Design Pattern for Pervasive Systems

Fatma Achour, Anis Jedidi, Gargouri Faiez
MIRACL
Multimedia, Information Systems and Advanced Computing Laboratory
Sfax University, Tunisia
E-mails: {fatma.achour@gmail.com, anis.jedidi@isimsf.rnu.tn, faiez.gargouri@isimsf.rnu.tn}

Abstract-**The ubiquitous computing, also named pervasive computing, gathers the characteristics of mobile computing and the techniques of context-awareness which are actually flexible, adaptable, and capable of acting autonomously. However, pervasive computing introduces a variety of software, hardware and users' engineering challenges. We look, while realizing our thesis, for a framework for the design and adaptation of the pervasive system. Firstly, we attempt to allow the software agents to assist the users in manipulating the resources on the Web. Then, we try to manage the interoperability between the resources and their adaptation to the various contexts of use (semantic, network, etc.). Finally, we present the knowledge in an accessible form to both man and machine. In this paper, we present the first phase of our work which resides in conceiving a structure for the design pattern of the pervasive system.**

Keywords - pervasive computing; models; transformation; MDA.

I. INTRODUCTION

The pervasive system combines ubiquity, mobility and context-awareness. The evolution of pervasive computing, fortunately, has opened new horizons to the classical information systems by integrating new technologies and services that may lead to a seamless access to information sources at anytime, anyhow and anywhere. But, for every plus there is a minus. This evolution has brought about new challenges to information modeling. No matter what the challenge is, these systems must be able to be used in various contexts according to the user environment, his profile, the used terminal, etc.

In the literature, several researchers are working on the pervasive systems adaptation. Each researcher offers his own dimensions for the context. In this paper, we intend to realize a design pattern structure useful for the adaptation of any application in a pervasive context. According to Alexander [1], a pattern describes a problem which frequently occurs in an environment as well as a solution that can be adapted to a specific situation.

This paper is organized as follows. Section 2 presents the creation steps of the Pattern Design. Section 3 presents the process to validate the design pattern structure.

II. STATE OF THE ART

Daniel Salber defines the context in [2] as the "Environmental information or context that covers information that is a part of an application operating environment and that can be sensed by the application. This typically includes the location, identity, activity and the state of the people, groups and objects. The context may also be related to places or the computing environment". For such a literature keen on the context model, Held - the oldest model - is deemed to be the simplest key value model to use [2]. After this model, more and more typical models came into sight. The models are often classified by the structure plans of data which are used to describe and transmit the contextual information. In this work, we present four models: CSCP, SOUPA, COMANTO and Activity.

A. CSCP (Comprehensive Structured Context Profiles)

In 2002, CSCP, which is the extension of CC/PP (Composite Capabilities / Preferences Profiles), was proposed by Held who did not define any fixed hierarchy to solve the problems of CC/PP. The context model CSCP represents the profile sessions and is based on RDF; it does not impose any fixed hierarchical structure for the context notion. Thus, it inherits the full flexibility and the expressive power of RDF. The CSCP, indeed, allows the merger of the profile fragments that are dynamically retrieved even from various web sites [3].

- The user profile: is composed of static characteristics (name, first name, etc.) and evolutionary characteristics which are defined by his environment (location, time, etc.) and his preferences
- The device profile: presents the material context (type of device, screen size, etc.) and the software context (operating system, version, etc.).
- The network profile: exposes information about the type of network, its characteristics, etc.
- The session profile: presents the connection from the user to the system.

B. SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications)

The SOUPA project was created in November, 2003. It is part of the work caring for the semantic web which was developed by the "Special Interest Unicomp" group [4]. The objective of this project is to define the ontology written in OWL which supports the pervasive applications. The SOUPA concepts cover themselves with the set of existing ontology vocabularies.

The ontology cited by SOUPA included the Friend Of Friend (FOF), DAML Time, Open Cyc, RCC, COBRA-ONT, MoGATU BDI and Rei ontology policy. In other words, SOUPA contains two distinctive connected subsets of ontology: the SOUPA Core and the SOUPA Extension. On the one hand, the first ontology allows defining the

 ISBN: 978-1-61208-266-0

generic vocabularies for the construction of the pervasive applications [5]. On the other hand, the second one allows defining the additional vocabularies to support a specific type of applications and provides examples of defining new extensions of SOUPA.

In this work, we focus on the SOUPA Core. The SOUPA core ontology includes a set of sub-ontology.

- Person: this ontology defines the vocabulary to describe the typical personal information and profile of a person.
- Policy and Action: the security and privacy are two increasing concerns in the development and deployment of the pervasive computing systems.
- Agent and BDI: during the construction of the pervasive systems, it is useful to present the computer model entities as agents.
- Time: SOUPA defined a set of ontology to model the time and the temporal relations.
- Space: this ontology is designed to support reasoning about the spatial relationships between the different types of geographical spaces and geographical coordinates.
- Event: is the event of the activities that have the same spatial and temporal extensions.

C. *COMANTO (COntext Management oNTOlogy)*

The ontology of context Daidalos, also called COMANTO (COntext management oNTOlogy), was created by Roussaki [6] in 2005. It is the outcome of a hybrid context modeling approach to handle the context objects and the context knowledge. The COMANTO ontology is proposed as a public context semantic vocabulary supporting the efficient reasoning about the contextual concepts (such as users, activities, tools, etc.) and their associations. The ontology classes of the context COMANTO can be described as:

- Person: it is the central entity in the ontology COMANTO. It represents all the human entities and offers the diverse properties of the data type to integrate the user into the related context.
- Place: it is the abstraction of a physical space. It offers a set of data properties that associate a physical location with its symbolic or geographical representation.
- Preferences: in order to model the user, the service, the network and the device preferences.
- Service: it stores the information relevant to the applications taken by the user.
- Legal Entity: This class is mentioned as a representation of the corporate actors involved in the pervasive computing supply chain.
- Device: the class "Device" is an abstract representation of mobile devices.
- Network: the class "Network" contains all the network information.
- Sensors: is another abstract class to get a true representation of the device.

D. *ACTIVITY Model*

The theory of activity is a descriptive tool to help understand the unity of knowledge and activity [7]. It focuses on the practice of the individual and collective work. An activity consists of a subject, an object and an artifact or a tool of mediation. The activity theory is applied to provide a model that covers all the possible contexts in ubiquitous computing. The model of context "Activity" appears in eight main categories. In each category, there are other levels of context. The elements of the context model can be described as follows:

- User: information about the user that the system is interested in.
- Tools: list of the available tools in the public place.
- Rules: norms, social rules and legislations through which the user is related to the others in his community.
- Community: information about the persons around the user who can have an influence on the activity.
- Object: the intention and the objective of the user.
- Time: it is a moment in a particular situation in which an activity takes place.
- The division of labor: the distribution of tasks between the members of the community.
- Result: the result of the object transformation.

III. DESIGN PATTERN STRUCTURE MODELING PROCESS

The various phases of the design pattern structure conception are presented in the Figure below (see Figure 1).

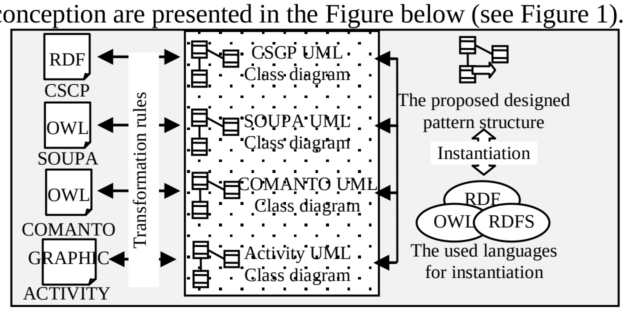

Figure 1. The various steps of the proposal execution.

Firstly, we intend to transform the four contextual models: CSCP, SOUPA, COMANTO and Activity into a UML class diagram. Secondly, we perform a comparative study of the different concepts of each model. Finally, we present a consistent class diagram of these models. These models will be validated by the instances creation in the medical, educational, commercial and archaeological fields.

A. *The transformation of contextual models*

1) The CSCP model transformation: The CSCP transformation of the graphical RDF (Resource Description Framework) model into a valid class diagram is realized according to the following transformation rules .

The "predicate" and "object" are transformed into UML classes,
The "subject" is transformed into association,

The “rdf:bag” is transformed into a composition with a constraint OCL.

The UML class diagram of the CSCP model is shown in the following Figure (see Figure. 2).

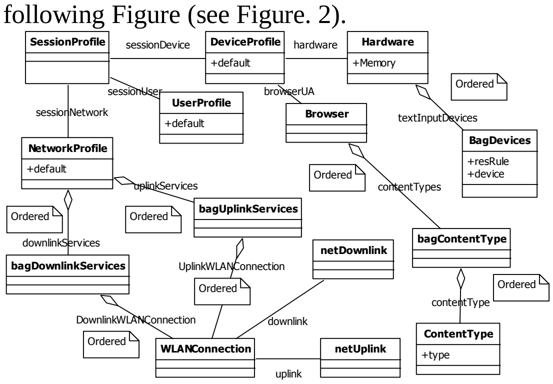

Figure 2. The class diagram of the CSCP model.

2) The SOUPA ontology transformation: The SOUPA core class diagram is achieved through a theoretical study of the different concepts in each ontology. In fact, a set of transformation rules are defined in a manual way to enssure this transformation. The class diagram corresponding to SOUPA ontology is represented in the Figure 3. The transformation rules [8] that allow for the realization of the class diagram represented in Figure 3 are presented as follows:

Owl: Property: is transformed into a UML attribute.
Owl: class: is transformed into a UML class.
Defined by: like “owl: property”, the property “defined by” is transformed into a UML attribute.
Union and intersection: these relations are replaced by the relations of the composition within the OCL constraints.
subClassOf: the property “subClassOf” is transformed into an extended relationship.

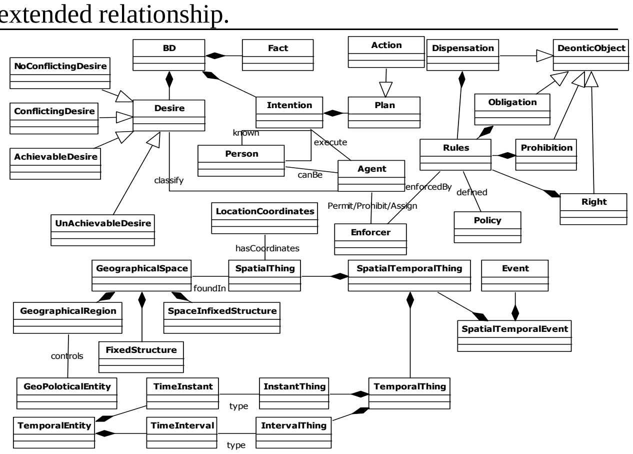

Figure 3. The class diagram of SOUPA Ontology.

We define the following set of OCL constraints from SOUPA Core.

SpatialTemporalEvent=intersect(SpatialTemporalThing, Event)
SpatialTemporalThing=intersect(TemporalThing,SpatialThing)
TemporalThing= Union (InstantThing, IntervalThing)
TemporalEntity= Union (TimeInstant, TimeInterval)
GeographicalSpace= Union(GeographicalRegion, FixedStructure, SpaceInFixedStructure)

3) The COMANTO ontology transformation: The transformation of ontology COMANTO into a corresponding class diagram (see Figure. 4) is carried out according to the following transformation rules [6].

The OWL nodes are transformed into UMLclasses.
Two-way links titled (in reverse) are transformed into a two-way association with two titles.
The unidirectional arcs between the nodes are transformed into a unidirectional association.
The relation “rdfs:subClassOf” is transformed into a heritage relation.
The relation “owl:disjointWith” is transformed into an association with the OCL constraint.

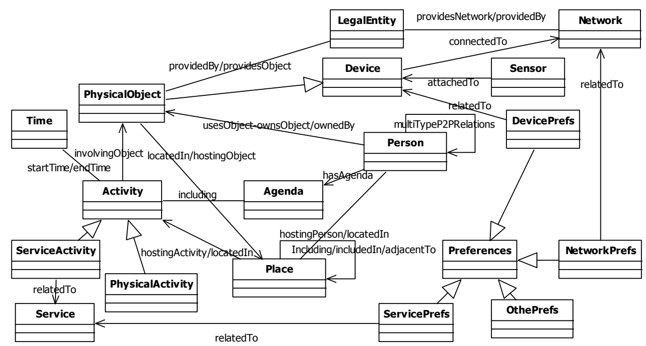

Figure 4. The class diagram of the ontology COMANTO.

4) The ACTIVITY model transformation: The transformation process of the context “Activity” structure into a UML class diagram represents the transformation of a simple graphic model into a UML class Diagram.

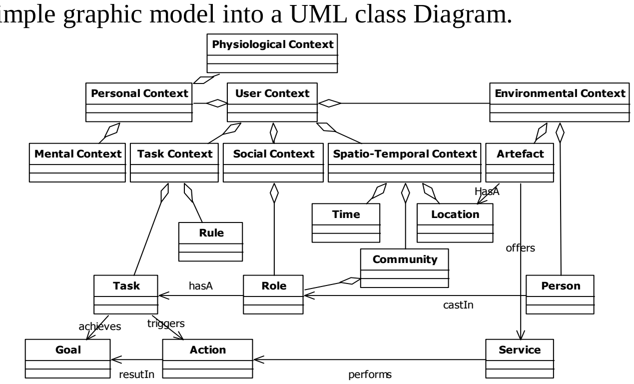

Figure 5. The class diagram of the Activity model.

The result of the Activity model transformation represented in the Figure above (see Figure. 5) is fulfilled through the use of the following three rules.

The nodes are transformed into classes,
The arcs are converted into oriented associations in the UML diagram,
The arcs of the type “partOf” are transformed by the UML

aggregations.

B. The comparative study

The adaptation of the applications to the pervasive system is essentially based on a given model. In this work, we put the stress on the design pattern structure of the adaptation. To unify the existing models, we try to make a structure according to the four suggested models; CSCP, SOUPA, COMANTO and Activity model. This design pattern structure is a UML class diagram that will be turned into an application ontology written in OWL. This ontology will allow for the support system creation to design the pervasive application.

After the theoretical study of the four presented models, we have made a comparison between the different concepts used in each model. We presented in the following table the intersection concepts of selected models (see Table I). The goal is to make an application structure compiled with all of these models. All these concepts will be transformed into a UML class diagram for the design pattern structure. The latter will be used in the phase of adaptation.

TABLE I. THE COMPARATIVE TABLE OF CONCEPTS.

M / C	CSCP	ACTIVITY	COMMANTO	SOUPA
Device	Artifact	Device	Device profile	Agent
Person	Person	Person	User profile	Person
Social	Role (social multi Type)	–	Property : context)	P2P relations known (person)
Rules	Rule	Legal entity	resRule	Rules
Time	Time	Time	–	Time
Place	Location	Place	–	Geographical space
Activity	Task	Activity	–	–
Action	Action	–	–	Action
Network	–	Network	Network profile	–
Desire	–	Preferences	–	BDI:desires
Service	Service	Service	–	–
Location Coordinate	–	Including Included in Adjacent to	–	Location Coordinate

C. The proposed class diagram

After the comparative study between the four models, we can extract the principal classes that can be used in the proposed design pattern structure. Indeed, a class is called principal if it is common among the concepts of the four presented models.

The associations between the proposed classes are presented according to this theoretical study. The result of this study is presented in the next points.

- Agent: this class is used for the presentation of the different actors in a pervasive system. It assembles the human actor and the device actor.
- Persons: present the human actors.
- Device: presents the peripheral devices in a pervasive system.
- Service: presents the services offered by each device.
- Networks: collect the characteristics of the different types of network.
- Location coordinates: present the spatial relation between the different locations in a pervasive system.
- Preferences: present the information profile of the person who realizes the activity in a pervasive system.
- Activity: presents the characteristics of the activity requested by the user.
- Rules: gather the different rules of activity, person and network interacting in a pervasive system.
- Time: presents the characteristics of the temporal and the relation ones of the different things in a pervasive system.
- Locations: represent the characteristics of localizations of the human and mobile devices in a pervasive system.

These classes will be presented in a UML class diagram (see Figure. 6). In a pervasive system, the context is defined by a set of characteristics that are related either to the user (classes; person, preference and role), to the physical environment (classes; location and location coordination), to the network (classes; network and rules), to the activity (classes activity, time and rules), to the devices (class device) or to the service (class services).

A characteristic is represented by a key/value. The attribute specifies the name of the characteristic and its value is given by the same name. For example, a network may be the characteristic of the flow rate. This instance may have the attribute “flow rate” and the value “1 Mbps”. The definition of the representation of the PSI needs to be dynamic and scalable since it is composed of one or more characteristics. These characteristics are not defined in advance, but they are defined by the system designer and according to his needs.

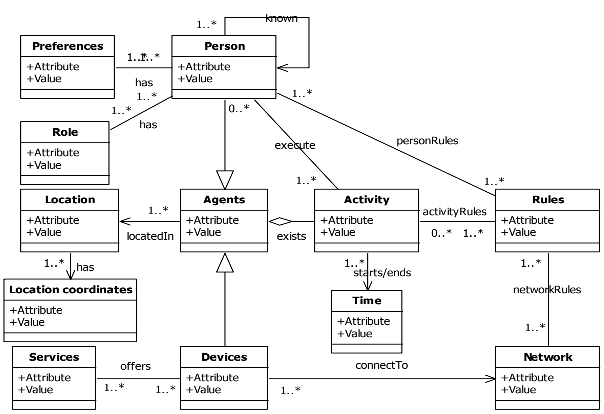

Figure 6. The proposed class diagram.

IV. THE DESIGN PATTERN STRUCTURE VALIDATION

UML is a semi-formal language that allows us to have a better visual clarity of our design pattern structure. However, to describe the instances of this design pattern structure, we have chosen the formalisms RDF /RDFS /OWL. The formal languages are inspired by the logic of description. They are more appropriate to describe the semantics because they provide us with the basic predicates that we can re-use.

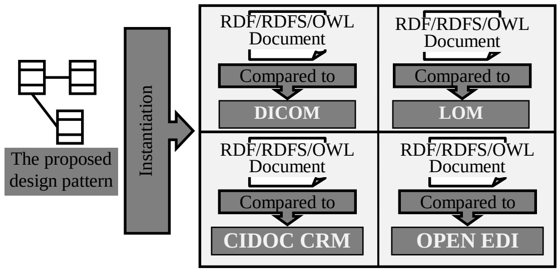

Figure 7. The various steps of validation.

To validate the proposed design pattern structure, we try to apply it to different fields. These instances are created manually and compared with the existing models. We are interested in the validation process of the respective domain ontology of medical, educational, archaeological and commercial field (see Figure 7).

A. The medical field instance

In order to validate the proposed model, we have created an ontology representing an instance of the design pattern structure in the medical field. The classes representing “the mobility” are not represented. The instance created in the medical field will be presented in the next RDF/RDFS/OWL document (see Figure 8). We have compared this instance to the class diagram corresponding to Digital Imaging C0mmunication in Medicine (DICOM).

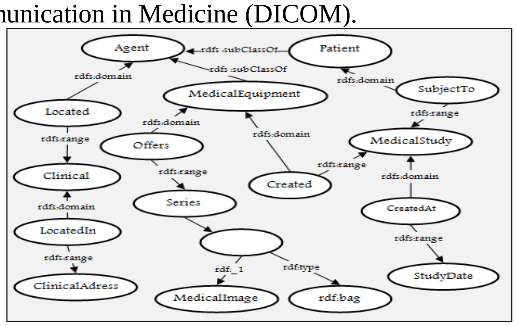

Figure 8. The instance of medical field.

DICOM is a format that defines the methods of connection, transfer and identification of medical data. We found that this instance represents an essential part of this format [9].

After the instance creation, we have created a comparative list of the set of the DICOM classes and the set of instance classes (see Table II). An important number of DICOM concepts are presented in the proposed RDF/RDFS/OWL instance.

TABLE II. THE MEDICAL FIELD COMPARISON.

DICOM	Image	Medical examination	Patient	Medical study	Medical series	Equipment	Terms of Reference	Result	Amendment	Report	Notification content of the study
Instance	*	*	*	*		*	*				

B. The commercial field instance

In order to validate the proposed model, we have created an ontology representing an instance of the design pattern structure in the commercial field. We used the e-commerce open EDI to validate the proposed model.

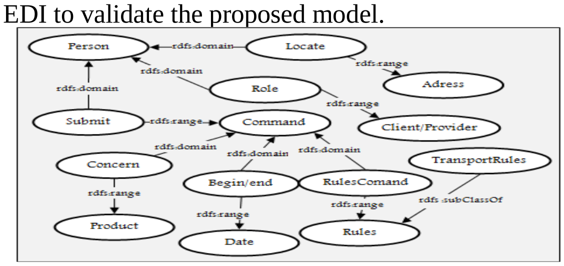

Figure 9. The instance of commercial field.

The open EDI proposes a standard to describe the roles of diverse companies and guarantees that their applications are able to communicate through the EDI messages [10]. To do it, we realized an instance in the commercial field; this instance is presented in Figure 9.

The proposed instance represents a part of the commercial domain. Information in the commercial field has been represented in this instance. The result of the comparison in the commercial field is represented in the following table (see Table III).

TABLE III. THE COMMERCIAL FIELD COMPARISON.

Open EDI	Buyer	Customer contact	Further address	Client	Contact supplier	Supplier	Conditions of Carriage	Header control	Storage location	Locus of control	Currency	Delivery date	Amount including tax	Command post	Place of collection	Place of delivery	Total
Instance	*	*	*	*	*	*	*		*	*		*					

C. The educational field instance

The second field is education. At this level, the standard used for comparison is the Learning Object Meta-data (LOM). The Learning Object Metadata is an international data model, usually encoded in XML, providing a model to describe the meta-data associated with the learning objects of any kind, digital or not [11].

A theoretical study of this standard allowed us to realize the instance of the following model. The instance of this field is represented in the following Figure.

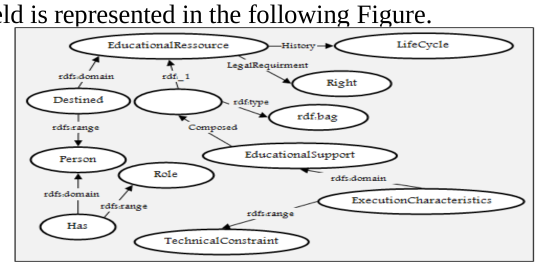

Figure 10. The instance of the educational field.

To compare this instance to the Learning Object Meta-data, we have created a comparative table among the set of

LOM classes and the set of instances. This list is represented in the following table (see Table IV). We found that we have talked about five classes defined in LOM.

TABLE IV. THE EDUCATIONAL FIELD COMPARISON.

LOM	General	Life Cycle	Meta-data	Technical	Educational	Right	Comment	Relation	Classification
Instance	*	*		*	*	*			

D. *The archaeological field instance*

The third field is archaeology. At this level, the standard used for comparison is the CIDOC Conceptual Reference Model (CIDOC CRM) standard. The CIDOC CRM is intended to promote a shared understanding of the cultural heritage information by providing a common and an extensible semantic framework in which any cultural heritage information can be mapped [12]. It is intended to be a common language for the domain experts and implementers to formulate the requirements of the information systems and to serve as a guide for the good practice of the conceptual modeling (see Figure. 10).

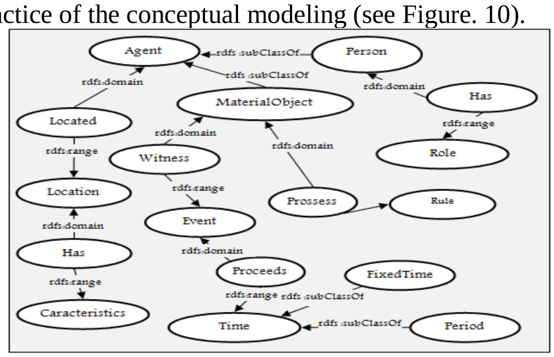

Figure 11. The instance of the archaeological field.

The comparison between the two class diagrams: CIDOC CRM and the proposed model instance lead to the result represented in the following table (see Table V).

TABLE V. THE ARCHAEOLOGICAL FIELD COMPARISON.

CIDOC CRM	Chronological period	Coordinats individual	Appellation	Temporal entity	Material entity	Conceptual	Agent	Dimension	Location	Event	Instance
Instance	*	*	*	*	*	*	*	*	*	*	*

These instances present a large part of each presented standard. The validation of the proposed model requires the validation of these instances. First, in the medical field we have discussed six concepts of DICOM standard, i.e., 55.5% of all concepts. Second, in the educational domain we have used five out of nine concepts in the LOM standard, a percentage amounting to 55.55%.Third, in the museum field, the created instance uses all the concepts of DICOM but not all the sub-concepts (classes) that are almost 100% of concepts. Finally, we have discussed the medical field and we noticed that this instance uses ten classes from the Open EDI standard. The created instances concern different fields. Thus, we mentioned a set of classes from the proposed meta-classes except the classes representing mobility.

V. CONCLUSION

In this work, we presented the various phases of the generic model realization for the pervasive environment. This generic model is presented in a class diagram and it is used in the adaptation phase. Besides, the proposed model is instanced for use in e-learning and NewsML applications through the descriptive languages as RDF, RDFS and OWL. After validating the proposed class diagram model, the instances of this proposed generic model used in the second phase of our proposition which is the adaptation phase. In this phase, we are going to precede three types of adaptation: context adaptation, content adaptation and network adaptation.

VI. REFERENCES

[1] C.Alexander, S. Ishikawa and M. Silverstein, “A pattern language: towns, buildings, construction”, Oxford University Press, ISBN 0195019199, New York, 1977.

[2] D. Salber, A. Dey and G. Abowd, “The Context Toolkit: Aiding the Development of Context-Enabled Applications”, Proceedings of HI’99, Pittsburgh, PA, ACM Press, pp. 434-441, May. 1999.

[3] S. Buchholz, T. Hamann and G. Hbsch, “Comprehensive Structured Context Profiles (CSCP): Design and Experiences”, 2nd IEEE Conference on Pervasive Computing and Communications Workshops PerCom 2004 Workshops 14-17, Orlando, FL, USA. pp. 43-47, March 2004.

[4] H. Chen, F. Perich, T. Finin and A. Joshi, “SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications”, International Conference on Mobile and Ubiquitous Systems: Networking and Services, Key: citeulike:1233279, pp. 258-267, August 2004.

[5] H. Chen, T. Finin and A. Joshi, “The SOUPA Ontology for Pervasive Computing”, Ontologies for Agents: Theory and Experiences, pp. 233-258, 2005.

[6] I. Roussaki1, M. Strimpakou, C. Pils, N. Kalatzis, M. Anagnostou, “Hybrid context modeling: A location-based scheme using ontologies”, Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW’06), pp.6-12, 2006.

[7] M. Kaenampornpan, E. O’Neill and G. Abowd, “Modelling context: an activity theory approach”, EUSAI, Vol. 3295, pp. 367-374, 2004.

[8] Wei Xu, Arta Dilo, Sisi Zlatanova and Peter van Oosterom, "Modelling emergency response processes: Comparative study on OWL and UML ", Joint ISCRAM-CHINA, Gi4DM Conference, Harbin, China, pp. 493-504, August, 2008.

[9] DICOM Standards Committee, Working Group 6, “Digital Imaging and Communications in Medicine (DICOM), DICOM Encapsulation of PDF Documents”, March, 2005.

[10] O. Troulet-Lambert, “UML et XML pour le commerce électronique Business to Business Business to Administration”, HERMES Science Europe, Paris, 2000.

[11] P. Barker, What are IEEE Learning Object Metadata / IMS Learning Resource Metadata?, cetis standards briefings series, 2005.

[12] The official web site: http://www.cidoc-crm.org/, consulted April 2013.