An Ontology-Based Context Model for Ubiquitous Applications

Boudjemaa Boudaa[1], Slimane Hammoudi[2], Mohammed Amine Chikh[3]
[1]Department of Computer Science
Ibn Khaldoun University
Tiaret, Algeria
[2]MODESTE, ESEO
Angers, France
[3]Department of Computer Science
Abou Bekr Belkaid University
Tlemcen, Algeria
boudjemaa.boudaa@univ-tiaret.dz, slimane.hammoudi@eseo.fr, mea_chikh@mail.univ-tlemcen.dz

ABSTRACT: *The development of context-aware ubiquitous ap-plications has been an emergent subject of several researches in pervasive computing. In different context modelling proposed approaches, we notice that there is still no consensus and their context models are either incomplete or domain specific. This paper aims to summarize these approaches and proposes a context model which should be common to be used by different context-aware applications, specific to cover the main contextual entities existing in the literature and flexible to allow taking into account new entities specific to a given application domain. To accomplish these requirements, we adopt an ODM (Ontology Definition Meta model) approach to create a high level modelling of context information using ontologies. The main objective of the ODM is to bridge the gap between traditional software tools for modelling and artificial intelligence techniques for making ontologies.*

Keywords: Context Model, Ontology, ODM, Context-awareness

Received: 30 October 2012, Revised 3 December 2012, Accepted 9 December 2012

1. Introduction

Nowadays, the new technologies in wireless communications as well as the increasing use of mobile devices (laptops, smartphones, tablet PC, ...) have stimulated the emergence of a new computing paradigm called pervasive computing. In fact, we have moved from the desktop computing paradigm to the mobile and ubiquitous computing paradigm. Pervasive computing, firstly introduced in 1991 by Weiser, refers to the seamless integration of devices into the user's everyday life. As indicated in [1], "*appliances should disappear into the background to make the user and his tasks the central focus rather than computing devices and technical issues*" as . Nowadays, computing applications operate in a variety of new settings; for example, embedded in cars or wearable devices. They use information about their context to respond and to adapt to a change in computing environment. They are, in short, increasingly context-aware.

Several definitions of "*context*" notion are given in literature [2] among which the most popular one is given by [3]: "*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*". In this paper, we concentrate on the user with his various mobilities that give rise to context change compared to the application that remains unchangeable in most cases.

Thereafter, this context information is used in context-aware pervasive system by implementing a mechanism of context-awareness or context-aware adaptation. With respect to con-text notion definition, there have been numerous attempts to define context-awareness terminology that was introduced by Schilit et al. [4]. The author in [5], defines context-awareness "*as the ability of a program or device to sense or capture various states of its environment and itself* ". Referring to this definition, a context-aware application must have the ability of collecting context information and adapting their behaviour to context changes at run-time. Context-aware applications react to information gathered from their environment and provide relevant information and/or services to the user, where relevance depends on the user's task [3].

The first step in the context-aware applications development is to model correctly all context information [6] with accuracy in order to stock it in repository with a better formalism in order to facilitate their re-use hereafter. The accurate context modelling leads to provide the correct behaviour in the correct form to the correct user at the correct time in the correct place [7].

In the literature, many contextual models have been presented in the pervasive computing field [8], [9], [10]. The earliest model was the key-value model that borrowed the database mechanism to put the value of contextual information as an environment variable (or key). Later, the traditional software engineering methodologies were applied to represent the context: the graphical model like ORM (Object-Role Modelling and UML) and the object-oriented model. The graphical model provides a clear and intuitive view of context by describing the facts and properties as nodes and the relationships between them as edges. The object-oriented models make use of their own features of encapsulation and reusability to introduce an efficient abstraction and classification mechanism for contextual modelling. Compared to these contextual modelling approaches, the logic model does not concern itself about how the context is organized or represented. It provides a formal and abstract context model about how to reason with part of the potentially available context and how to solve the compatibility among different contexts. However, the ontological models have the advantages of the object-oriented and logicmodelling approaches. They provide a formal way to model context into well-structured terminologies, and also support formal reasoning [9].

According to [8], [11], [12], [10], the ontologies [13] offer a better formalism to build consensual terminologies of context domain in a formal way so that they can be more easily shared and re-used by different partners in mobile and pervasive environments. Ontological models have clear advantages regarding support for interoperability and heterogeneity. Moreover, since they support the representation of complex relationships and dependencies among context data, they are particularly well-suited to their cognition of high-level context abstractions [10]. A set of ontologies have been developed specifically for use in pervasive computing, the most of them are summarized in [8], [9], [10], none of which appears to cover adequately the space of concerns applicable to application designers. For this purpose, the aim of this paper is to present a general covered ontological context model that ensures to fulfil the following requirements:

1) Sufficiently general to be used by different user-centered mobile applications,

2) Sufficiently specific to cover the main contextual entities proposed in the state of the art of context-aware mobile applications and,

3) Sufficiently flexible to allow an extension and to take into account new entities specific to a given application domain.

This model is based on the combination the Model-Driven Development (MDD) and Ontologies approaches for to benefit of more advantages in both paradigms. The Ontology Definition Metamodel (ODM) proposed by OMG [14] allows making this ontological model in model driven development process for easy development of context-aware mobile applications.

After this introduction, the advantages of using ontologies are clearly detailed in section 2. In section 3, a definition of context-aware ubiquitous application has been exposed. Section 4 exhibits, with an example, our context modelling based on ontologies and using the ODM specification. Then in section 5, we present the most of the related works. Finally, section 7 concludes this paper and indicates a future research.

2. Using Ontologies in Context Modelling

The term of ontology [13] was borrowed from philosophy and introduced into the knowledge engineering field as a means of abstracting and representing knowledge. Ontology is a promising means for knowledge sharing and re-use. It allows the capture and specification of domain knowledge with its intrinsic semantics through consensual terminology and formal axioms. Ontologies support a set of modelling primitives to define classes, individuals, their attributes and their relations. The formal axioms specify the intended meaning of the terminology with the modelling primitives and their apriori relationships in an explicit way [15]. The consensual terminology makes it possible to share and reuse data and knowledge across different components in a system and across different pervasive systems.

Various ontology-based models of context information are proposed in literature (see section V) where each argues the benefits by focusing on ontologies in context modelling. In this section, we synthesize the most important of these benefits in which we are based to propose our ontology-based model to represent context information:

2.1 Large-scale and extensible formalism

Ontologies are represented by several web languages like OWL that focus on the premise of the open world assumption (OWA: Open World Assumption) [16]. This means, we can't assume we know all context information about an application domain. And consequently, the lack of specific context information does not mean denial. For example, a teacher and student classes of an ontology, we can't say that the teacher can't be a student because we didn't declare that these two concepts with owl: disjointWith. In contrast, databases and object-oriented programming (OOP) focus on the closed world assumption (CWA: Closed World Assumption) related to the failure or negation (Negation as Failure), as it amounts to believing false every predicate that cannot be proved to be true[1]. The CWA typically applies when a system has complete control over information; this means that we assume we know all the information about the domain. Therefore, the lack of specific context information means it is denied. For example, in OOP, if a typical airline database does not contain a seat assignment for a traveller, it means the traveller has not checked in. As conclusion, when we want to describe a context information in open and large-scale environment such as pervasive environment, we should be based on an OWA formalism like ontologies (OWL) to describe knowledge in a way that is extensively able to create, add or modify other artefacts (concepts, relationship, axioms, ...) to the existing context ontologies in flexible and dynamic manner.

2.2 Expressiveness and semantic richness

Pervasive computing systems, by nature, are complex and heterogeneous which must integrate the context information from diverse and heterogeneous sources. Ontologies based on more expressiveness and semantic web language as OWL, provide a well-founded mechanism for the representation and exchange of such structured information like context information to be shared and reused by different pervasive computing systems [9]. OWL language offers more capabilities, namely (i) to describe complex context data with a consensual conceptualization, including concepts, taxonomies and relationships between concepts and their properties; and (ii) to provide a formal semantics to context data by using axioms and constraints. The axioms and constraints are the main building blocks for making the semantic interpretation for the concepts and relationships of the ontologies, thus, making the possibility to share and/or reuse context among different information sources [10].

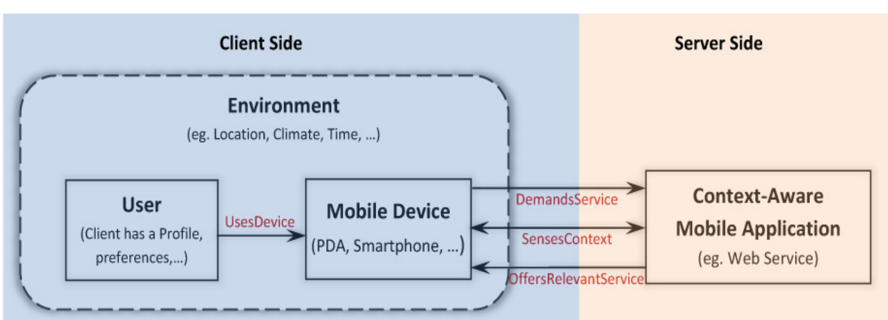

Figure 1. Context-Aware Ubiquitous Application Overview

2.3 Reasoning and inferring capabilities

Application domain of ontologies exceeds information modelling and describing to reasoning about it. In the literature, many

[1]http://en.wikipedia.org/wiki/Closed_world_assumption

pervasive computing projects have applied ontologies for their reasoning power. The reasoning languages (as, OWL-DL, SWRL) allow (i) to check for consistency of the concepts and their relationships describing a context model, and, more importantly, (ii) to derive others context information from basic ones using rule-base inference engine (e.g., the user's activity can be automatically recognized [10]). Ontology is a promising technology that can be used to model the maximum context information surrounding a situation, and by reasoning mechanism, to deduce other contextual information that could complete this situation context describing.

3. Context-Aware Ubiquitous Application

Before presenting our context model, it is important to give an explicit definition of these applications whose development will be based on this context model.

In pervasive computing scope, a class of applications that has raised increasing interest in the research community is context-aware ubiquitous applications invoked by a mobile user (user-centered). These applications can, dynamically, capture and take advantage of contextual information (such as, user location, time and weather, device and user activities,...). Thus, context-aware applications can sense their context of use, and adapt their behaviour accordingly to run on different mobile devices. From an implementation point of view, the emerging technology of web services seems promising for such mobile applications.

Figure 1 shows a simple vision of user-centered context-aware mobile application and its invocation. The user hidden behind his mobile device (at client side) demands service or information to the application (at server side). The latter, and before providing a relevant response depending on the context, should sense and take into account all contextual information surrounding the user and his interaction with application as user's preferences and some information on the environment in which he evolves (location, time, weather,...).

4. Ontology-Based Context Modelling

In the survey on context-aware service engineering [11], the authors propose six categories of approach to manage context: Middleware solutions and dedicated service platforms,

Use of ontologies, Rule-based reasoning, Source code level programming/language extensions, Model-driven approaches and Message interception.

Generally, any of these approaches categories has its pros and cons. For example, the source code level approach can give more freedom to developers to do all kinds of context-aware adaptation, but this approach does not separate apart the concerns on context-adaptation (or context-awareness) and suffers from a significant maintenance cost. As to the model-driven approach[17], apart from its advantages, it requires to keep the consistency between high-level models and low-level executable code at all times, which brings extra complexity.

Also, it is desirable the use of a hybrid approach combining two or more approaches at the same time [12] which bring extra benefits and present other approaches for context-aware mobile application development.

In this way, the aim of this paper is to propose a context modelling using MDA and ontology approaches. We will present below our context model based on ontologies structure to capture all context information. As we have seen in section II, the use of ontologies allows the interoperability in a pervasive environment in which it is very necessary that the context terminology should be commonly understood by all participating devices.

4.1 Ontologies and ODM specification

In information systems, an ontology conceptualises formally the knowledge within a domain as a set of concepts and relationships between these concepts at different levels of abstraction in order to describe this domain. And, to render such description shared and reused. Ontology is a "*formal, explicit specification of a shared conceptualisation*" [18].

The contribution of ontologies to understand, share and integrate the information is well-established. Indeed, research and practice in this area are beginning to bear fruit, especially for the semantic Web. Recently, the ontologies are used to model context information in various works for developing context-aware mobile applications (section V). And for its broad use namely in Model-Driven Engineering (MDE), the OMG proposed Ontology Definition Metamodel (ODM) [14], a standard metamodel which aims to allow the modelling of an ontology according to MOF formalism (see figure 2). Moreover, we can treat all ontologies by MDE with all its benefits, mainly, the transformation process between models. The main objective of the ODM is to bridge the gaps between ontologies and MDA modelling spaces [19].

For this goal, ODM defines five metamodels (RDFS, OWL, Topic Maps, Common Logic and Description Logic), two UML profiles (the profile RDFS/OWL, Topic Maps on profile) and a set of QVT mappings from UML to OWL, Topic Maps to OWL and RDFS/OWL to Common Logic [14]. We are interested by the OWL ontological language [20] for its high expressiveness and especially by its OWL-DL (Description Logic) version that allows reasoning to infer other high level context information from basic low level context information. In this work, we will use the OWL Metamodel and the UML Profile for OWL defined in ODM to describe our ontology-based context modelling.

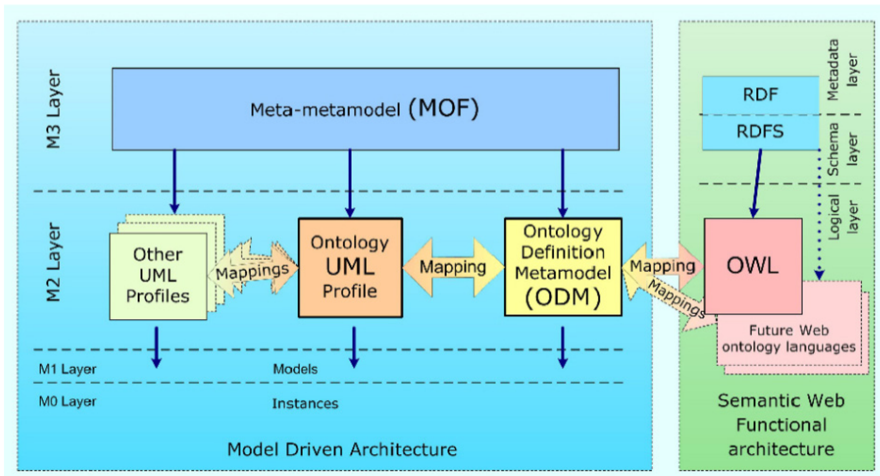

Figure 2. The Ontology Definition Metamodel in MDA

4.2 MeetingGuide Example

MeetingGuide (MG) aims to facilitate the organization of meetings, including those scientific (conference, seminar, workshop, ...), by giving all the necessary information to participants. It is a context-aware pervasive system that must accompany the participant in his various mobility areas from departure to the meeting cities, into hotel and meeting rooms.

To realize this system, it is envisaged the development, at least, a three context-aware applications able to run on various mobile devices (Laptop, PDA, Smartphone, Tablet PC, ...)[2], namely:

• **BookingTravel application:** For a participant who would like to join the meeting from his departure city, this application proposes a travel booking by several conveyance means (bus, train, boat, plane, ...). This reservation take into account the distance between the location (departure city) and place of meeting (meeting city) according to budgetary constraints of the participant and his preferences. Here, the context information are the location of cities, the balance of bank account and preferences of a participant.

• **Presentation application:** Used for any presentation depending on location (as context information). For instance, in the Meeting room, this application gives the current program communications ordering and filtering on interest centre (context: areas of interest in participant profile). And in hotel room, it gives the programs of TV channels selected by favourites tv show of the participant and speaking in his native language (context: language and preferences).

• **VisitingCity application:** Suggests to visit the meeting town following to the weather forecast (context: weather forecasts) by providing the principal outdoor attractions in addition to indoor attractions in case of nice weather. If not, it provides only the indoor attractions in this town.

[2]Actually, the majority of these devices are equipped with Infrared, Bluetooth, WIFI or GPS technologies able to sense and give a context information.

For this case study, it should be noted that the participant's location is captured by using miscellaneous technologies of sensing: (i) Wireless technology (e.g. WiFi access points in the meeting room) (ii) Infrared or Bluetooth technologies (e.g. in the hotel room, the TV is equipped with an infrared interface) (iii) GPS technology (for choosing the means of conveyance, for example).

4.3 An ODM-Driven Context Model

Within the model-driven development, the use of a metamodel not only guarantees a strong and focused semantics tied to a particular application domain, but also offers a precise abstract syntax and a common representation to any developed model. In this research, we present a context modelling using ontological formalism within MDA space. Thanks to ODM that provides a OWL metamodel, conform to MOF specification, composed of set of diagrams to represent the different ontology elements (OWL Class Descriptions Diagram, OWL Properties Diagram, OWL Individuals Diagram, ...). And, taking into account the importance of UML, the ODM developers defined mappings between the standard UML metamodel and OWL metamodel by proposing an Ontology UML Profile (OUP) focused on UML extension mechanisms. The purpose of OUP is to enable the use of the standard UML graphical notation to develop ontologies [19]. OUP reuses UML con-structs when they have the same semantics as OWL, or, when this is not possible, it uses a stereotypes UML constructs that are consistent and as close as possible to OWL semantics [14].

Our ontology-based ODM-driven context model is schem-atized in Figure 3 that is structured by three sub-ontologies at three levels of abstraction and generality [21]: generic, domain and application heavyweight ontologies containing axioms and constraints for more semantic expressiveness.

1. Generic Ontology: is a top-level ontology [22] which describes, independently from any particular domain, the three main dimensions of the context information concerning *user, mobile device and environment* [23] and their relationships (*usesDevice and evolvedIn*):

• **User concept:** a person (or application, sometimes) who is the central entity in pervasive computing systems, that has a personnel information like profile, preferences. He evolves in an environment and uses mobile devices to invoke services,

• **MobileDevice concept:** is used by the user for service access and to capture any contextual information of the surrounded environment and of computational device hardware (device used, band-width networking, ...), and

• **Environment concept:** is the space in which the user evolves. It represents the whole information encircling the user and its computational device that can be relevant to a context-aware application. It can include different categories of information such as spatial context information (city, building, address), temporal context information (time, date, season) and climatic context information (weather forecast, temperature, ...)

2. Domain Ontology: captures the most common elements of primary context in pervasive computing defined in literature (location, profile, time, climate, activity, ...) [9]. Each concept of domain ontology should be attached to an extensible specific context ontology model named Application ontology.

3. Application Ontology: is a specific ontology which describes the task-specific contexts for a specific application (such as weather, hotel and conference in our case study). The application ontologies might build on generic and domain ontologies [24].

The context sources allow the individualization of these sub-ontologies can be provided by two different ways: sensing or inferring methods. Sensing is used directly for low-level and basic context, whereas, inferring is used to provide a high-level context from this low-level using rule-based inference languages [10] that expresses rules of reasoning, such as a logic description [25] used in SWRL (Semantic Web Rule Language)[3]. For instance, with SWRL rules and a rule-based engine[4] to run these rules, from the participant's language is French (as low-level context information sensed in his profile), it can be inferred that television programs speaking in French or translated to it (as high-level context information) are among his preferences. This can be expressed by the next SWRL rule: `Participant(?p)` **`and`** `hasLanguage(?p,"french")` **`implies`** `hasPreferencesTV(?p, "FrenchEmissions").`

[3] www.w3.org/Submission/SWRL/

[4] such as Java Expert System Shell (JESS: www.jessrules.com)

Another example, the high-level context "*ForecastWeather = isHarsh*" is derived from temperature, wind-speed or likelihood of rain as low-level context that it can be expressed with a SWRL rule, for wind-speed or likelihood cases, as follow:

```
Weather(?w) and (hasWindSpeed(?w, ?Wind)
and swrlb:greaterThan(?Wind, 25)) or
(hasLikelihoodRain(?w, ?Rain) and
swrlb:greaterThan(?Rain, 10)) implies
ForecastWeather(?w, "isHarsh").
```

It should be noted that the sensed contextual information (at low-level) is allowed by two types of context providers: Local Source and Remote Source. The former is for local providing of context information that is available at client (mobiledevice) or server (application) sides. For instance, an area of interest exists in profile that represents context information to inform any conference participant by all communications according to his research area. The latter is for other context information that can be provided by a third-party (e.g. Web services return temperature, likelihood of rain, wind speed, TV programs, ...).

In these ontologies, Figure 4 (a and b) shows, respectively, the use of relationships and attributes (ObjectProperty and DataTypeProperty stereotypes) using the Ontology UML Profile notation according to OWL metamodel [19].

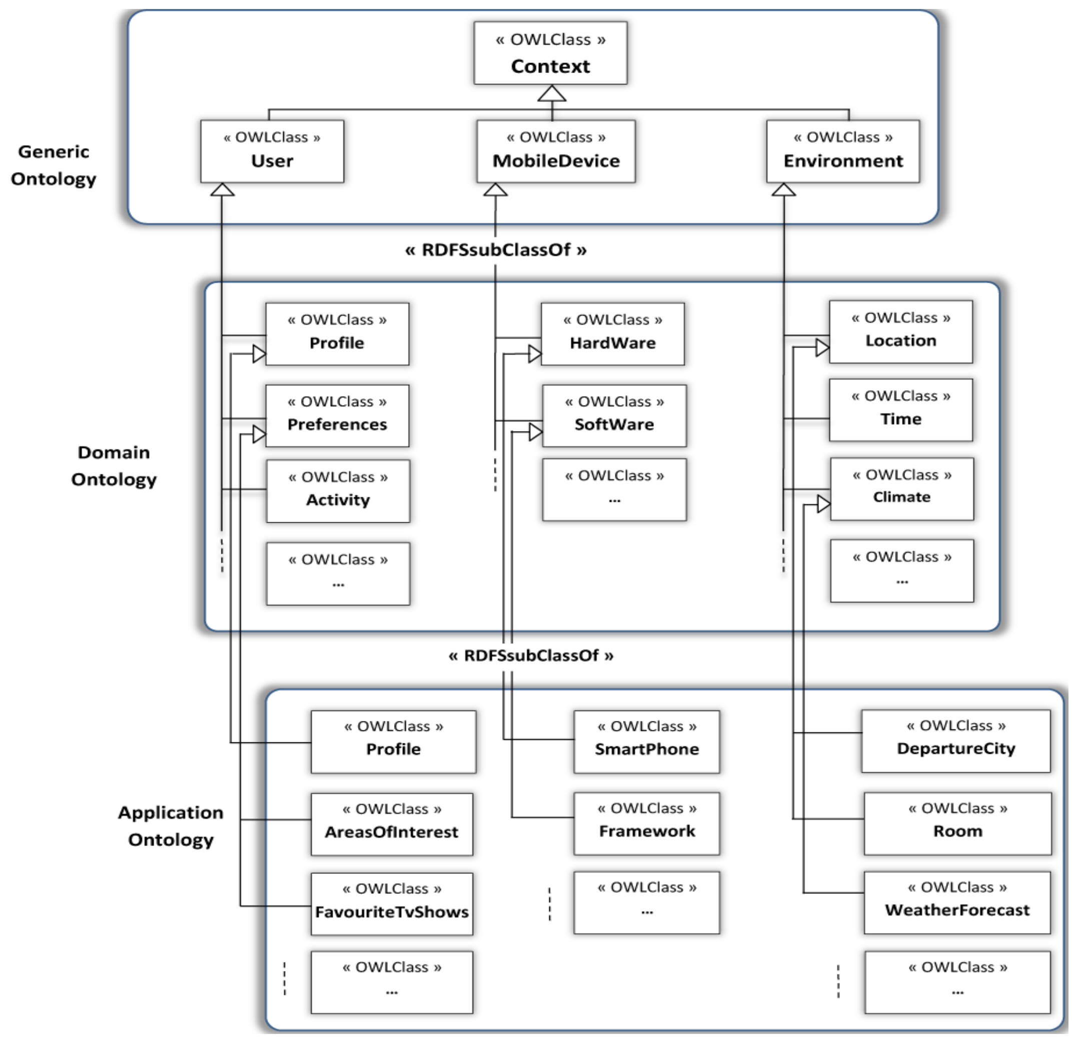

Figure 3. An Ontology-based Context Model

The proposed ontology-context model satisfies the requirements cited above, in which the generic level ontology gives a high level abstraction by proposing general entities to be used by different user-centered mobile applications. Whereas, the domain ontology level allows covering all main contextual entities proposed in the literature of context modelling and the application ontology is to specify the context in enclosing application space. The Domain and Application ontologies are sufficiently flexible allowing to cover any context entity of pervasive situation surrounding the user and to enable taking into account new specific entities to a given application domain. Moreover, by the reasoning and inferring capabilities on these ontologies, we can to derive other contextual information that could complete context describing this situation.

5. Related Work

There are few works using hybrid approach combining ontologies and MDD for representing the context information in pervasive environment. To our best knowledge, there are only two works. In [26], Ou & al. have examined how OMG's Model Driven Architecture (MDA) can be applied to tackle the issues of context modelling and Context-Aware Application (CAA) modelling and developing. A Context Ontology Model (COM) is presented to model context information at two levels: upper-level (ULCOM) and extended specific level (ESCOM). Using RDFS/OWL metamodels, ULCOM captures ontology of concepts that are essential for generically characterizing context in the pervasive services domain under three main entities, Entity, EntityProperty, and EntitySpecification. ESCOM defines specific concepts and their properties for context as extensions of ULCOM entities. For instance, PDA, laptop, PC, mobile-phone normally are used in a pervasive computing environment. Also in this work, a Model Driven Integration Architecture (MDIA) is proposed to integrate rigorous model specifications and generate CAA implementations either automatically or semi-automatically. The principal advantage of this approach is to explore the feasibility of amalgamating UML, MDA and ontology languages (such as RDFS and OWL) towards context ontology modelling and MDA-based integration architecture for automatic development of context-aware applications aiming at improving accuracy and reducing time and costs. Finally, a case study is presented in [27], focusing on the development of a context-aware m-commerce tourist application that demonstrates in detail how the proposed approach works in practice.

Also, Vale et Hammoudi, in [23], have presented an architecture for the development of mobile context-aware applications based on Ontology Definition Metamodel (ODM) in which a context and context-aware metamodels are exposed for use in model-driven development process.

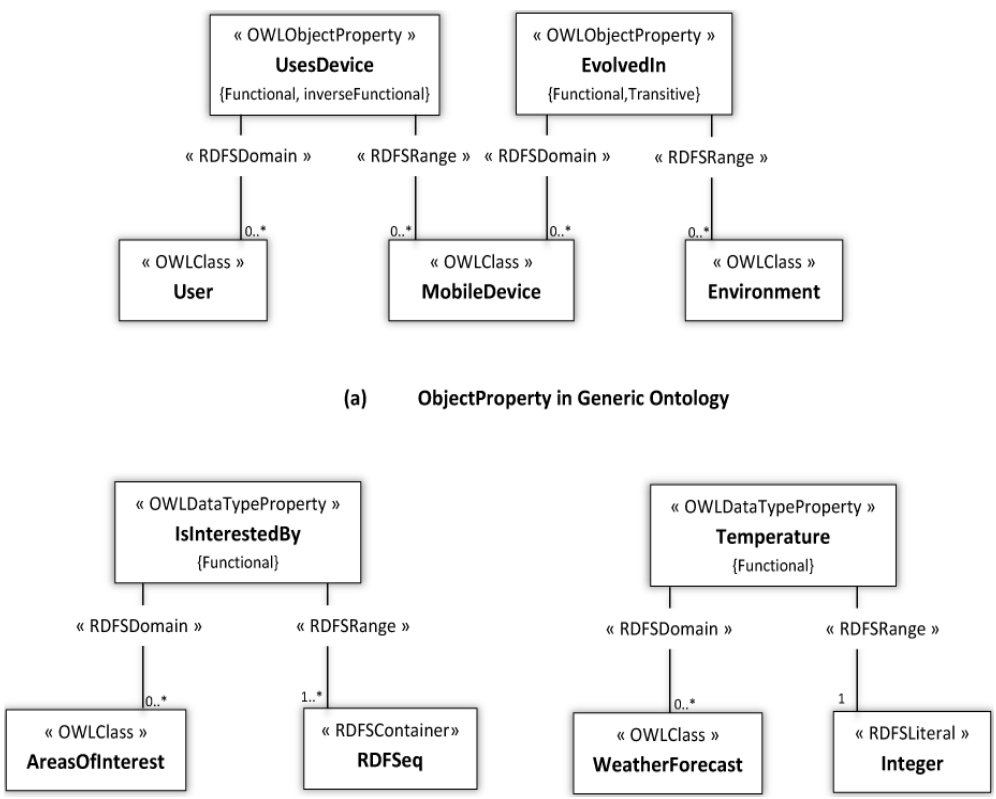

Figure 4. Context Properties Modelling

Otherwise, there are several propositions of ontology-based model context in the literature [9]. Chen & al. have developed CoBra-Ont [28], an OWL ontology which is a collection of ontologies for describing places, agents, events and their associated properties in an home area intelligent environments. This work is registered in their Context Broker Architecture (CoBrA) that provides knowledge sharing, context reasoning, and privacy protection supports for pervasive context-aware systems.

Based on the previous ontology, the same authors presented another context ontology in [29]. Standard Ontology for Ubiquitous and Pervasive Applications(SOUPA) provides knowledge sharing, reasoning on context and interoperability in a pervasive and ubiquitous environment. SOUPA deals with more areas of pervasive computing than CoBra-Ont and map many of its concepts to concepts of existing common ontologies like friend-of-friend (FOAF)[5] ontology to enable interoperability by using ontology mapping constructs OWL standard (owl: equivallentClass and owl: equivallentProperty). It is constituted of two sets of ontologies called SOUPA-Core and SOUPA-Extensions. SOUPA-Core defines a generic vocabulary which is universal for different pervasive computing applications (e.g., person, agent, policy, time, space), while SOUPA-Extensions define additional ontologies to support specific kind of applications and provide examples for extending future ontologies (e.g., home, office, attraction).

The CONtext ONtology (CONON) [30] is context OWL ontology for modelling context in pervasive computing environment and supporting logic based context reasoning. Wang & al. have proposed an upper context ontology that captures general concepts about basic context such as location, activity, person or computational entity, and domain specific ontologies for detaining these general concepts and their features in each sub-domain covered in a hierarchical manner. Also and based on this context ontology, the authors have studied the use of logic reasoning to check the consistency of context information, and to reason over low-level, explicit context to derive high-level, implicit context. The notions of upper ontology and domain-specific ontologies are discussed in similar work [31] for presenting Service-Oriented Context-Aware Middleware (SOCAM).

In their project called Context-Driven Adaptation of Mobile Services (CoDAMoS)[6] and suiting some requirements of mobile computing, Prenveneers & al. have presented an adaptable and extensible context ontology for creating context-aware computing infrastructures, ranging from small embedded devices to high-end service platforms [32]. It is defined around four modelling entities Users, Environment, Platforms and Services. This ontology has been designed with the aim of solving many challenges, such as application adaptation, automatic code generation, code mobility, and generation of device-specific user interfaces to allow interoperability in an Ambient Intelligence environment.

Strang & al. [33] describe a context modelling approach using ontologies as a formal foundation. They introduce their Aspect-Scale-Context (ASC) model and show how it is related to other models. A Context Ontology Language (CoOL) is derived from the model, which is used to enable context-awareness and contextual interoperability during service discovery and execution in a distributed architecture.

6. Conclusion and Future Work

To model context information in ubiquitous environment, we presented, in this paper, a new conceptualisation using ontologies. Our extensible context model uses Model-Driven Development with its MDA standard and ontologies with the ODM specification. The MDD makes easy the development of context-aware application and the ontologies for their extensibility in ubiquitous environments. The ODM specification defines an Ontology UML Profile that makes this combination possible by bridging the gap between them.

To cover all context information for these applications, an ontology-based model is proposed with three levels of ontologies; Generic and Domain ontologies for popular and shared concepts used by context-aware applications developers, and Application ontology for a specific description of context in close space.

As future work on the dynamic adaptation of these applications, we will present a context-awareness mechanism based on the present ontology-based context model and focused on Aspect-Oriented Modelling (AOM).

[5]http://www.foaf-project.org/
[6]http://distrinet.cs.kuleuven.be/projects/CoDAMoS/

References

[1] Weiser, M. (1991). The computer for the 21st century, Human-computer interaction: toward the year 2000, 0, 933–940.

[2] Bazire, M., Brézillon, P. (2005). Understanding context before using it, *In*: Modeling and Using Context, 5th International and Interdisciplinary Conference, CONTEXT 2005 (A. K. Dey, B. N. Kokinov, D. B. Leake, and R. M. Turner, eds.), vol. 3554 of Lecture Notes in Computer Science, p. 29–40, Springer.

[3] Dey, A. K. (2001). Understanding and using context, *Personal and Ubiquitous Computing*, 5, 4–7.

[4] Schilit, B., Theimer, M. (1994). Disseminating active map information to mobile hosts, *IEEE Network*, 8, 22–32.

[5] Pascoe, J. (1998). Adding generic contextual capabilities to wearable com-puters. *In*: 2nd International Symposium on Wearable Computers (I. C. Press, ed.), (Los Alamitos, California), p. 92–99.

[6] Sheng, Q. Z., Benatallah, B. (2005). Contextuml: A uml-based modeling language for model-driven development of context-aware web services, *In*: Proceedings of 2005 International Conference on Mobile Business (ICMB 2005), (Sydney, Australia), p. 206–212, 11-13 July.

[7] Streitz, N., Nixon, P. (2005). Introduction, Commun. ACM, 48, 32– 35, Mar.

[8] Strang, T., Linnhoff-Popien, C. (2004). A context modeling survey, *In*: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp - The Sixth International Conference on Ubiquitous Comput-ing, Nottingham/England.

[9] Ye, J., Coyle, L., Dobson, S., Nixon, P. (2007). Ontology-based models in pervasive computing systems, Knowl. Eng. Rev., 22, 315–347, December.

[10] Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D. (2010). A survey of context modelling and reasoning techniques, *Pervasive Mob. Comput*., 6, 161–180, April.

[11] Kapitsaki, G. M., Prezerakos, G. N., Tselikas, N. D., Venieris, I. S. (2009). Context-aware service engineering: A survey, *Journal of Systems and Software*, 82 (8) 1285–1297.

[12] Kapitsaki, G. M., Prezerakos, G. N., Tselikas, N. D. (2010). In book Enabling Context-Aware Web Services: Methods, Architectures, and Technologies, ch. Context-aware Web Service Development: Methodo-logies and Approaches, p. 3–29. Chapman & Hall/CRC Press, 1st ed..

[13] Gruber, T. R. (1993). A translation approach to portable ontology specifications, *Knowledge Acquisition*, 5, 199–220.

[14] OMG. (2009). Ontology definition metamodel, version 2.0, <http://www.omg.org/cgi-bin/doc?formal/09-05-01.pdf>.

[15] Guarino, N. (1998). Formal Ontology in Information Systems, *In*: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy. Amsterdam, The Netherlands, The Netherlands: IOS Press, 1st ed.

[16] Drummond, N., Shearer, R. (2006). The open world assumption. electronic.

[17] Boudaa, B., Camp, O., Hammoudi, S., Chikh, M. A. (2012). Model-Driven development of Context-Aware services: Issues, techniques and review, *In*: International Conference on Information Technology and e-Services 2012 (ICITeS'2012), (Sousse, Tunisia), p. 160–167, Mar.

[18] Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing, *Int. J. Hum.-Comput. Stud*., 43, 907–928, Dec.

[19] Gasevic, D., Djuric, D., Devedzic, V. (2009). Model Driven Engineering and Ontology Development. Springer Publishing Company, Incorporated, 2nd ed.

[20] McGuinness, D., van Harmelen, F. (2004). Owl web ontology language overview, w3c recommendation, feb., <http://www.w3.org/tr/owl-features/>.

[21] de Bruijn, J. (2003). Using Ontologies - Enabling Knowledge Sharing and Reuse on the Semantic Web, Technical Report DERI-2003-10-29, DERI.

[22] Ye, J., Stevenson, G., Dobson, S. (2011). A top-level ontology for smart environments, *Pervasive and Mobile Computing*, 7 (3) 359 – 378. Knowledge-Driven Activity Recognition in Intelligent Environments.

[23] Vale, S., Hammoudi, S. (2009). Odm-based architecture for the development of mobile context-aware applications, *In*: Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments, PETRA '09, (New York, NY, USA), p. 9:1–9:4, ACM.

[24] Chandrasekaran, B., Josephson, J. R., Benjamins, R. V. (1999). What are Ontologies and why do we need them?, *IEEE Intelligent Systems*, p. 20–26, Jan.

[25] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., Patel-Schneider, P. F. eds. (2003). The description logic handbook: theory, implementation, and applications. New York, NY, USA: Cambridge University Press.

[26] Ou, S., Georgalas, N., Azmoodeh, M., Yang, K., Sun, X. (2006). A model driven integration architecture for ontology-based context modelling and context-aware application development, *In*: Model Driven Architecture - Foundations and Applications (Rensink, A., Warmer, J. eds.), V. 4066 of Lecture Notes in Computer Science, p. 188–197, Springer Berlin / Heidelberg.

[27] Georgalas, N., Economou, D. (2009). Model-driven development of pervasive applications using context and ontologies: An m-commerce case study, *The Electronic Journal for Emarging Tools and Applications, Special Issue Electronic Commerce in Pervasive Environments*, 3 (1).

[28] Chen, H., Finin, T., Joshi, A. (2003). An ontology for context-aware pervasive computing environments, *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, 18, 197– 207.

[29] Chen, H., Perich, F., Finin, T., Joshi, A. (2004). Soupa: Standard ontology for ubiquitous and pervasive applications, *In*: International Conference on Mobile and Ubiquitous Systems: Networking and Services, p. 258–267.

[30] Wang, X. H., Zhang, D. Q., Gu, T., Pung, H. K. (2004). Ontology based context modeling and reasoning using owl, *Pervasive Computing and Communications Workshops, IEEE International Conference on*, V. 0, p. 18.

[31] Gu, T., Wang, X. H., Pung, H. K., Zhang, D. Q. (2004). An ontology-based context model in intelligent environments, *In*: Proceedings of CNDSMS Conference, p. 270–275.

[32] Preuveneers, D., Bergh, J. V. D., Wagelaar, D., Georges, A., Rigole, P., Clerckx, T., Berbers, E., Coninx, K., Bosschere, K. D. (2004). Towards an extensible context ontology for ambient intelligence, *In*: Proceedings of the Second European Symposium on Ambient Intelligence, p. 148–159, Springer-Verlag.

[33] Strang, T., Linnhoff-Popien, C., Frank, K. (2003). Cool: A context ontology language to enable contextual interoperability, *In*: Volume 2893 of Lecture Notes in Computer Science (LNCS)., Paris/France, p. 236–247, Springer Verlag.