The Generic Model for Pervasive Information System

Fatma Achour
Multimedia, InfoRmation systems and Advanced Computing Laboratory
Sfax University, Tunisia
Email:fatma.achour@gmail.com

Anis Jedidi and Faiez Gargouri
Multimedia, InfoRmation systems and Advanced Computing Laboratory
Sfax University, Tunisia
Email: anis.jedidi@isimsf.rnu.tn, faiez.gargouri@isimsf.rnu.tn

***Abstract*—The ubiquitous computing-also called pervasive computing-regroups the characteristics of mobile computing and the techniques of context-awareness that are flexible, adaptable, and capable of acting autonomously on behalf of users. However, the pervasive computing introduces a variety of software, hardware and users engineering challenges. The pervasive computing modeling is a new research area. Indeed, we are interested in purposing methodology for modeling pervasive systems. This paper provides an overview of proposed generic model of pervasive information system. We also propose an instantiation process to validate this generic model. The generic model validation is realized by two ways; manually and automatically. The last way is obtained by the creation of automatic system of OWL instantiation. This purposed generic model represents the highest level of abstraction. It only presents the general concepts of pervasive system.**

***Index Terms*— Pervasive computing, CSCP, SOUPA, COMANTO, Activity, Transformation, MDA.**

I. Introduction

The ubiquitous systems-also known as pervasive system lists all mobile system characteristics and "context awareness" techniques. Indeed, the pervasive information systems are characterized by heterogeneous data or access and the interrogation of these systems are carried out at anytime, anywhere and from different types of equipment. Pervasive information systems are based on the **NNN** paradigm (aNywhere, aNytime, with aNything). Therefore these systems must be used in different contexts depending on the user environment, his profile and the terminal being used.

In this paper, we've focused on proposing a generic model for pervasive system. This model is reusable depending on its context of use. The various phases of model realization are shown in the above figure (see Figure.1.). In fact, we tend to work, firstly, on the transformation of every model in UML class diagram. Secondly, we will perform a comparative study of different concepts of each model and finally we'll present a class diagram valid to all these models.

This model is validated by their manual instantiation in different fields using the RDF, RDFS, and OWL language. This model also is used to create metadata ontology by the implementation of an automatic OWL instantiation system using the two languages Java and ATL.

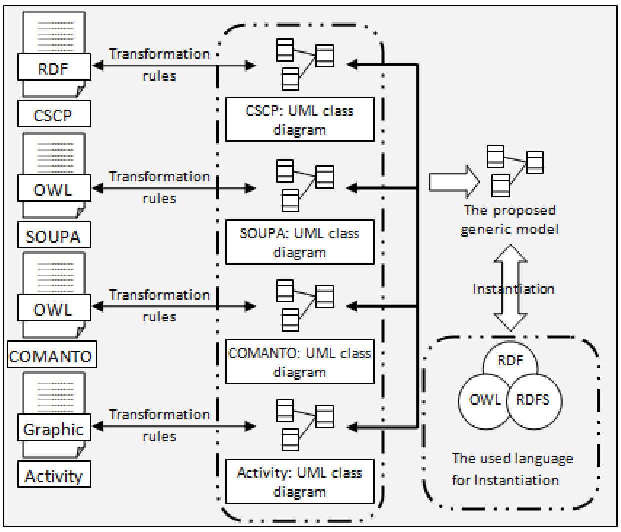

Figure 1. The various stages of proposal

II. The Proposed Generic Model

In order to propose a generic model for pervasive information systems (PIS), we conducted a theoretical study on a set of context model. Indeed, Daniel Salber defines the context in [1] "*Environmental information or context covers information that is part of an applications operating environment and that can be sensed by the application. This typically includes the location, identity, activity and state of people, groups and objects. Context may also be related to places or the computing environment.*"

A. The Used Contextual Model

For a literature interested in the context model, Held – the oldest model - is the simplest key value model to use

doi:10.4304/jetwi.4.4.343-351

[1]. After this model, more and more typical models appeared.

The models are often classified by the structure's plans of data, that are used to describe and transmit the contextual information. Indeed, six categories of context modeling are introduced. In this work, we will present four models: CSCP, SOUPA, COMANTO and Activity. The goal is to propose a model for the pervasive systems.

1) The ACTIVITY model transformation: The theory of activity is a descriptive tool to help understand the unity of knowledge and activity [2]. In fact, it concentrates on the practice of the individual and collective work. It also allows identifying the role of the material artifacts in the working process. 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 possible contexts in ubiquitous computing. Indeed, the presented model of activity uses the existing information in the theory of the activity while adding the dimension "time".

The transformation processing of context "Activity" structure in UML class diagram represents the transformation of a simple graphic model to UML class Diagram.

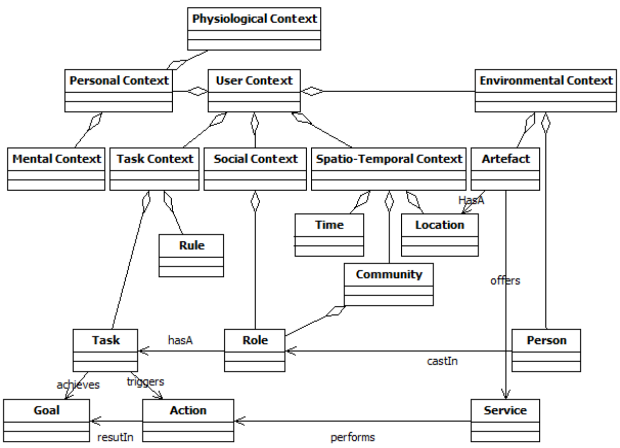

Figure 2. The class diagram of model Activity.

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 concerning the user that the system is interested in.
- **Tools:** list of the available tools in the public place.
- **Rules:** norms, social rules and legislations with which the user is related to the others in his community.
- **Community:** information concerning the persons around the user who can have an influence on the activity.
- **Object:** the intention and then objective of the user.
- **Time:** it is a moment in a particular situation where an activity took place.
- **Division of labour:** the distribution of the tasks between the members of the community.
- **Result:** the result of the object transformation.

The result of transformation represented in the figure 2 above is due to the use of the following rules:

The nodes are transformed into classes,

The arcs are converted into oriented associations in the UML diagram,

The arcs of type "partOf" are transformed by UML aggregations.

---

2) The SOUPA ontology transformation: The SOUPA project is created in November, 2003. It is part of the work caring out semantic web by the group "Special Interest Unicomp" [3]. The objective of this project is to define ontology, written in OWL, which supports pervasive applications.

The SOUPA concepts cover themselves with the set of existing ontology vocabularies. Ontologies referenced by SOUPA included the Friend Of Friend (FOF), DAML Time, Open Cyc, RCC, COBRAONT, MoGATU BDI and Rei ontology policy. SOUPA contains two distinctive connected subsets of ontologies: the SOUPA Core and the SOUPA Extension. The first ontology allows defining generic vocabularies for the construction of the pervasive applications [4]. While the second allows defining additional vocabularies to support a specific type of applications and provide examples of defining new extensions of SOUPA.

In this work, we are interested in SOUPA Core. The SOUPA core ontology includes a set of sub-ontologies.

- **Person:** this ontology defines a 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 pervasive computing systems.
- **Agent and BDI:** during the construction of pervasive systems, it is useful to present the computer model entities as agents.
- **Time:** SOUPA defined a set of ontologies to model the time and the temporal relations.
- **Space:** this ontology is designed to support reasoning about spatial relationships between different types of geographic spaces from geographic coordinates.
- **Event:** is the event of the activities that have the same spatial and temporal extensions.

The SOUPA core class diagram is achieved through a theoretical study on the different concepts in each ontology.

In fact, a set of transformation rules are defined in a manual way to assure this transformation.

The class diagram corresponding to SOUPA ontology is represented in figure 3.

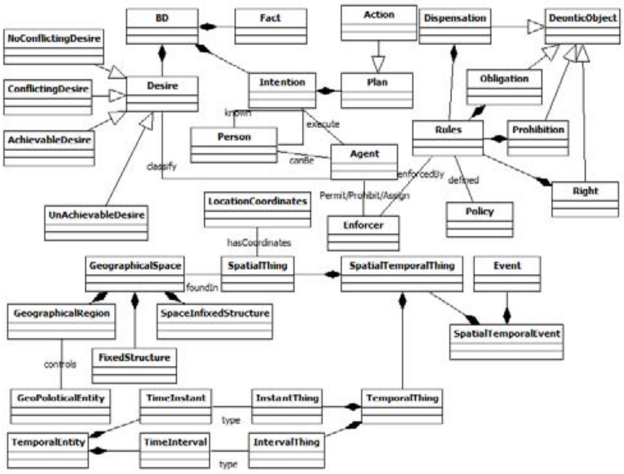

Figure 3. The class diagram of SOUPA Ontology.

The transformation rules that allow the realization of 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 relations of composition with OCL constraints.

SubClassOf: the property "subClassOf" is transformed by an extend relationship.

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 ontology of context Daidalos, also called COMANTO (Context management oNTOlogy) was created in 2005 by Roussaki [5]. It is the result of a hybrid context modeling approach to handle context objects and context knowledge.

The COMANTO ontology is proposed as a public context semantic vocabulary supporting efficient reasoning on contextual concepts (such as users, activities, tools, etc.) and their associations. The ontology is used to collect a structured semantic representation about generic context information and not a domain or application-oriented.

The ontology classes of context COMANTO can be described as follow:

- **Person:** it is the central entity in the ontology COMANTO. It represents all the human entities and offers diverse properties of 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 representation or geographic.
- **Preferences:** in order to represent user, service, network and device preferences.
- **Service:** it stores information relevant to applications taken by the user.
- **LegalEntity:** This class is mentioned as a representation of corporate actors involved in the supply chain pervasive computing.
- **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.

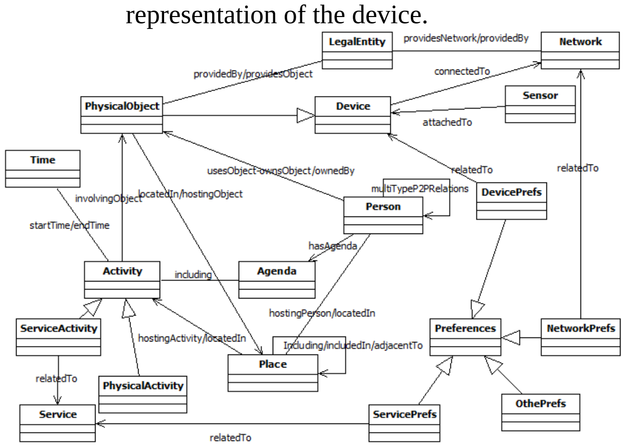

Figure 4. The class diagram of the ontology COMANTO.

The transformation of ontology COMANTO in a corresponding class diagram is carried out according to the following transformation rules.

The OWL nodes are transformed by UML classes.

Two-way links titled (in reverse) are transformed into two-way association with two titles.

The unidirectional arcs between nodes are transformed into unidirectional association.

The relation "rdfs:subClassOf" is transformed into a legacy relation.

The relation "owl:disjointWith" is transformed into association with OCL constraint.

---

4) The CSCP model transformation: The language CC/PP (Composite Capabilities / Preferences Profiles), is a recommendation of W3C (World Wide Web Consortium) within the framework of the works on device dependence to support the negotiation of contents between a Web browser and a server. CC/PP is used to customize the content based on their abilities and preferences [6].

In 2002, an extension of CC/PP was proposed by Held.

The CSCP (Comprehensive Structured Context Profiles) which does not define any fixed hierarchy to solve the problems of CC/PP. The context model CSCP represents profile sessions and is based on RDF; it does not impose any fixed hierarchical structure for the context notion, thus inherits the full flexibility and expressive power of RDF.

The CSCP allows the merging of profile fragments that are dynamically retrieved even from different web sites [7].

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

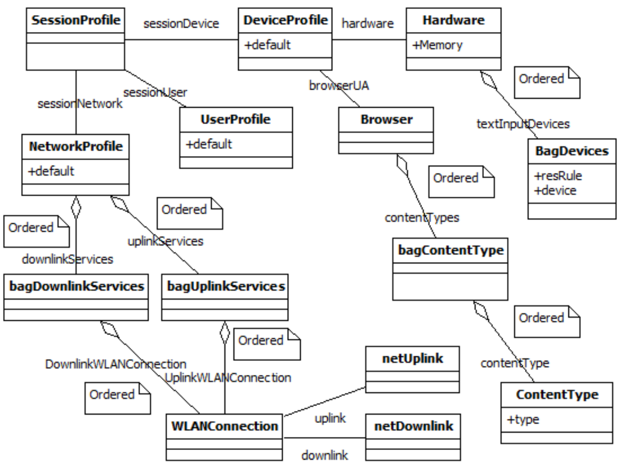

Figure 5. The class diagram of the CSCP model.

The CSCP transformation of the model graph RDF (Resource Description Framework) in a valid class diagram is realized according to the following rules of transformation.

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. 5.).

B. The Proposed Model

The adaptation of the applications to the pervasive system is essentially based on a model. To unify existing models, we try to make a model according to four proposed ones; CSCP, SOUPA, COMANTO and Activity model.

The objective of this transformation is to define an abstract model which satisfies the constraints of an application while respecting the semantic relations between concepts (classes).

The objective of beginning with a comparative table between concepts of every mode is to collect all the concepts that will be used at the level of the proposed model.

1) Pre-processing of data: After the theoretical study of four presented models, we have made a comparison between the different concepts used in each model. The goal is to make an application structure complies with all of these models. The result of the theoretical study is presented in the table II.

All these concepts will be transformed into a valid class diagram. The latter will be used in the phase of adaptation.

2) The proposed UML class diagram: To define structure that is extensible, reusable, flexible and interoperable, we propose a generic model of PIS that incorporates a semantic dimension. It was designed for the adaptation of applications to the pervasive system.

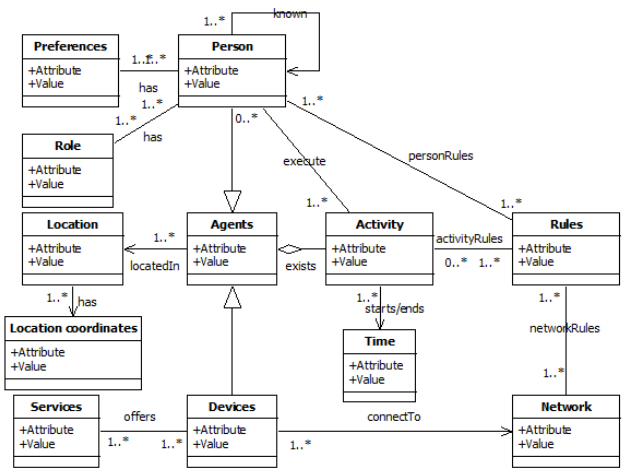

Figure 6. The proposed UML class diagram.

In 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 side of the same name. For example, a network may be

TABLE I.
THE COMPARATIVE TABLE OF CONCEPTS

ACTIVITY	COMMANTO	CSCP	SOUPA	Device
Artifact	Device	Device	profile	Agent
Person	Person	Person	User profile	Person
Social	Role (social context)	multyType P2P relations	–	Property: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

the characteristic of the flow rate. This instance may have the attribute "flow rate" and the value "1 Mbps".

The representation that we make to the PSI provides that its definition to be dynamic and scalable since it is composed of one or more characteristics. These characteristics are not defined in advance and are defined by the system designer according to their needs

- **Agent:** this class is used for presentation of different actor in pervasive system. It assembles the human and the device actor.
- **Persons:** presents the static and the dynamic characteristics of human actors.
- **Device:** presents the devices characteristics in pervasive system.
- **Service:** presents the services characteristics offer by each device.
- **Networks:** regroups the characteristics of different types of network.
- **Location coordinates:** represent the spatial relation between different locations in pervasive system.
- **Preferences:** presents the user preferences.
- **Activity:** presents the characteristics of activity requested by the user.
- **Rules:** regroups the different rules of activity, person and network interacting in pervasive system.
- **Time:** presents the characteristics of temporal and the relation ones of different thing in pervasive system.
- **Locations:** represents the characteristics of localizations of human and mobile device in pervasive system.

III. The Generic Model Manual Instantiation

UML is a semi-formal language that allows us to have a better visual clarity of our generic model [9]. However, to describe instances of this generic model we have chosen formalisms RDF/RDFS/OWL to create ontology. The justification of ontology chooses is described in [8]. The formal languages are inspired by the logic of description. They are more appropriate for describing the semantics because they provide us with basic predicates that we can re-use: disjunction (owl:disjointWith), equivalence (owl.equivalentClass), equality (owl:sameClass), generalization (rdfs:subClassOf and rdf:type)...

In addition, the use of RDF/RDFS/OWL allows us to validate our model experimentally using existing programming interfaces for the semantic web. They are able to interpret and to reason representations of RDF/RDFS/OWL. Also, note that UML and RDF languages are not disjointed. The basic building block of RDF is the triple [subject, predicate, object] are implicit in UML and in all languages. Thus, the associations between classes and the relationships between class and its properties in UML model can be spelled by RDF triples.

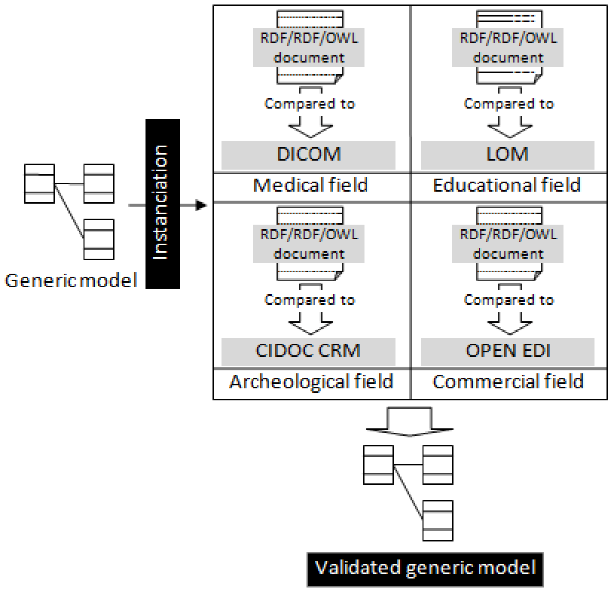

Figure 7. The validation process of the generic model

To validate the proposed generic model, we try its instantiations in different fields. These instances are created manually and compared with existing models. We are interested in all validation process on the respective domain's ontology of medical, educational, archaeological, and commercial field.

A. The Generic Model Validation in the Medical Field

In order to validate the proposed model, we've created an ontology representing an instance of generic model 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 set of classes corresponding to DICOM. Therefore, DICOM (Digital Imaging C0mmunication in Medicine) is a format that defines methods of connection, transfer and identification of medical data [10]. We found that this instance represents an essential part of this format. After the instance creation, we have compared the set of DICOM classes and the set of instance classes. We found that this instance represents an essential part of this format.

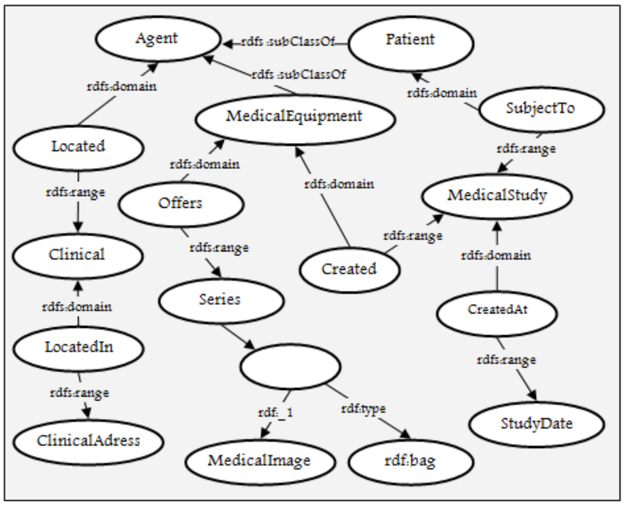

Figure 8. The instance in medical field.

B. The Generic Model Validation in the Educational Field

The second used field for the similitude is the educational one. At this level, the standard used for comparison is the LOM (Learning Object Meta-data).

Learning Object Meta-data is an international data model, usually encoded in XML, providing a model for describing meta-data associated with learning objects of any kind, digital or not [11]. A theoretical study of this standard allowed us to realize the instance of model that follows. The instance of this field is represented in the following RDF/RDFS/OWL document (see Figure.9.). Comparing this instance to the

Learning Object Meta-data, we've compare the set of LOM classes and the set of instance ones. In fact, we observed that we have talked about five classes defined in LOM.

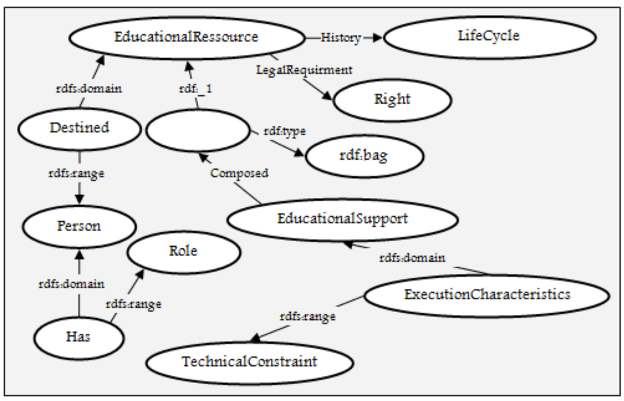

Figure 9. The instance in educational field.

C. The Generic Model Validation in the Archaeological Field

The third used field is the archaeological. At this level, the standard used for comparison is the CIDOC (CIDOC Conceptual Reference Model) standard.

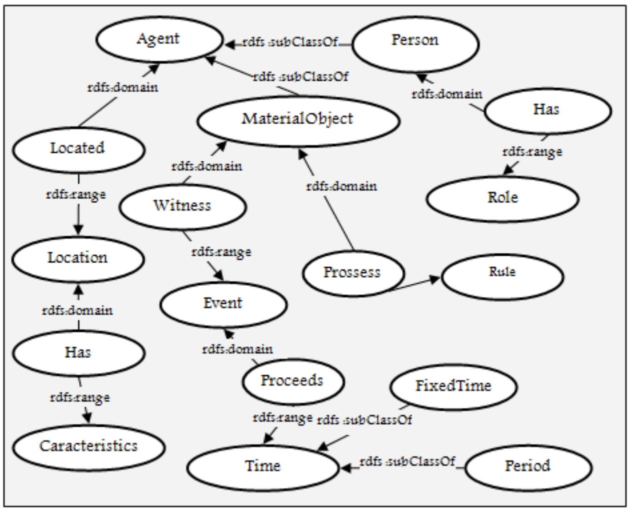

Figure 10. The instance in archaeological field.

The CIDOC CRM is intended to promote a shared understanding of cultural heritage information by providing a common and extensible semantic framework that any cultural heritage information can be mapped to [12]. It is intended to be a common language for domain experts and implementers to formulate requirements for information systems and to serve as a guide for good practice of conceptual modeling. The instance of this field is represented in the following RDF/REFS/OWL document (see Figure.10.).

D. The Generic Model Validation in the Commercial Field

We used the e-commerce open EDI to validate the proposed design pattern structure. The open EDI proposes a standard to describe the roles of diverse companies and guarantees that their applications are able to communicate by EDI messages [13]. To do it, we realized an instance in the commercial field; this instance is presented in the following RDF/RDFS/OWL document (see Figure. 11.).

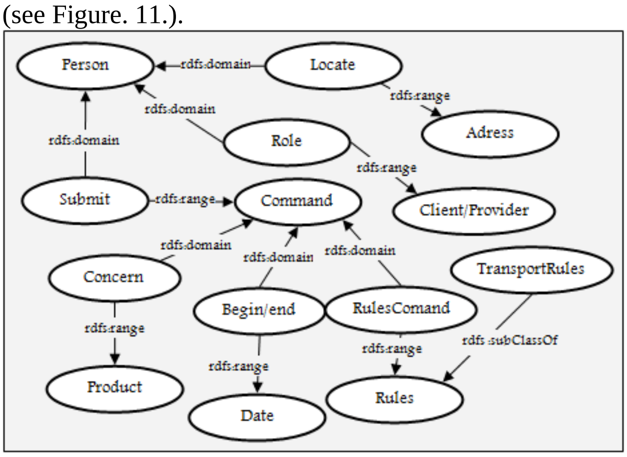

Figure 11. The instance in commercial field.

The comparison of the Open EDI concepts with the proposed instance concepts showed that that the proposed instance represent a part of commercial domain. Indeed, various information in commercial field, have been represented in this instance.

These instances represent a large part of each presented standard. In fact, the validation of the proposed generic model requires the validation of these instances. First, in the medical field, the comparison result demonstrates that we've discussed six concepts of DICOM standard that is to say 55,5% of all concepts. Second, in the educational domain we've used five concepts out of nine 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 sub-concepts (classes) that is almost 100% of concepts. Finally, we've discussed the medical field and we noticed that this instance uses ten classes from Open EDI standard. The created instances concerns different fields, we mentioned a set of classes from proposed meta-classes except the classes representing the mobility.

IV. The Automatic Instantiation Of The Proposed Model

We created an automatic instantiation system of the proposed model using Java and ATL languages in order to produce the metadata ontology [14] that will be used when adapting pervasive information systems (see Figure 12),

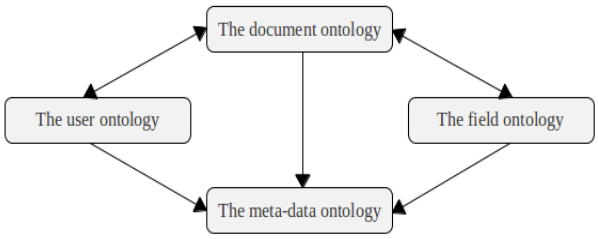

Figure 12. The relations between ontologies [14].

- **The field ontology:** It represents the field's typical model consisting of semantic network (concepts and relations) for the application field.
- **The meta-data ontology:** It provides metadata information for the fragments.
- **The document ontology:** It contains the document and the adaptation models. It consists in adapting the rules which are composed of model adaptation elements.
- **The user ontology:** It describes the characteristics of the user adaptation and allows the adaptation model to define the classification rules and the adaptive methods' stereotype.

The process of creation of this system is happening by three steps.

- **Step 1: Proposed Model Transformation:** It consists in transforming the proposed UML model to an ontological form represented by OWL.
- **Step 2: OWL Instantiation with Jena**: It prepares all Java methods which consist in

creating OWL classes conforming to the proposed model.

- **Step 3: Instantiation OWL with a Java prototype:** It creates a set of Java interfaces that simplify the task instantiation

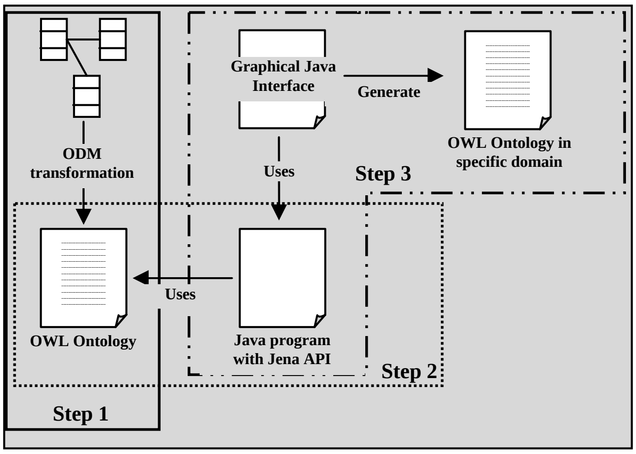

Figure 13. The process of automatic validation.

A. The Proposed Model Transformation

To transform the proposed model to OWL ontology, we integrated the complete work of ODM (Ontology Definition Metamodel) [15] which offers a set of transformations based on a set of meta-models for the approximation of the metamodeling and ontology world.

This work presents an implementation of the ODM using the ATL language (Atlas Transformation Language) [16] which was designed to perform the transformations within the MDA Framework. It supports the three meta-models; UML2.0, OWL and XML that are defined in the ODM. Figure 14 shows the complete transformation scenario.

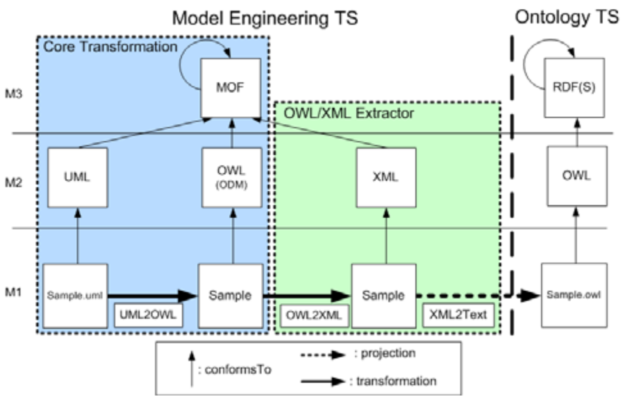

Figure 14. UML2OWL: The completed scenario transformation [17].

The transforming process model accepts an UML model file as input and generates a new OWL text file as output. This is the first step in performing the automatic instantiation system. The input and output of this transformation are represented in Figure 15.

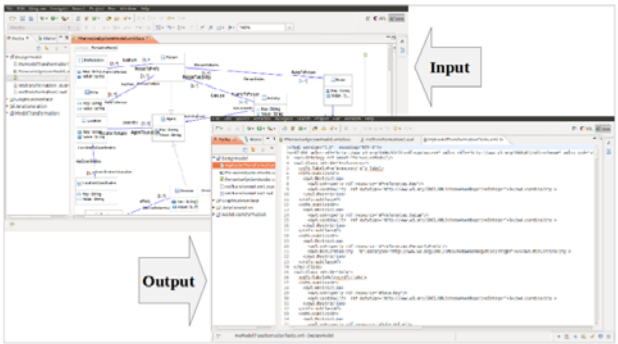

Figure 15. The input and the output transformation.

B. OWL Instantiation with Jena API

In this step we are considering the implementation of all OWL classes and the methods allowing their creation and the properties already defined according to the transformation of the proposed model result. To achieve this we used the Jena API [18] that provides a set of java classes dedicated to the handling of the described ontologies in OWL. The whole of Jena classes used in the implementation of this system are:

- **The model management (OntModel):** provides the possibility of creating an ontology OWL.
  <owl:Ontology
  rdf:about="PervasiveModel"/>
- **The classes management (OntClass):** allows the management of the ontology classes with cardinalities for each class.
  <owl:Class rdf:ID="Role">
  <rdfs:label>Role</rdfs:label>
  <rdfs:subClassOf>
  <owl:Restriction> <owl:onProperty
  rdf:resource="#Role.PersonToRole"/>
  <owl:minCardinality
  xml:lang="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:minCardinality>
  </owl:Restriction> </rdfs:subClassOf>
  <rdfs:subClassOf>
  ...
  </rdfs:subClassOf>
  <rdfs:subClassOf>
  ...
  </rdfs:subClassOf>
  </owl:Class>
- **The ressources (createResource) and the properties (createObjectProperty) management:** allowing the addition of a resource in a Model with the adequate properties.
  <owl:ObjectProperty
  rdf:ID="Person.RulesToPerson">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Rules"/>
  <owl:inverseOf
  rdf:resource="Rules.PersonToRules"/>
  </owl:ObjectProperty>

C. Instantiation OWL with a Java prototype

The major advantage of Java is the facility that this language provides for quick programs with a graphical, almost professional, interface. In designing this interface, we choose the plugin "visual editor". The Visual Editor project provides reference implementations of a graphical user interface builder for the JFC and SWT widget toolkits [19]. The program behind the graphical interface is intended to create OWL ontology, a way that we also used with the Jena plugin. The output of the graphical interface is an OWL ontology described through figure 16. This ontology conforms to the ontology resulting from primary processing.

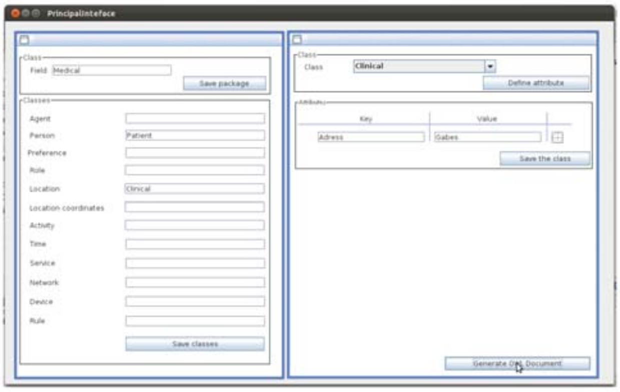

Figure 16. The prototype interface.

V. Conclusion

In this work, we presented the various phases of generic model realization for the pervasive environment. This generic model is presented in a class diagram and it will be useful in the adaptation phase. Besides, it will be instanced to use in e-learning and NewsML applications through the descriptive languages as; RDF, RDFS and OWL.

The proposed automatic system allowed us to define the meta-data ontology that conforms to the proposed UML class diagram.

After validating the proposed class diagram model, the instances of this proposed generic model will be used in a second phase of our proposition which is the adaptation.

In this phase, we are going to precede three types of adaptations: context adaptation, contents adaptation and network adaptation.

References

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

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

[3] 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 (August 2004), pp. 258-267. Key: citeulike: 1233279, 2004.

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

[5] F. Topcu, “Context Modeling and Reasoning Techniques”, SNET Seminar in the ST, 2011.

[6] T. Strang, T. Finin and C. LinnhoffPopien, “A Context Modeling Survey”, Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp, 2004.

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

[8] F.Siddiqui, M.A. Alam “Web Ontology Language Design and Related Tools: A Survey”, Journal of Emerging Technologies in Web Intelligence (JETWI), Vol. 3, No. 1, February 2011, PP. 47-59

[9] M. CHEVALIER, C. SOULE-DUPUY, P. TCHIENEHOM, “Interopérabilité de Profils pour l’Accès à des Ressources”, inforsid, 2006.

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

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

[12] The official web site CIDOC/CRM : http://www.cidoc-crm.org/

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

[14] S. Garlatti, S. Iksal, S.Declarative, “Declarative Specifications for Adaptive Hypermedia Based on a Semantic Web Approach”, ACTI - International Conference Proceedings, 3-540-40381-7, pp.81– 85, 2003.

[15] D. Gaaevic, D. Djuric, Devedzic, Vladan and Selic, “Bran, Model Driven Architecture and Ontology Development”, Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2006

[16] The Web site ATL. http://www.eclipse.org/m2m/atl, visited 18 Mai 2012.

[17] The Web site ODM. http://www.eclipse.org/m2m/atl/usecases/ODMImplementation/, visited 29 Mai 2012.

[18] R. Fitri Sari, N. Ayuningtyas, “Implementation of Web Ontology and Semantic Application for Electronic Journal Citation System”, Journal of Emerging Technologies in Web Intelligence (JETWI), Vol. 2, No. 1, February 2010, PP. 34-41

[19] The Web site Custom widget: http://www.eclipse.org/articles/Article-VE-Custom-Widget/customwidget.html, visited 23 Mai 2012.