring heart rate variability and physical activity, this application predicts seizures and contacts nearby relatives or healthcare professionals automatically. In addition, Mr. Janssen can be informed moments in advance about the seizure, being able to stop ongoing activities, such as driving a car or holding a knife. The aim is to provide Mr. Janssen with both higher levels of safety and independence allowing him to function more freely in society despite his disorder.*"[2]

The first main activity is Focus identification for application. Therefore, we must analyze the possible agents that interact with system and the tasks theses agent could perform. We identify the roles a person can play in this scenario, which are *Epileptic Patient*, and *Caregiver*. An Epileptic Patient represents the persons who suffer from an epilepsy medical condition and need to notify upcoming seizure. The Caregiver represents the persons who have volunteered to assist epileptic patients having an epileptic seizure. Table VIII represents examples rules that can be enabled to Focus <Epileptic Patient receives notifications upcoming seizure>.

Analyzing the application scenario, we identify entities, properties and association types necessary to model the healthcare application. Five contextual entities are identified: the epileptic patient (*EpilepticPatient*) and the caregiver (*CaregiverPatient*) agents that interact with system. A person geographical location (*Geolocation*) which described by the latitude, longitude and the altitude of the person's current location. The device (*Device*) carried by the patient. The detection of epileptic seizure generates seizure alarm (*EpilepticAlarm*) generated by devices attached to the patient's body. These devices collect patient's biosignals (*BioSignalPatient*) in order to predict an epileptic seizure. Caregivers can set their status to (i) onCall, which specifies they are currently available to receive requests for helping patients, (ii) notOnCall, which specifies they are not available for receiving requests for help, (iii) busy, which specifies they are currently receiving requests, but are busy at the moment; (iv) emergencyOnly, which specifies they are currently available for receiving requests only on emergency situations. An epileptic patient may be also doing a potentially hazardous activity, which is captured by a Boolean attribute of the contextual property (*Activity*). The hazardous activity value is derived from the history activities recorded in history property (*HistoryActivities*).

Fig. 2 shows a context model of the healthcare epilepsy system designed by the eclipse EMF editor which ensures that our proposed model is compliant with our metamodel proposed in Fig. 1.

For our context model will be clearer to the reader we decided to present it as a class diagram. Fig. 3 shows a context model designed by the power AMC.

VI. Conclusions

In this paper we first presented and criticize context metamodel already proposed in the literature. Then we proposed a new generic context metamodel based on the weaknesses of the work of others. It represents the historical values of each element of context. In addition, our metamodel represented the temporal aspect and the quality of the context for judging the reliability of information. The development of the proposed metamodel was the first step towards a model approach for the development of context aware adaptive applications. Further, we aim to refine the proposed conceptual metamodel and extend it support the self-adaptive process oriented context aware applications by modeling the dynamic aspect involved in using context.

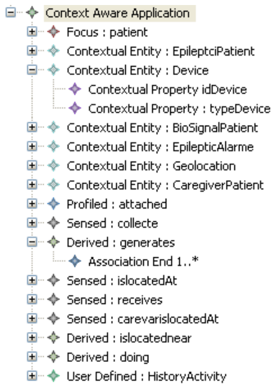

Fig. 2 Context model of the healthcare epilepsy system designed by the eclipse editor EMF

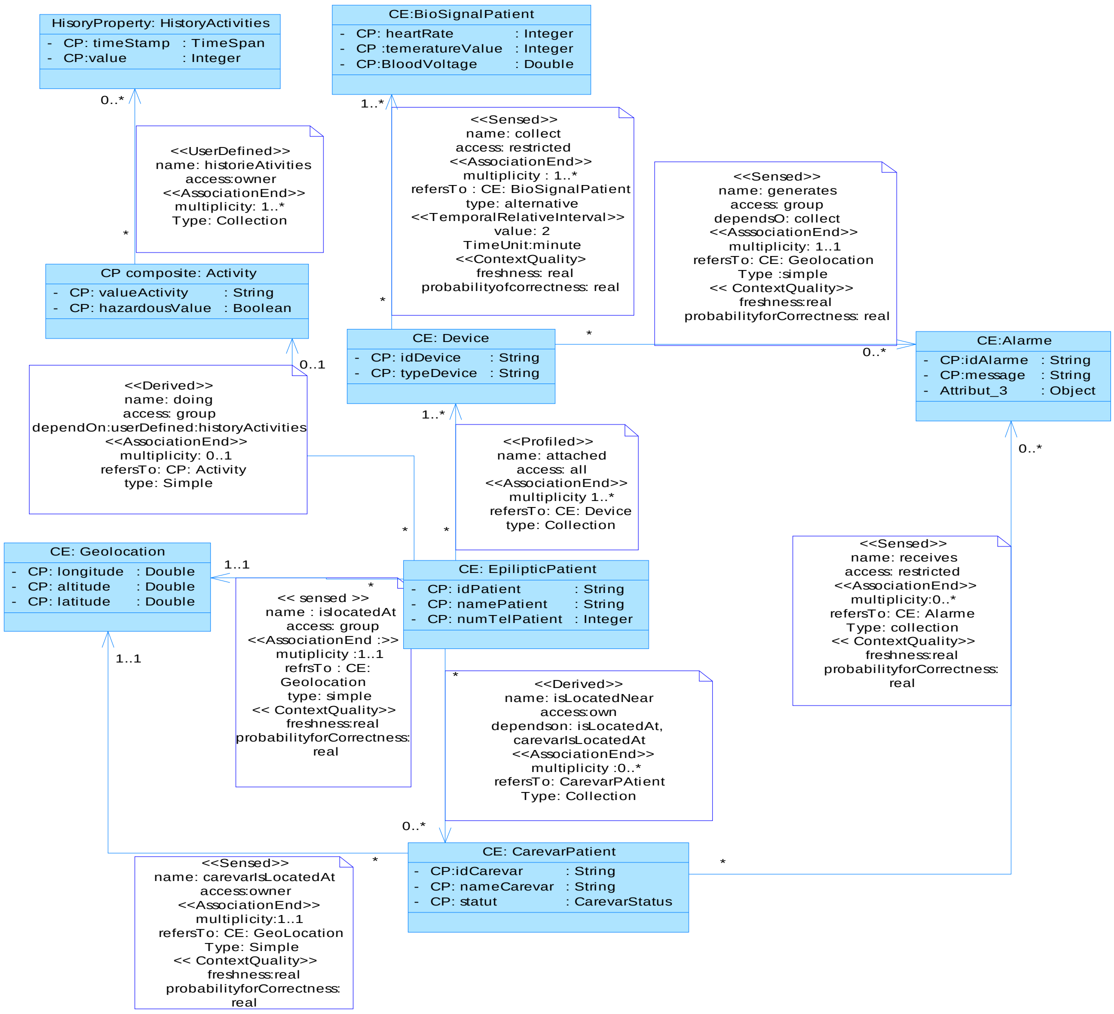

Fig. 3 Context model of the healthcare epilepsy system designed by the PowerAMC

REFERENCES

[1] Ayed and Delanote , “MDD Approach for the Development of Context-Aware Applications,” In: LNAI 4635, 6th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'07), v. 4635, 2007, pp. 15- 28.
[2] P. Costa, “ Architectural support for context aware applications from context models to services platforms,” Enschede, the Netherlands , thesis, 2007.
[3] Dey and Abowd, “Towards a better understanding of context and context-awareness,” Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness, ACM Press, New York, 2000.
[4] Farias, Leite, and Calvi, “A MOF metamodel for the development of context aware mobile applications,” Proc. ACM Symposium on Applied Computing, ACM, New York USA, 2007.
[5] Hachani, sophie Dupuy-Chessa and Angès Front, « Une approche générique pour l'adaptation dynamique des IHM au contexte , » Grenoble university, CNRS, LIG BP72, 38402 Saint Martin d'Héres Cedex, France, 2009.
[6] Henricksen , Indulska and Rakotonirainy,” Modeling Context Information in Pervasive Computing Systems,”.In Proceedings of the First InternationalConference on Pervasive Computing (Pervasive´02), LNCS (2414), p. 167-180, 2002.
[7] MorfeoProject : http://forge.morfeo- project.org/wiki_en/index.php/ Context_MetaModel, accessed 25/1/2012
[8] OMG: Meta Object Facility (MOF) Core Specification, OMG Available Specification, version 2.0. Object Management Group, 2006.
[9] Simons, ”CMP : A UML Context Modeling Profile for Mobile Distributed Systems,” Proceedings of the 40^{th} Hawaii International Conference on System Sciences, 2007.
[10] Vieira, P. Tedesco and Salgado,” Using a Metamodel to design Structural and Behavioral Aspects in Context-sensitive Groupware,” Proceedings of the 2010 14^{th} International Conference on Computer Supported Cooperative Work in Design. 2010 EEE.