Extending UML Class Diagram Notation for the Development of Context-aware Applications

Mohamed Salah Benselim
LabSTIC Laboratory, University of 08 Mai 45, Guelma, Algeria
Email: msbenselim@yahoo.fr

Hassina Seridi-Bouchelaghem
LabGed Laboratory, University of Badji Mokhtar, Annaba, Algeria
Email: h_seridi@yahoo.com

***Abstract*— The development of context-aware applications in a pervasive environment is a very hard work because each situation can be influenced by several factors and features such as users mobility, information heterogeneity and systems distribution. So, new application will be able to adapt its services with the change of context of use and satisfy all users' preferences. In this work we present an UML class diagram extension for representing and modeling context because UML standard notation does not support all aspects of the context of use in an adequate manner. This extension is defined by some extensibility mechanisms and it is presented as a set of new tools for the unified modeling language. The proposed extension is based on UML notation and it permits obtaining a specific graphic representation of a contextual situation. Also, it facilitates the extraction and the modeling of all elements that can influence the current situation of the user. Our proposal consists on creating some stereotypes that are described by several tagged values and some constraints and that can be applied to the contextual model classes. Then we use a class diagram to describe the different types of context and their relationships. A case study is done in the medical domain in which we propose a new contextual model including all new stereotypes by using StarUML software modeling platform.**

***Index Terms*—Software engineering, context-aware, modeling, UML, Class diagram, extensibility mechanisms**

I. Introduction

The development of context-aware software needs specific notations and concepts of modeling language to provide adapted applications and personalized information. Unified modeling language (UML) has been the standard object oriented modeling language used for modeling many aspects of software systems [1]. The main characteristic of UML is a set of extensibility mechanisms (stereotypes, tagged values and constraints) which are used for adding new modeling concepts and notations. So, UML is able to be extended in order to represent all aspects of a specific domain with appropriate notation. For this, we propose an extension of this language that can help us to model context-aware applications with specific notation. Using the new vision of concerns separation cited in [2] we isolate the contextual elements from the global aspects of a system. Each of these elements has many features and constraints related to users, applications or environment. Then we propose some new UML components (stereotypes) to describe and to model all contextual elements. Also we attach some tagged values and constraints to the proposed stereotypes. Using these extensibility mechanisms we can build a complete UML profile for context modeling. We note that context is composed by different elements that have various properties and characteristics and that can be related through different relationships and each contextual element should be able to be represented by thus extension of UML.

II. Related Works

Several studies have investigated the field of context and especially taking into account the contextual changes that surround the application execution and introducing user's preferences. But not many that used extension mechanisms to represent or to model the context of use. In [3] authors presented a model driven development framework for context awareness as an extension of UML. It consists of a domain specific modeling language called CAMEL (Context Awareness ModEling Language) which enables enriching independently defined UML models with the model of context aware behaviors. A visual language for context models in mobile distributed systems has been presented in [4] and the Context Modeling Profile (CMP) has been proposed. CMP is provided as a lightweight extension of UML without modifying the UML metamodel. In [5] a UML profile is proposed in order to model context-aware applications. This profile allows the context-aware applications designed by this profile to be integrated with existing UML-based software. Sheng and Benatallah proposed "Context UML" in order to develop the context-aware Web Services [6]. They modified some metaclasses of UML and so they built an extension of the UML metamodel. The proposed metamodel is then used for modeling context. Some works have introduced the UML extensibility mechanisms in different fields. In [7] authors presented an extension of UML for the representation of Web Service Description Language

doi:10.4304/jetwi.5.1.35-44

(WSDL). The proposed extension provides an UML notation which allows obtaining graphic representation of a Web service and facilitates the automatic generation of WSDL code from an UML diagram. In [8] a set of agents is presented and these agents are based on UML extension and are called AgentUML. There are other works that used UML extension like [9] where the UML modeling language is extended in order to model the structural and dynamic aspects of Multi-agent systems and [10] where authors proposed to extend and to customize the unified modeling language with web design concepts by using the Hypermedia Design Model. In the domain of ontology we have to note that many studies have been led by using the extending UML such as [11] and [12]. In [13] authors propose a new metamodel that capture concept of an extended real-time transaction model by using UML class diagram. The new contribution of our survey is a set of new specific notation of UML class diagram destined to the context-aware applications modeling by using StarUML platform. And because this software platform is extensible for certain domains we can add a new menu system (as profile) to allow users for using easily our proposed UML notations. Using this opportunity we have created a set of extension mechanisms that represent a part of a particular UML profile destined to the context awareness domain. Our proposed profile is called “CAProf” for Context-Aware Profile.

III. Discussion: Context and UML

Future applications have to provide exact results and personalized information and they must be adequate with every variation of user's current situation. Ubiquitous information systems are characterized by many features like mobility, heterogeneity and distribution. These constraints represent the context of use of a situation. The concept of "context" has been defined in [14] as follows: “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 a few elements that constitute the main components of the 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. 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. 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 nomad users. Because of user mobility, not only the time and location are constantly changing but also the surrounding objects (including nearby people) and environmental factors. 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. So, the user is facing a different situation that he is in a new context of use and with which he must be adapted. In previous works [15][16] we have presented a new development approach that can take into account all changes in context of use during the application development process and we proposed a new vision of MDA approach (Model Driven Architecture). Then we have demonstrated how it is possible to isolate the contextual aspects of a system and how to develop the contextual constraints independently from the business constraints of this system and from the technological constraints of the chosen platform [17]. This separation allows us to study all contextual information in a separate branch and without remaking the entire development process because of any changes in the context of use. As result we proposed a contextual model that is conform to a proposed metamodel by using the standard Unified Modeling Language (UML). But in spite of that we were not completely satisfied because the standard UML concepts do not support all aspects of the context of use in an adequate manner. Therefore we decided to find a better solution to represent the context by explicit and appropriate notation. UML is a standard object oriented modeling language offered by the Object Management Group (OMG) and it is used for modeling many aspects of software systems. UML has several features and characteristics, among these features we are interested here by extensibility mechanisms (stereotypes, tagged values and constraints) that are used to introduce new elements for modeling specific domains such as context awareness. So, UML can be extended for modeling contextual aspects that influence the current situation of the user. This extension can be considered as a profile and it is done by adding some new elements to the standard unified modeling language.

IV. Extension of UML class diagram notation

As shown in the previous section, the context of use can be represented by specific UML notation. So we have to extend the unified modeling language by adding new elements that can represent the contextual aspects in appropriate form. But before modeling the context, we must identify and separate all elements that are able to influence the current situation of a user. We note that this situation may be influenced by several factors and constraints as follows [2]:

- Constraints related to the user himself (Identity, Profile, Behavior, Preferences, etc.).
- Constraints related to the application (Software, Hardware, Networks, interaction mode, etc.).
- Constraints related to the environment (Time, Location, Weather, Nearby persons, Surrounding objects, Available resources, etc.).

Each of these constraints must be represented by an adequate notation of UML language. For this we propose an extension of the standard UML that contains stereotypes, tagged values and constraints. A stereotype permits to define a new meaning of an existing UML metamodel element. Tagged values are always attached to a stereotype and their role is to indicate attributes of

the created stereotype. Constraints define some restrictions of semantics for each added new element. Generally, all UML concepts (class, attribute, association, etc.) can be stereotyped.

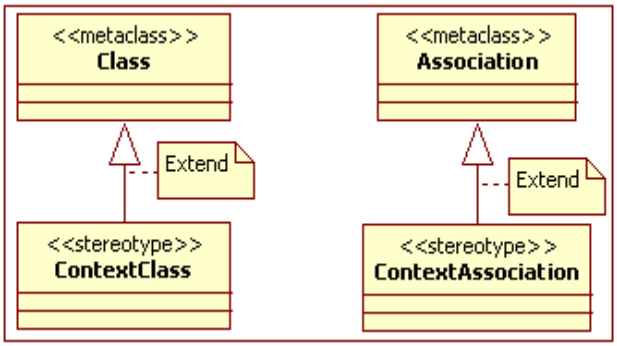

Figure 1. Extension of the "Class" and "Association" UML metaclasses

Fig. 1 presents a general view of the proposed extension. It shows how the UML metaclasses "Class" and "Association" are extended in order to create new UML elements that play a specific role in context modeling. The stereotype "ContextClass" is a new metaclass obtained by extending an existing UML metamodel element and it can be associated with other metaclasses. Its role is to represent the context of use of a system and it is defined by a set of elements "ContextElements" that represent all contextual aspects. Because of mobility of the user and because applications are distributed and distant, each of these contextual elements may have various features and constraints such are previously cited. To assure a better representation of the context in distributed applications development we propose to create specific stereotypes for taking into account all the components of this context and for providing an appropriate model. Fig. 2 presents some stereotypes that are associated with the "ContextElement" stereotype and that define all specific constraints related to the context of a situation. "ContextRelevancy" indicates the relevance quantification of an element, it represents the level of importance that each contextual element may have. 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 stereotypes that are obtained by specialization of "ContextType" stereotype:

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

"ContextAttribute" is a stereotype that permits to represent the properties of a contextual element class and to define all values that can be attached to instances of this class.

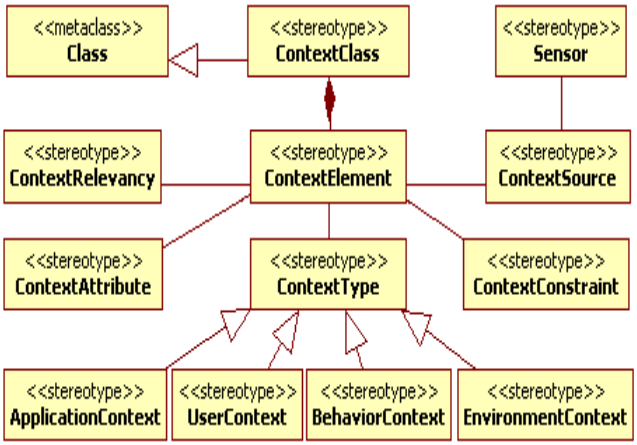

Figure 2. The proposed "Class" stereotypes

The "ContextConstraint" constraints are defined by a restriction or a condition applied on an element class and they are used to model complex and important information such as domain, table, etc. The source of the contextual element "ContextSource" allows representing the specific hardware or software used for contextual information sensing. "ContextSource" assures the processing of all information captured by the "Sensor".

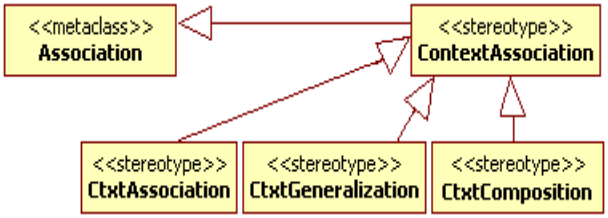

Figure 3. The proposed "Association" stereotypes

Also, the UML metaclass "Association" is stereotyped in order to define new relationships that associate the created "Class" stereotypes (Fig. 3). Each "ContextAssociation" can be specialized in order to have specific relationships such as (but not limited) "CtxtAssociation", "CtxtGeneralization" or "CtxtComposition". These association stereotypes represent respectively a simple association, a generalization (or specialization) or a composition relationship between two "Class" stereotypes and a short description is presented in Table 1.

TABLE I.

DESCRIPTION OF RELATIONSHIPS

Relationship	Base class	Associated classes
IsProvidedFrom	CtxtAssociation	ContextElement, ContextSource
HasRelevancy	CtxtAssociation	ContextElement, ContextRelevancy
IsSpecializedTo	CtxtGeneralization	UserContext, ContextType
IsComposedBy	CtxtComposition	ContextClass, ContextElement

Constraints that are attached to the defined stereotypes are specified in order to extend the UML semantics. They permit to provide some conditions which are used as restrictions in modeling process. These constraints can be represented by using natural language or using OCL (Object Constraint Language). In Table 2 we present a short description of some constraints expressed with natural language.

TABLE II.

DESCRIPTION OF CONSTRAINTS

Stereotype	Description of attached constraint
ContextClass	It must be extended from the UML metaclass "Class"
ContextElement	It must be related to the "ContextClass" class by composition
ContextElement	It must be related to any of "ContextRelevancy", "ContextAttribute", "ContextType", "ContextConstraint" and "ContextSource" classes by simple association
ContextRelevancy	It must be related to the "ContextElement" class by simple association
Sensor	It must be related to the "ContextSource" class by simple association
ApplicationContext	It must be related to the "ContextType" class by generalization
UserContext	It must be related to the "ContextType" class by generalization
BehaviorContext	It must be related to the "ContextType" class by generalization
EnvironmentContext	It must be related to the "ContextType" class by generalization

Other example using OCL, if we need to limit the relevancy application field of a contextual element, we can define a specific constraint that will be attached to a "ContextElement" stereotype. This constraint can be written with OCL language as follows:

```
Context ContextClass inv:
Self.ContextElement->forAll(e : ContextElement |
        e.relevancy <= self.MaxValue
    and e.relevancy >= self.MinValue)
```

TABLE III.

DESCRIPTION OF SPECIFIC TAGGED VALUES

Tagged value	Attached stereotype	Description
LevelOfRelevance=high|medium|low	ContextRelevancy	Degree of importance associated to each contextual element
TypeOfSensor = hardware|software	Sensor	Indicates the type of used tool for information sensing
StateOfUser= sitting|standing|...	UserContext	Indicates the state of the user in the current situation

Tagged values are attached to stereotypes and they are considered as meta-attributes of metaclasses. They can be defined by a given attribute name and its value. They are used to represent and to specify attributes for the defined stereotypes. Table 3 presents an example of three proposed tagged values.

V. IMPLEMENTATION

UML has been the standard software modeling language that provides various concepts and notations for systems modeling. Many specific requirements of certain domains can not be represented using UML standard notation. To solve this problem, UML provides some mechanisms for extending its notation to be adequate with any specific domain. In our study we have used this opportunity to model all contextual requirements of a system. So, we have proposed new modeling elements such as stereotypes, constraints and tagged values that can be used to build a class diagram of a contextual model. To implement our proposal, we used StarUML software modeling platform because it is an extensible platform which supports UML language and provides excellent extensibility, customizability and flexibility [18]. This implementation is a part of a complete UML profile project that we have named “UML CAProf” (UML Context-Aware Profile). UML CAProf will help and assist designers to model the contextual requirements of a system and to develop context-aware applications in ubiquitous information systems. This implementation will be made on the following steps:

- Preparation of UML profile file
- Creating stereotypes
- Creating constraints
- Creating tagged values
- Extending (or customizing) StarUML menu system

A. Preparation of UML Profile File

UML profile is a set of extensibility mechanisms (stereotypes, constraints and tagged values) that are required for a specific software domain or development platform. The first step of our implementation is to prepare the profile document file which will be defined in the XML (eXtended Markup Language) format. This file will contain all profile components and the main structure of the document is as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
<PROFILE version="1.0">
  <HEADER>
      ...
  </HEADER>
  <BODY>
      <STEREOTYPELIST>
         <STEREOTYPE> ... </STEREOTYPE>
      </STEREOTYPELIST>

      <TAGDEFINITIONSETLIST>
         <TAGDEFINITION> ... </TAGDEFINITION>
      </TAGDEFINITIONSETLIST>

      <DATATYPELIST>
         <DATATYPE> ... </DATATYPE>
      </DATATYPELIST>

      <DIAGRAMTYPELIST>
         <DIAGRAMTYPE> ...    </DIAGRAMTYPE>
      </DIAGRAMTYPELIST>

  </BODY>
</PROFILE>
```

B. Creating Stereotypes

According to Fig. 2 and Fig. 3, the list of proposed stereotypes includes: ContextClass, ContextElement, ContextSource, ContextRelevancy, ContextAttribute, ContextConstraint, ContextType, Sensor, ApplicationContext, UserContext, BehaviorContext, EnvironmentContext, ContextAssociation, CtxtAssociation, CtxtGeneralization and CtxtComposition. For each stereotype we indicate the name, a short description and the base UML class of this stereotype. Stereotypes are defined in the document file of CAProf profile by using XML format as follows:

```
<STEREOTYPELIST>
            <STEREOTYPE>
            <NAME>ContextClass</NAME>
            <DESCRIPTION> </DESCRIPTION>
            <BASECLASSES>
                  <BASECLASS>UMLClass</BASECLASS>
            </BASECLASSES>
      </STEREOTYPE>
      <STEREOTYPE>
            <NAME>ContextElement</NAME>
            <DESCRIPTION> </DESCRIPTION>
            <BASECLASSES>
                  <BASECLASS>UMLClass</BASECLASS>
            </BASECLASSES>
      </STEREOTYPE>
```

C. Creating Constraints

All proposed constraints are introduced by using the “Constraint Editor” of StarUML menu System. For each constraint we can specify the constraint body by using the natural language or using OCL language (Fig. 4).

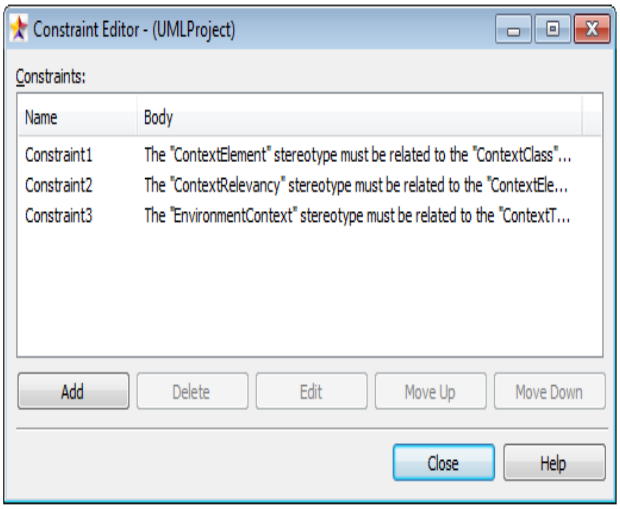

Figure 4. Defining constraints with StarUML Constraint Editor

D. Creating Tagged Values

The “Tagged Value Editor” of StarUML menu System can be personnalized with needed properties. For this we have used the following XML document:

```
<TAGDEFINITIONSETLIST>
  <TAGDEFINITIONSET>
      <NAME>Default</NAME>
      <BASECLASSES>
            <BASECLASS>UMLElement</BASECLASS>
            </BASECLASSES>
            <TAGDEFINITIONLIST>
            <TAGDEFINITION>
                  <NAME>StateOfUser</NAME>
                  <TAGTYPE>Enumeration</TAGTYPE>
                  <DEFAULTDATAVALUE>SITTING</DEFAULTDATAVALUE>
                  <LITERALS>
                  <LITERAL>SITTING</LITERAL>
                  <LITERAL>STANDING</LITERAL>
                  <LITERAL>MOVING</LITERAL>
                  </LITERALS>
            </TAGDEFINITION>
            <TAGDEFINITION>
```

Then the user will be able to introduce the values of system requirements in corresponding fields (Fig. 5).

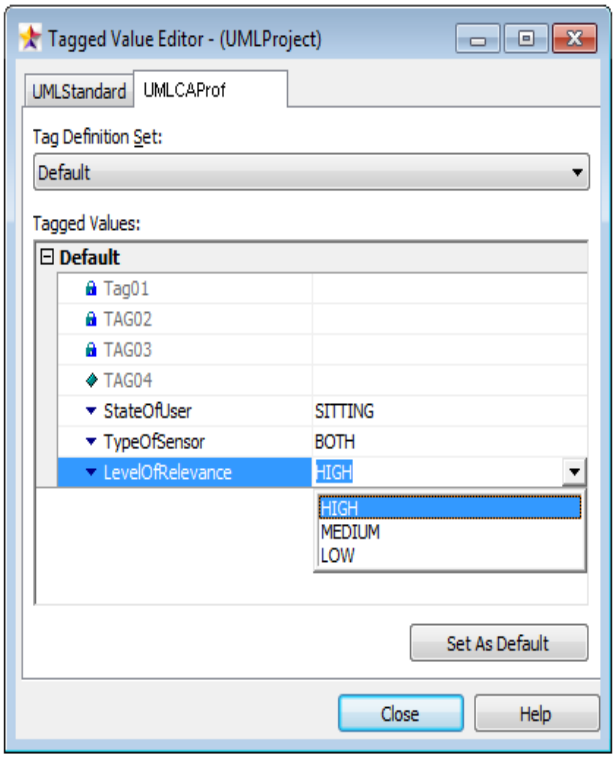

Figure 5. Customized Tagged Value Editor

E. Extending StarUML Menu System

The StarUML menu system can be extended by adding new menu items related to the proposed extensibility mechanisms. This will facilitate the task of the user to call or to use the new concepts. To customize the StarUML menu system with CAProf profile we have to add it in the Profile Manager of StarUML by using the following XML document:

```
<?xml version="1.0" encoding="UTF-8" ?>
<PROFILE version="1.0">
  <HEADER>
      <NAME>UMLCAProf</NAME>
      <DISPLAYNAME>UML Context-Aware Profile</DISPLAYNAME>
      <DESCRIPTION>UML Context-Aware Profile</DESCRIPTION>
      <AUTOINCLUDE>true</AUTOINCLUDE>
  </HEADER>
```

And the user will be able to include the UML Context-Aware Profile in his project at any time he needs to develop context-aware applications (Fig. 6).

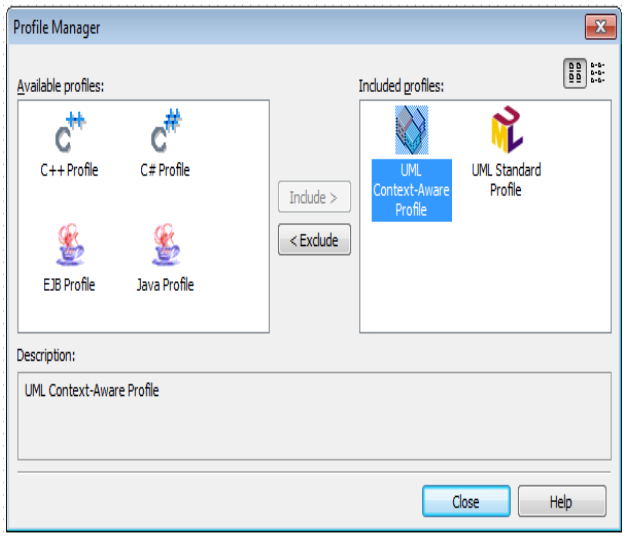

Figure 6. Including CAProf in the StarUML Profile Manager

F. Profile Document File Summary

Below we expose a summary of the document XML file of the UML CAProf profile.

```
<?xml version="1.0" encoding="UTF-8" ?>
<PROFILE version="1.0">

<HEADER>
<NAME>UMLCAProf</NAME>
<DISPLAYNAME>UML Context-Aware
Profile</DISPLAYNAME>
<DESCRIPTION>UML Context-Aware
Profile</DESCRIPTION>
<AUTOINCLUDE>true</AUTOINCLUDE>
</HEADER>

<BODY>

      <STEREOTYPELIST>
          <STEREOTYPE>
<NAME>ContextClass</NAME>
<DESCRIPTION> Context Class </DESCRIPTION>
      <BASECLASSES>

      <BASECLASS>UMLClass</BASECLASS>
              </BASECLASSES>
              </STEREOTYPE>

<STEREOTYPE>
<NAME>ContextElement</NAME>
<DESCRIPTION>Context Element
</DESCRIPTION>
<BASECLASSES>
<BASECLASS>UMLClass</BASECLASS>
              </BASECLASSES>
              </STEREOTYPE>

<STEREOTYPE>
<NAME>ContextSource</NAME>
```

```
<DESCRIPTION> Context Source
</DESCRIPTION>
<BASECLASSES>
      <BASECLASS>UMLClass</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>

<STEREOTYPE>
<NAME>ContextRelevancy</NAME>
<DESCRIPTION> Context Relevancy
</DESCRIPTION>
<BASECLASSES>
      <BASECLASS>UMLClass</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>
            <STEREOTYPE>
<NAME>ContextAttribute</NAME>
<DESCRIPTION> Context Attribute
</DESCRIPTION>
<BASECLASSES>
      <BASECLASS>UMLClass</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>

<STEREOTYPE>
<NAME>ContextConstraint</NAME>
<DESCRIPTION> Context Constraint
</DESCRIPTION>
<BASECLASSES>
      <BASECLASS>UMLClass</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>

<STEREOTYPE>
<NAME>ContextType</NAME>
<DESCRIPTION> Context Type </DESCRIPTION>
<BASECLASSES>
      <BASECLASS>UMLClass</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>

<STEREOTYPE>
<NAME>Sensor</NAME>
<DESCRIPTION> Sensor </DESCRIPTION>
<BASECLASSES>
      <BASECLASS>UMLClass</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>
            <STEREOTYPE>
<NAME>ApplicationContext</NAME>
<DESCRIPTION> Application Context
</DESCRIPTION>
<BASECLASSES>
      <BASECLASS>UMLClass</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>

<STEREOTYPE>
<NAME>UserContext</NAME>
<DESCRIPTION> User Context </DESCRIPTION>
<BASECLASSES>
      <BASECLASS>UMLClass</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>

<STEREOTYPE>
<NAME>BehaviorContext</NAME>
<DESCRIPTION> Behavior Context
</DESCRIPTION>
<BASECLASSES>
      <BASECLASS>UMLClass</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>

<STEREOTYPE>
<NAME>EnvironmentContext</NAME>
<DESCRIPTION> Environment Context
</DESCRIPTION>
<BASECLASSES>
      <BASECLASS>UMLClass</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>

<STEREOTYPE>
<NAME>ContextAssociation</NAME>
<DESCRIPTION> Context Association
</DESCRIPTION>
<BASECLASSES>
<BASECLASS>UMLAssociation</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>

<STEREOTYPE>
<NAME>CtxtAssociation</NAME>
<DESCRIPTION> Context Association
</DESCRIPTION>
<BASECLASSES>
<BASECLASS>UMLAssociation</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>

<STEREOTYPE>
<NAME>CtxtGeneralization</NAME>
<DESCRIPTION> Context Generalization
</DESCRIPTION>
<BASECLASSES>
<BASECLASS>UMLAssociation</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>

<STEREOTYPE>
<NAME>CtxtComposition</NAME>
<DESCRIPTION> Context Composition
</DESCRIPTION>
            <BASECLASSES>
<BASECLASS>UMLAssociation</BASECLASS>
            </BASECLASSES>
            </STEREOTYPE>
            </STEREOTYPELIST>
```

```
<TAGDEFINITIONSETLIST>
    <TAGDEFINITIONSET>
        <NAME>Default</NAME>
        <BASECLASSES>
    <BASECLASS>UMLElement</BASECLASS>
        </BASECLASSES>
    <TAGDEFINITIONLIST>
        <TAGDEFINITION lock="True">
        <NAME>Tag01</NAME>
        <TAGTYPE>String</TAGTYPE>
        </TAGDEFINITION>
        <TAGDEFINITION lock="True">
        <NAME>TAG02</NAME>
        <TAGTYPE>String</TAGTYPE>
        </TAGDEFINITION>
        <TAGDEFINITION lock="True">
        <NAME>TAG03</NAME>
        <TAGTYPE>String</TAGTYPE>
        </TAGDEFINITION>
        <TAGDEFINITION>
        <NAME>TAG04</NAME>
        <TAGTYPE>String</TAGTYPE>
        </TAGDEFINITION>

    <TAGDEFINITION>
    <NAME>StateOfUser</NAME>
    <TAGTYPE>Enumeration</TAGTYPE>

<DEFAULTDATAVALUE>SITTING</DEFAULTD
ATAVALUE>
    <LITERALS>
    <LITERAL>SITTING</LITERAL>
    <LITERAL>STANDING</LITERAL>
    <LITERAL>MOVING</LITERAL>
    </LITERALS>
    </TAGDEFINITION>
    <TAGDEFINITION>
    <NAME>TypeOfSensor</NAME>
    <TAGTYPE>Enumeration</TAGTYPE>

<DEFAULTDATAVALUE>HARDWARE</DEFAU
LTDATAVALUE>
    <LITERALS>
    <LITERAL>HARDWARE</LITERAL>
    <LITERAL>SOFTWARE</LITERAL>
    <LITERAL>BOTH</LITERAL>
    </LITERALS>
    </TAGDEFINITION>
    <TAGDEFINITION>
    <NAME>LevelOfRelevance</NAME>
    <TAGTYPE>Enumeration</TAGTYPE>

<DEFAULTDATAVALUE>MEDIUM</DEFAULT
DATAVALUE>
    <LITERALS>
    <LITERAL>HIGH</LITERAL>
    <LITERAL>MEDIUM</LITERAL>
    <LITERAL>LOW</LITERAL>
    </LITERALS>
    </TAGDEFINITION>
    </TAGDEFINITIONLIST>
    </TAGDEFINITIONSET>
    </TAGDEFINITIONSETLIST>
    <DATATYPELIST>
        <DATATYPE>
        <NAME>Integer</NAME>
        </DATATYPE>
        <DATATYPE>
        <NAME>Float</NAME>
        </DATATYPE>
        <DATATYPE>
        <NAME>Boolean</NAME>
        </DATATYPE>
        <DATATYPE>
        <NAME>String</NAME>
        </DATATYPE>
    </DATATYPELIST>
    <DIAGRAMTYPELIST>
        <DIAGRAMTYPE>
        <NAME>ContextualDiagram</NAME>
        <DISPLAYNAME>Contextual
    Diagram</DISPLAYNAME>
        <BASEDIAGRAM>ClassDiagram</BASEDIA
    GRAM>
        <ICON>contextualDiagram.bmp</ICON>
    </DIAGRAMTYPE>
        </DIAGRAMTYPELIST>
    </BODY>
</PROFILE>
```

VI. CASE STUDY

To demonstrate our approach, an illustrative case study in the medical field (drugs) has been tested. The main goal of this project is to get a new contextual model that is conform to a proposed metamodel including all new stereotypes and by using StarUML software modeling platform. However, we can present a summary of this project. Our survey concerns several situations of using the pharmaceutical products (drugs). Drugs can be used by different users in various situations; that is to say that every situation has its own context of use including all factors and constraints cited in previous sections. Using the proposed UML extensibility mechanisms we can represent the context of use according to each situation. We consider that our system is influenced by the following contextual elements: user, location, device, temperature, network, nearby persons and surrounding objects.

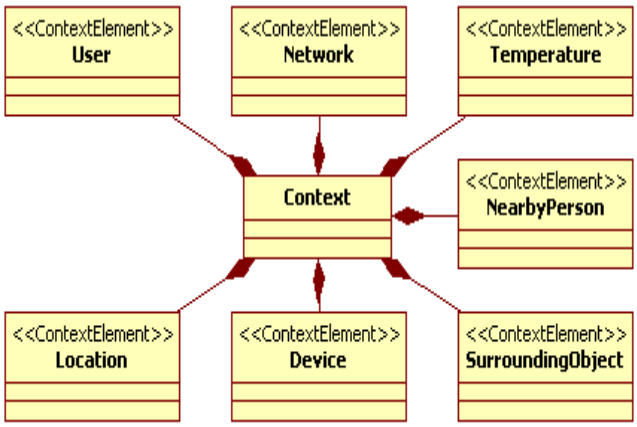

Figure 7. The contextual elements of the system

According to the "IsComposedBy" relationship (Table 1) and the attached constraint of "ContextElement" stereotype (Table 2), these elements compose the context of the system and they can be represented using the proposed stereotype "ContextElement" such as shown in Fig. 7.

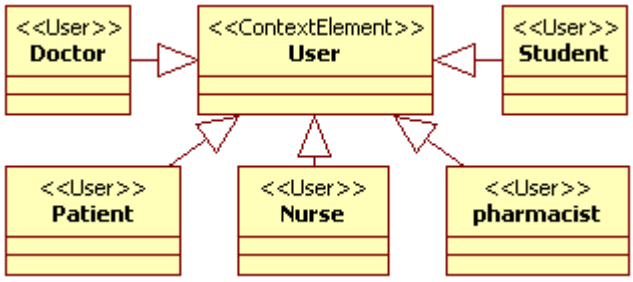

Figure 8. Specialization of users

Obviously contextual elements must have all required information that have been specified as stereotypes in Fig. 2 like attributes, constraints, type, relevancy and source.

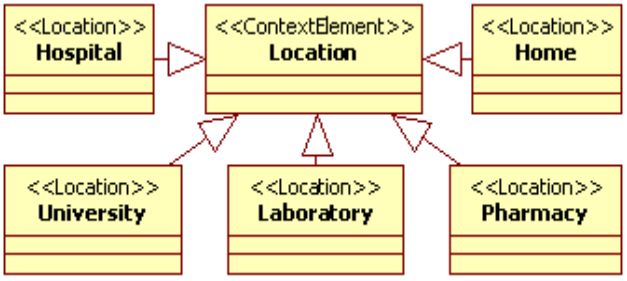

Figure 9. Specialization of locations

Each of the contextual elements can be specialized to represent specific entities. So we have to extract all possible occurrences of these entities that can be implied by the system in different situations. By specialization of "ContextElement" stereotypes we provide an appropriate representation of the specific entities and this will permit us to build a contextual model using an adequate notation. Fig. 8 and Fig. 9 present some specialization examples of contextual elements respectively related to users and locations. Here we note that a person, like a Doctor, can be considered as a user when he uses drugs or he can be considered as a nearby person if he is present in the same location with another user (patient).Because of user's mobility in pervasive environment, the current situation of a user is variable and it changes constantly. So we have to model many various situations (different contextual situations for the same user) of using the system. In our project, a patient (as a user) can be in different situations when using drugs; for example, he can be at the hospital, at home or at the pharmacy. Here we have three contextual situations for the same user. Other example, a doctor (as a user) uses drugs at the hospital, at home, in the laboratory or at the university when teaching. In this case we have four different contextual situations.

TABLE IV.

COMPARISON OF THREE CONTEXTUAL SITUATIONS

Contextual Elements	Contextual Situation 1	Contextual Situation 2	Contextual Situation 3
User	Patient	Patient	Patient
Location	Hospital	Home	pharmacy
Used Device	PDA, Mobile	PDA, PC	PC
Existing Network	LAN	Internet	Internet
Nearby persons	Doctor, nurse	parent	Pharmacist, friend
Surrounding objects	Medical tools	TV, Phone, Printer	Phone, Fax

To illustrate the difference between three contextual situations related to the first example (patient) we collect corresponding information in Table 4. Having these elements and according to the proposed UML extension we can model the system by giving an appropriate class diagram. Needed relationships are summarized in Table 5.

TABLE V.

DESCRIPTION OF NEEDED RELATIONSHIPS

Relationship	Related classes
IsLocatedIn	User, Location
Use	User, Device
HasTime	Device, Time
UseNetwork	Device, Network
IncludePersons	Location, NearbyPerson
ContainObjects	Location, SurroundingObject

Fig. 10 presents a simplified class diagram for modeling the context of our system. In this proposed contextual model we presented all contextual elements that compose the context of use of the system and its relationships.

VII. CONCLUSION AND FUTURE WORKS

Context awareness is a particular domain that claims specific modeling concepts that are able to represent all contextual aspects of a system with appropriate tools. In this paper we proposed an extension of UML class diagram notation that represents a part of an UML profile

for context modeling. This profile is called “CAProf” and it is presented as a set of new elements notation defined by an extended version of the unified modeling language. We created and presented some stereotypes obtained by extending the UML metaclasses "Class" and "Association" by using the StarUML extensible software platform. Then we attached some tagged values and constraints to the proposed stereotypes. In the case study we presented a description of some contextual elements and we proposed a class diagram of a contextual model that can be used in context-aware applications development. As perspective we hope to continue this study while offering a more complete list of extensibility mechanisms related to other UML diagrams (use case, sequence and activity). This will permit us to create a complete UML profile for context modeling. Also we hope to use the proposed extended UML in information systems adaptation and for information personalization.

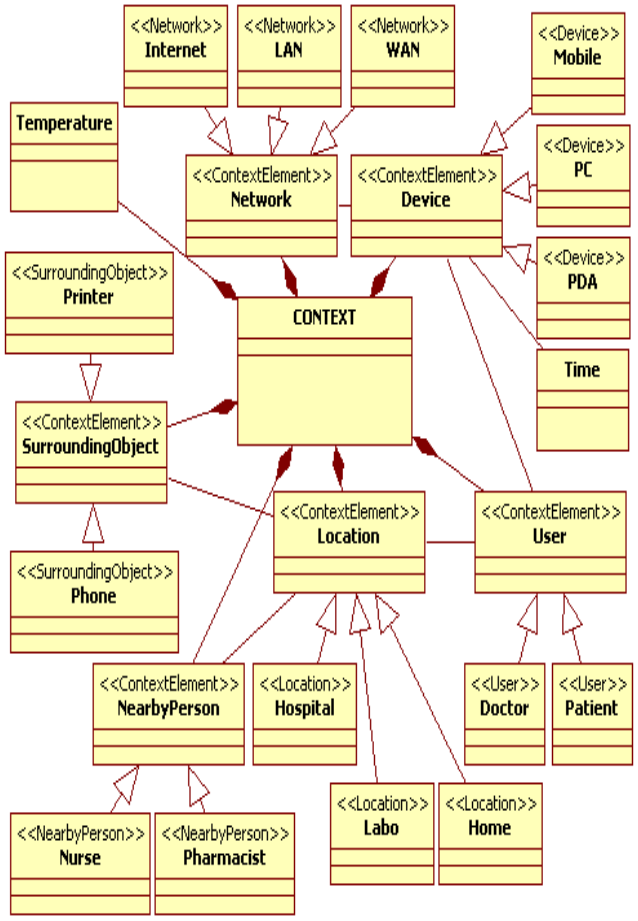

Figure 10. The proposed class diagram

REFERENCES

[1] OMG/UML, “UML 2.1 Superstructure Specification, Document: ptc/06-04-02, 2006.

[2] M.S. Benselim and H. Seridi-Bouchelaghem, “Development of context-aware applications in ubiquitous information systems,” Proc. of the 13th International Conference on Enterprise Information Systems, (ICEIS, June 8-11, 2011, Beijing, China), vol. 3, pp 223-228.

[3] A. Sindico and V. Grassi, “Model driven development of context aware software systems,” International Workshop on Context-Oriented Programming (COP 2009, July 7, 2009, Genova, Italy), pp.1-5.

[4] C. Simons, “CMP: A UML Context Modeling Profile for Mobile Distributed Systems,” In proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS 2007, Jan 3-6, 2007, Hawaii, USA).

[5] D. Ayed, D. Delanote and Y. Berbers, “MDD approach for the development of context-aware applications,” In: Kokinov, B., Richardson, D.C., Roth-Berghofer, T.R., Vieu, L. (eds.) CONTEXT 2007. LNCS (LNAI), vol. 4635, pp. 15–28. Springer, Heidelberg, 2007.

[6] Q.Z. Sheng and B. Benatallah, “ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services,” 4th International Conference on Mobile Business, IEEE Computer Society. Sydney, Australia, 2005.

[7] V. de Castro, E. Marcos and B. Vela, “Representing WSDL with Extended UML,” Revista Columbiana de Computation, vol. 5, 2004.

[8] J. Odell, H.V.D. Parunak and B. Bauer, “Extending UML for Agents,” In Proceedings of AOIS Workshop at AAAI, 2000.

[9] V.T. da Silva and C.J. P. de Lucena, “Extending UML to Model Multi-Agent Systems, ” PUC-Rio Inf.MCC 08/04 March, 2004.

[10] L. Baresi, F. Garzotto and P. Paolini, “Extending UML for modeling web applications,” In Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS 2001, Hawaii, USA), pp 1285 -1294, 2001.

[11] K. Baclawski, M.K. Kokar, P.A. Kogut, L. Hart, J. Smith., W.S. Holmes, J. Letkowski and M.L. Aronson, “Extending UML to Support Ontology Engineering for the Semantic Web,” In: Gogolla, M., Kobryn, C. (eds.) UML 2001. LNCS, vol. 2185, pp. 342–360. Springer, Heidelberg 2001.

[12] S. Brockmans, P. Haase, P. Hitzler and R. Studer, “A Metamodel and UML Profile for Rule-Extended OWL DL Ontologies,” In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 303–316. Springer, Heidelberg 2006.

[13] M. Kaddes, L. Amanton, B. Sadeg, M. Ali, M. Abdouli, R. Bouaziz “RT-ETM: Toward Analysis and Formalizing Transaction and Data Models in Realtime Databases,” International Journal of Web Application, Volume 3, Issue 2, June, 2011, Pages 72-79.

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

[15] M.S. Benselim and H. Seridi-Bouchelaghem, “Contextual adaptation of ubiquitous information systems,” Proc. of the International Conference on Multimedia Computing and Systems, (ICMCS 2009, April 02-04, 2009, Ouarzazate, Morocco), p 135.

[16] M.S. Benselim “Une approche pour le développement d'applications sensibles au contexte,” XXVIIe congress of INFORSID, (May 26-29, 2009, Toulouse, France), pp 479-480.

[17] I. Bouacha, H. Seridi, M.S. Benselim, “Development of context-aware web services using the MDA approach,” International Journal of Web Science (IJWS), Vol. 1, Issue 3, pp 224-241, 2012.

[18] StarUML 5.0 Developer Guide 2005, www.staruml.com/starumldoc/developer-guide