A Context Ontology for Mobile Environments

María Poveda-Villalón, Mari Carmen Suárez-Figueroa, Raúl García-Castro, Asunción Gómez-Pérez
Ontology Engineering Group. Departamento de Inteligencia Artificial.
Facultad de Informática, Universidad Politécnica de Madrid.
Campus de Montegancedo s/n.
28660 Boadilla del Monte. Madrid. Spain
mpoveda@delicias.dia.fi.upm.es, {mcsuarez, rgarcia, asun}@fi.upm.es

Abstract. The widespread use of mobile devices is leading to a next generation of applications that exploit user contextual information to provide a richer experience. One of the activities to perform during the development of these context-aware applications is to define a model to represent and manage context information. Currently, there is a lack of consensual models, and this supposes a handicap when developing these applications. This paper presents a context ontology network to model context-related knowledge that allows adapting applications based on user context. We describe the methodological process followed during the ontology development as well as the ontology network obtained from this process. Besides, we provide an example of how to extend the ontology for a particular use case in a concrete domain.

Keywords: context, ontology network, context ontology, mobile

1 Introduction

The number of mobile devices is increasing exponentially and these devices provide more powerful functionalities over time. In the ambient intelligence environment, mobile devices, services and agents cooperate to support end users in performing different tasks, and this forces any application developed under the ambient intelligence paradigm to be aware of contextual information and to be able to automatically adapt to context changes.

The development of context-aware applications should be supported by adequate context modelling and reasoning techniques. Modelling context knowledge is a crucial task to support the delivery of the right information at each moment, the adaptation and personalization of the information, and the anticipation of the results.

By context, we refer to any information that can be used to characterize the situation of an entity, where an entity can be a person, a place or a (physical or computational) object [5]. As context can be considered a specific kind of knowledge, it can be modelled as an ontology. Ontology-based models of context allow (a) representing complex context knowledge and (b) providing a formal semantics to context knowledge, which supports the sharing and/or integration of context information.

Although there are a variety of context ontologies developed with different approaches [2, 3, 4, 8, 12, 13, 18], no consensual model exists that can be broadly reused for modelling context in applications. Furthermore, even if there have been plenty of efforts for developing context ontologies, only a few of them are available to be studied in detail and reused [2, 3, 4, 8, 12].

Our objective in this paper is to present the mIO! ontology network, a context ontology in the mobile environment that aims to represent contextual knowledge about the user that can influence his interaction with mobile devices. In this environment the user is able to interact with both services provided by companies and services created and provided by himself by means of his mobile device.

The remainder of the paper is structured as follows. Section 2 summarizes the state of the art on context ontologies. Section 3 briefly describes the methodology followed during the mIO! ontology network development, that is, the NeOn Methodology. Then, Section 4 presents a brief overview of the whole ontology development process of our context ontology. Next, Section 5 presents in detail the mIO! ontology network and Section 6 presents one of the extensions we developed. Finally, Section 7 concludes and shows some future lines of work.

2 State of the Art

Currently, different approaches have been used for building context ontologies. However, there is no widely accepted model that can be reused for modelling context knowledge in different applications. Furthermore, even if there have been plenty of efforts for developing context ontologies, only some implementations of these ontologies are available [2, 3, 4, 8, 12]. In this section we summarize in an alphabetical order the most well-known context ontologies, without taking into account whether there are implementations available.

The ASC (for Aspect-Scale-Context) model [13] includes as core concepts the following ones: aspects, scales, and context information. Each aspect aggregates one or more scales, and each scale aggregates one or more context information. This ontology is especially useful to describe concepts related to measurement units; however, it is not very useful to describe more abstract context information, like user activities.

The CC/PP (for Composite Capabilities/Preference Profile) model [8] is a W3C initiative that proposes an infrastructure to describe device capabilities and user preferences. This model can be used to guide the adaptation of the content presented to the device. The model architecture is based on profiles and uses RDF as implementation language. The representation model consists of a hierarchical structure of components divided into the following three areas: hardware, software and application.

COBRA-ONT [3] is an ontology that defines some of the common relationships and attributes that are related to people, places and activities. The main objective of this ontology is to enable knowledge sharing and ontology reasoning within the CoBra (for Context Broker Architecture) infrastructure. COBRA-ONT defines key ontology categories such as action, agent, time, space, device, etc.

The CoDAMoS [12] ontology defines four main core entities: user, environment, platform, and service. This ontology has been designed with the aim of solving the following challenges: application adaptation, automatic code generation, code mobility, and generation of device-specific user interfaces.

CONON [18] (for Context Ontology) defines general concepts such as location, activity, person or computational entity, whose terms are thought to be extensible in a hierarchical way by adding domain specific concepts. The authors divide their context model into an upper ontology and a specific ontology. On the one hand, the upper ontology is a high-level ontology that captures general features of basic contextual entities. On the other hand, the specific ontology defines the details of the general concepts and their features in each subdomain covered.

The Delivery context [2] ontology provides a formal model of the environment characteristics in which different devices interact with concrete services. This ontology includes the device characteristics, the software used to access the service and the network providing the connection, among others. The main entities modelled in this ontology are environment, hardware, software, location, and measure.

SOUPA [4] is divided into two main blocks called SOUPA-Core and SOUPA-Extensions. SOUPA-Core defines concepts that should appear in a lot of scenarios (e.g., person, agent, policy, time, space), while SOUPA-Extensions supports particular concepts in narrower domains (e.g., home, office, entertainment).

As we can observe in Table 1[1] none of the available context ontologies covers all the subdomains that must be modelled in the mIO! ontology network. In fact, there are subdomains such as Interface, Provider, and Source that are not covered by any of the ontologies studied.

Table 1. Subdomains addressed by each available context ontology.

Ontology / Subdomain	CC/PP [8]	COBRA-ONT [3]	CoDAMoS [12]	Delivery Context [2]	SOUPA [4]
Device	X	X	X	X	
Environment		X	X	X	
Interface					
Location		X	X	X	X
Network				X	
Provider					
Role		X	X		
Service			X		
Source					
Time		X	X		X
User	X	X	X		X

[1] This table shows whether the available context ontologies address the eleven subdomains that must be represented in the mIO! ontology network.

3 Summary of the NeOn Methodology

When ontology developers think about the use of ontologies for solving a particular problem, a provisional[2] work team of ontology developers (if possible involving ontology engineers, software developers, domain experts, and final users) should be established. Such a team will be concerned at least with the following activities: environment and feasibility study, knowledge acquisition, ontology requirements specification, and scheduling.

After the provisional team has been established, the team should carry out an environment and feasibility study. This allows them to decide whether ontologies should be developed or not for the specific problem.

Once such a decision has been taken, the provisional team should start with the knowledge acquisition activities. These activities should be carried out during the whole development; however, ontology developers should acquire most of the knowledge at the beginning of the ontology development.

Simultaneously with the knowledge acquisition activities, ontology developers should specify the requirements that the ontology should fulfil, by means of the ontology requirements specification activity. The objective of this activity is to output the ontology requirements specification document (ORSD) that includes the purpose, the scope and the implementation language of the ontology network, the target group and the intended uses of the ontology network, as well as the set of requirements that the ontology network should fulfil.

After the ontology requirements specification activity, it is advisable to carry out a quick search for knowledge resources, using as input those terms included in the ORSD. Then, the scheduling activity must be carried out, using the ORSD and the results of the previous search. During the scheduling activity, the team establishes the ontology network life cycle and the human resources needed for the ontology project.

Then, the ontology developers assigned to the ontology project should carry out (1) the required scenarios for the ontology project being developed as well as (2) the conceptualization, formalization, and implementation of the ontology.

It is worth mentioning that the NeOn Methodology considers a set of 9 scenarios [14] for the ontology development. Concretely, there is a basic and mandatory scenario ("Scenario 1: From specification to implementation") that can be combined with the rest of scenarios that are identified as follows:

- Scenario 2: Reusing and reengineering non-ontological resources
- Scenario 3: Reusing ontological resources
- Scenario 4: Reusing and reengineering ontological resource
- Scenario 5: Reusing and merging ontological resources
- Scenario 6: Reusing, merging and reengineering ontological resources
- Scenario 7: Reusing ontology design patterns
- Scenario 8: Restructuring ontological resources
- Scenario 9: Localizing ontological resources

[2] This first team can be considered as provisional because the definitive team is established during the scheduling activity.

4 mIO! Ontology Network Development

The development of the mIO! ontology network has been performed following the NeOn Methodology [14], briefly described in Section 3. In this section, we describe a subset of the activities carried out.

We performed the ontology requirements specification activity [16] that refers to the activity of collecting the requirements that the ontology should fulfil (e.g., the reasons to build the ontology, the identification of target groups and intended uses, etc.) [17].

As the output of this activity we obtained the ORSD[3] for the mIO! ontology network. Such document explains, among other things, that the ontology should represent the domain of the user context with the aim of adapting services to the user according to his context, and that its main users will be users that interact with mobile devices and service providers. The ontology will be used mainly for storing and editing information about users, including their contexts, profiles, and roles, services, providers, and devices. Also, it establishes that the ontology must be implemented in the ontology representation language OWL[4] DL to get the maximum expressiveness without losing computational completeness. In addition, the ORSD includes the ontology requirements divided into non-functional requirements and functional requirements. As an example of non-functional requirements we can mention that the ontology has to follow a modular architecture and that its language should be English.

Regarding functional requirements, we would like to note that they were written in two different ways. On the one hand, most of the requirements were written in the form of competency questions (CQs) [7]. These are questions that the ontology must answer; for example, "Where is the device X? The device Z is at coordinates X, Y". On the other hand, some requirements difficult to express as competency questions were written to describe the domain characteristics. These characteristics are natural language sentences; for example, "A device belongs to one or more environments".

As it can be observed in [1], at early stages of the mIO! ontology development process there were 202 functional requirements divided into 119 competency questions and 83 domain characteristics. Due to the revisions made to the requirements in following iterations, the second version of the ontology requirements was composed by 205 competency questions and 161 domain characteristics.

The next activity carried out within the mIO! ontology network was the scheduling one [16], which refers to the activity of identifying the different activities and processes to be performed during the ontology development, their arrangement, and the time and resources needed for their completion.

Taking into account the ORSD and the result of the quick search for possible knowledge resources to be reused, we divided the ontology network development into three iterations. For each iteration, we selected the scenarios shown in Table 2 to be carried out in combination with Scenario 1.

---

[3] The whole ORSD for the mIO! ontology network is available in [10].
[4] http://www.w3.org/TR/2004/REC-owl-guide-20040210/

It can be observed that we focus our ontology network development in the reuse of knowledge resources. We planned to reuse both ontological and non-ontological resources in the first and second iterations, respectively. In addition, the reuse of ontology design patterns is included in all the iterations along the development.

Table 2. Relationship between iteration and scheduled scenarios to be carried out.

	Iteration		
Scenario	1st	2nd	3rd
Scenario 2: Reusing and reengineering non-ontological resources		X	
Scenario 3: Reusing ontological resources	X		
Scenario 7: Reusing ontology design patterns	X	X	X
Scenario 8: Restructuring ontological resources		X	X
Scenario 9: Localizing ontological resources			X

5 mIO! Ontology Network Description

In this section we present the resulting ontology after carrying out the first and second iterations scheduled for the ontology development process introduced in Section 4. It should be noted that the result of the first iteration can be found in [1]. Also, it is worth mentioning that by the time of writing this document, the first and second iterations have been completed and the third iteration is still ongoing.

The goal of the mIO! ontology network is to represent knowledge related to context as a whole, e.g., information on location and time, user information and its current or planned activities, as well as devices located in his surroundings. The ontology aims at solving the challenge of adapting the applications based on the user context. Since the context knowledge is quite broad, the mIO! ontology network consists of a core ontology that interlinks different ontologies. Such ontologies describe the different subdomains needed for modelling context knowledge.

Fig. 1 presents the current high-level conceptual model of the mIO! ontology network obtained as a result of the second iteration. This model contains eleven modular ontologies: User, Role, Environment, Location, Time, Service, Provider, Device, Interface, Source, and Network. This conceptual model can be considered the mIO! core. The figure also includes those knowledge resources that were reused for building some ontologies as well as the connections between the ontologies by means of relationships.

The current version[5] of the mIO! ontology network has been implemented in the OWL DL ontology language and contains 433 classes, 277 object properties, 156 datatype properties, 364 instances, and has a SROIQ(D) expressiveness. It should be added that such version fulfils 91 competency questions and 96 general characteristics. The main differences between the first and the second version of the mIO! ontology network are that the second one (a) includes the Source subdomain among those modelled, (b) covers 54 requirement more than the first version, and (c) reuses more types of knowledge resources, both ontological and non-ontological ones.

[5] http://www.oeg-upm.net/index.php/es/ontologies/82-mio-ontologies

Next, we present a brief description of each of the subdomains presented in Fig. 1 as well as the knowledge resources that were reused in each ontology.

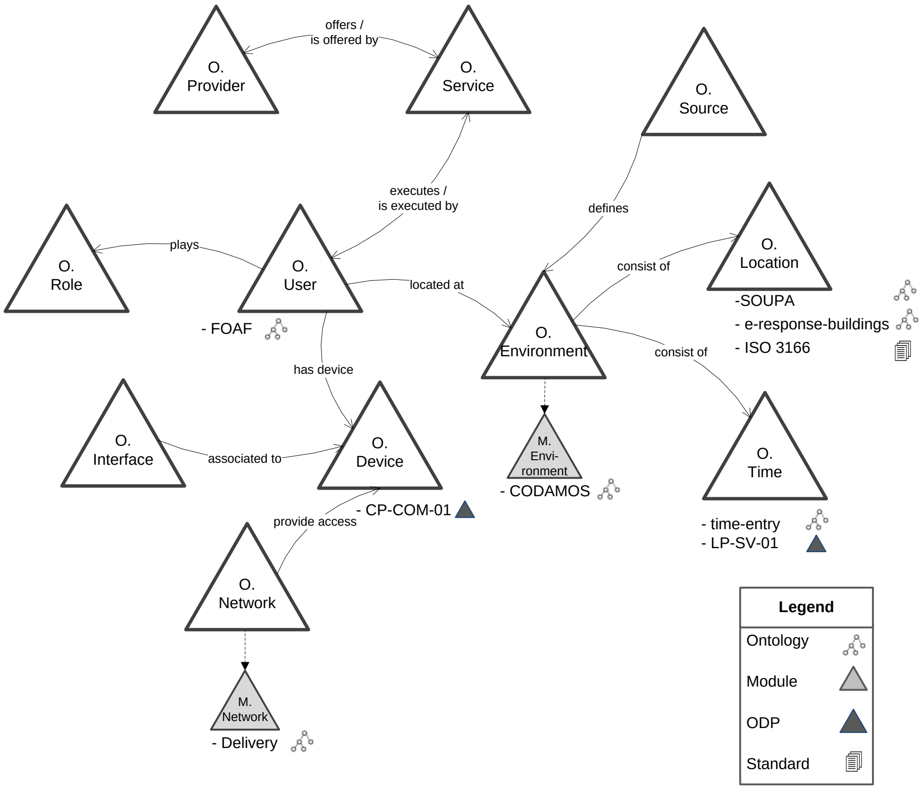

Fig. 1. mIO! ontology network conceptual model.

- **Device ontology**. It models knowledge about devices and includes a wide range of them as well as their characteristics. This model also includes knowledge about the charging mode of the devices and their compatibility with standards. It is worth mentioning that during the development of this ontology we have reused the "*Componency*" pattern (*CP-COM-01* [11]) to represent that objects either are proper parts of other objects, or have proper parts. In addition, we have created a wide categorization of devices reusing the "*Taxonomy*" pattern (*AP-TX-01* [15]) as Fig. 2 shows.
- **Environment ontology**. It models knowledge about environments, which includes environmental conditions such as humidity, luminosity, noise, etc. During the development of the environment ontology, we have reused an ontology module[6] from *CoDAMoS* in order to obtain only the relevant knowledge for our model. In this way, we have obtained a core model about

[6] We used the "Ontology Module Extraction" plug-in of NeOn Toolkit (http://neon-toolkit.org/wiki/Ontology_Module_Extraction) during the module extraction activities.

environments and their relationship with the abovementioned environmental conditions.

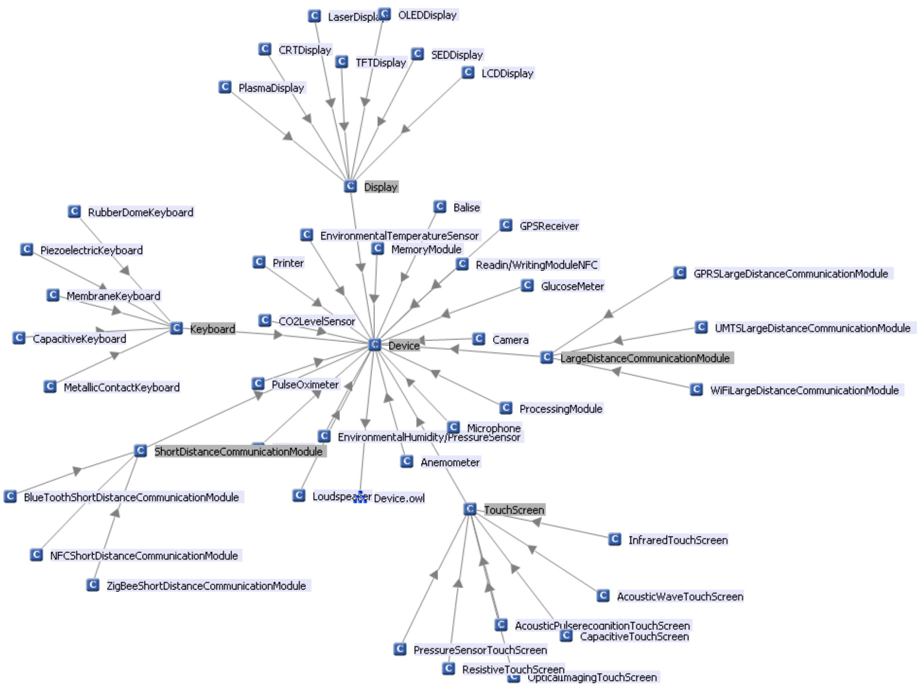

Fig. 2. Taxonomy of devices.

- **Interface ontology**. This ontology models knowledge about the user interfaces, their types, modalities and characteristics, which can be provided by the different devices. This model also distinguishes between input and output interface modalities.
- **Location ontology**. It models knowledge about locations such as buildings, location coordinates, spatial entities, distance, countries, etc. During the development of this ontology we have reused ontologies as *SOUPA*, which has been described in Section 2, in order to include information about spatial things and units of area and distance. We have also reused and pruned the e-response-buildings ontology, which provides a wide categorization of buildings, with the aim of extending the building classification represented in the *SOUPA* ontology. Finally, we have reused and made reengineering over the non-ontological resource *ISO3166*[7], which represents consensuated knowledge about countries and their *ISO3166* codes. In addition, the concept `Country` from the *ISO3166* has been matched to the existing concept `Country` within the mIO! ontology network.

[7] http://www.iso.org/iso/en/prods-services/iso3166ma/index.html

- **Network ontology**. It models knowledge about communication networks including network topologies, operators and administrators, accessibility, price systems, coverage, etc. During the development of this ontology, we have reused an ontology module from the *Delivery Context* ontology in order to obtain only the relevant knowledge for our model. The *Delivery Context* ontology, described in Section 2, provides, among other things, information about network types and modes.
- **Provider ontology**. This ontology contains a wide categorization of the service providers including both those that offer a set of services in a business (e.g. a hosting service) and those that offer mIO services[8]. Also, this ontology models simple or aggregated service providers depending on whether a provider is an aggregation of some of them.
- **Role ontology**. It models knowledge about roles, profiles, preferences, etc. It reuses an ontology about profiles that personalizes a role in a given situation. This ontology also reuses the *Reco*[9] ontology to represent the preferences of a particular user.
- **Service ontology**. This ontology contains a wide range of services including both those that are typically offered in a business (e.g. a food service) and mIO! services. As we can observe in Fig. 3, a mIO! service has a digital signature and consists of components which has functionalities. Also, a mIO! service acts as capacity which has input and output parameters. In addition, a mIO! service has a title, a description and tags as attributes.

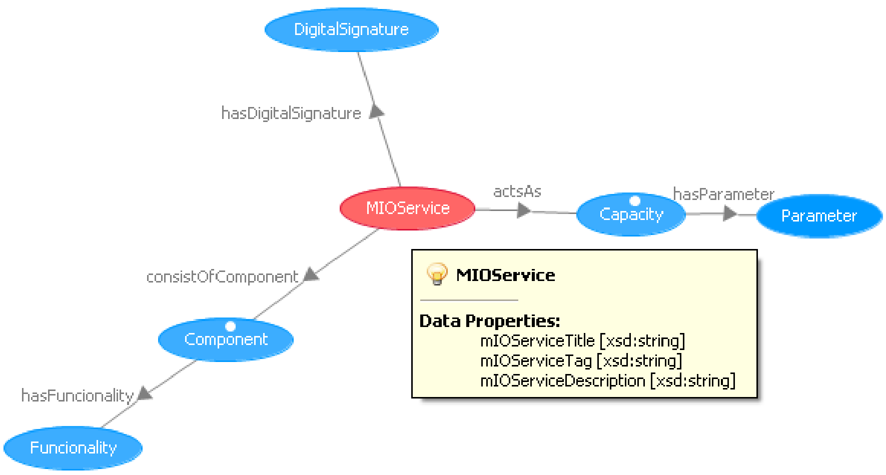

Fig. 3. Excerpt of the mIO! service model.

---

[8] A mIO! service is a context-sensitive software entity, which has a user interface. This type of services is executed on a server embedded on a mobile or a network providing a value to a user.

[9] http://ontologies.ezweb.morfeo-project.org/reco/spec

- **Source ontology**. This ontology models knowledge about context sources which are elements that provide context information. Several elements within the ontology could act as a context source, for example, a user, a device, a service, etc. The context sources could be simple or aggregated depending on whether a source is an aggregation of some of them.
- **Time ontology**. It models knowledge about time such as temporal units, temporal entities, instants, intervals, etc. This ontology has been mainly developed by reusing the *OWL Time* ontology, which is a reimplementation of the *DAML time* ontology[10] carried out by the *Semantic Web Best Practices and Deployment Working Group*[11] (SWBPD) from the W3C. In addition, this ontology has been extended by reusing the "*Specified Values: Set of Individuals*" (*LP-SV-01* [15]) pattern that allows representing an enumeration of *n* individuals (which are different between them). This pattern has been reused to model the days of the week. Accordingly to the abovementioned pattern the class `WeekDay` is defined as the enumeration of the instances: `Monday`, `Tuesday`, `Wednesday`, `Thursday`, `Friday`, `Saturday` and `Sunday`.
- **User ontology**. It models knowledge about users, groups, organizations, etc. This ontology includes knowledge about a user such as his employment status, skills, mobility pattern, and online identities. The *FOAF*[12] ontology as a whole has been reused during the development of this ontology.

In addition, it is worth mentioning that ontology design patterns have been also reused within the whole ontology network. For example, we have reused the "*Modular Architecture*" (*AP-MD-01* [15]) pattern as it is shown in Fig. 1. Besides, as we can see in Fig. 4, the "*N-ary Relation: New Class*" (*LP-NR-01* [15]) pattern that has been used to represent locations at a given point in time.

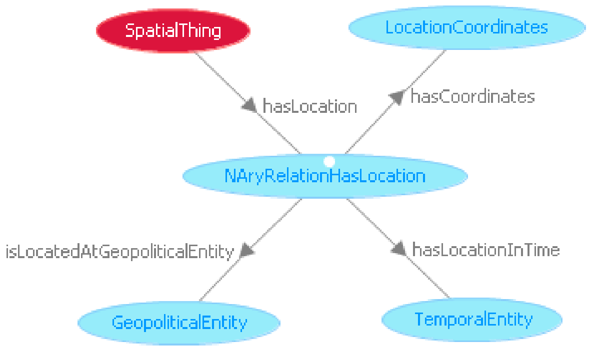

Fig. 4. N-ary pattern applied to modelling locations at a given point in time.

[10] http://www.cs.rochester.edu/~ferguson/daml/
[11] http://www.w3.org/2001/sw/BestPractices/
[12] http://www.foaf-project.org/

Finally, we can classify the abovementioned ontologies in the different types of ontologies identified in the literature [6] based on the subject of their conceptualization. The ontologies in the mIO! ontology network can be classified into application ontologies, domain ontologies, and general ontologies, as Fig. 5 shows in the form of a pyramid.

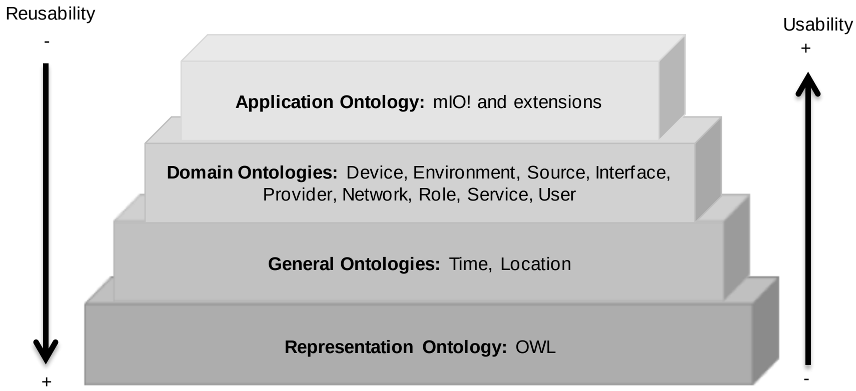

Fig. 5. Pyramid of reusability and usability applied to the mIO! ontology network.

- **Application ontologies** have to be built for a certain application. In our case, we can classify in this type of ontologies the mIO! ontology network itself and the possible extensions that are developed ad-hoc for a certain use case.
- **Domain ontologies** are specific of a certain area or domain, but still model general aspects of that domain recurrent in more specific applications. In this sense, we consider here the following ontologies: Device, Environment, Source, Interface, Provider, Network, Role, Service, and User.
- **General ontologies** model universal or domain-independent areas that can be used across different domains of knowledge. Within the mIO! ontology network, we can identify the ontologies of Time and Location in this category.

It is worth mentioning that the more general the ontology is, the more possibilities for reuse there will be, and the less usable the ontology will be in order to achieve a particular goal. On the contrary, the more specific the ontology is, fewer possibilities for reuse exist, but the more usable the ontology is.

6 Paddle Extension

This section presents how we have reused the mIO! ontology network, presented in Section 5, to develop an application ontology for a concrete use case. This application ontology is the result of extending the mIO! ontology network with specific knowledge of the use case about paddle described in [9].

The first step was to obtain a textual description of the use case domain written in natural language and to extract ontological requirements from the text. Some of the ontological requirements extracted from the paddle use case are:

1. Paddle clubs offer services to create paddle matches at their sport facilities.
2. A paddle player can either play at the right or the left, or at both sides.
3. A paddle player has a certain skill level.
4. A paddle player has a certain temporal availability to play paddle matches.
5. A paddle player has a certain spatial availability to play paddle matches.
6. A paddle player can visualize his matches in poles installed at the sport facilities.

Next, we identified which ontologies should be reused to fulfill the extracted requirements. To that end, we have analyzed the requirements and matched them in terms of the mIO! ontology network elements. Once the ontology elements involved in the requirements are identified, we obtained which ontologies are involved in the use case.

Fig. 6 shows an example of this process, taking the first requirement as starting point. Generalizing for the whole use case, the following ontologies were involved in the paddle use case: Device, Environment, Location, Provider, Role, Service, Source, Time, and User.

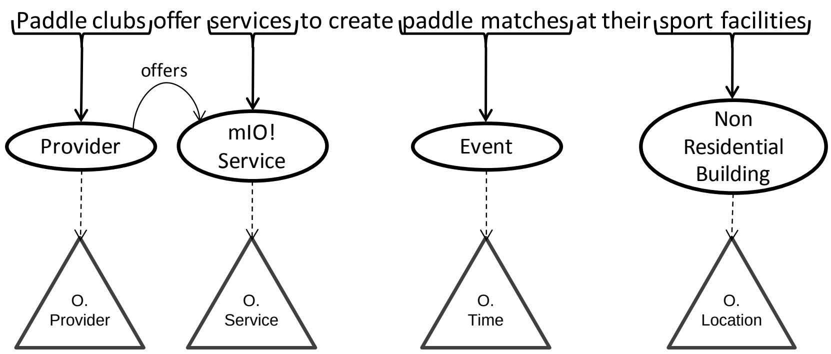

Fig. 6. Relations among terms, ontology elements, and ontologies.

Once the ontologies involved in the extension were identified, we selected those that should be restructured [16] to cover the new requirements. In the paddle use case was necessary to restructure the Location, Device, Time, and Role ontologies.

Finally, Fig. 7[13] shows the result of the restructuring process carried out to adapt the mIO! ontology network to the abovementioned requirements extracted from the paddle use case. For example, we have extended the ontology which models roles with knowledge about characteristics related to the profile of a paddle player such as which level he has or which side of the field he prefers. This ontology has been connected to the time ontology by modelling the information about when a paddle player has availability to play. Also, the role ontology has been connected to the

[13] In this figure each ontology is represented by means of a rectangle with rounded corners. The figure shows the mIO! ontology network core, which includes the Location, Device, Time and Role ontologies, and the mIO! paddle extension. The rectangles and named arrows represent, respectively, concepts and relationships in a given ontology. Finally, arrows that have a triangle in one extreme represent "subclassOf" relationships whereas dotted arrows represent that a given element belongs to the ontology which are connected to.

location ontology to represent that a paddle player is used to play in certain locations. The ontology about location has been extended with sport facilities, specifically with paddle facilities.

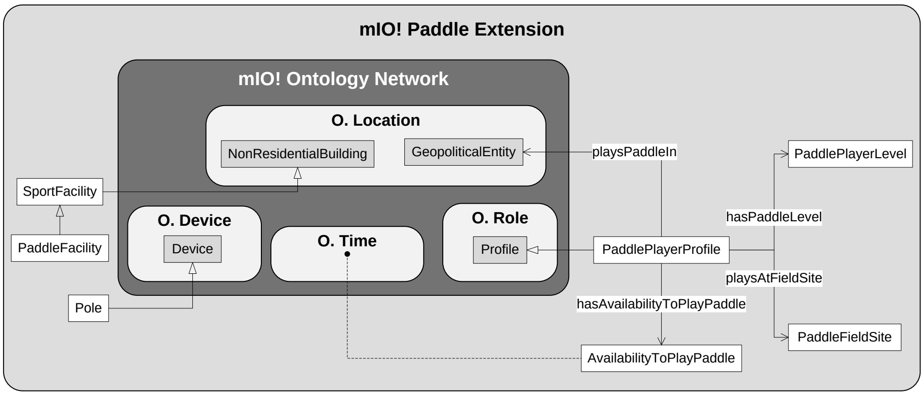

Fig. 7. mIO! paddle extension.

At last, we show an example of instantiation of the paddle extension. Fig. 8 represents an excerpt of a scenario described within the use case about paddle [9]. This scenario is about a semi-professional paddle player called Rosa. As we can observe in Fig. 8 Rosa is very skilled playing paddle, in fact, she is a good player at both sides of the field. It is also represented that Rosa has availability to play paddle both at afternoons and mornings. Finally, we can observe that Rosa creates and provides two services called “learnToPlayWithMe” and “improveYourPaddleSkills”.

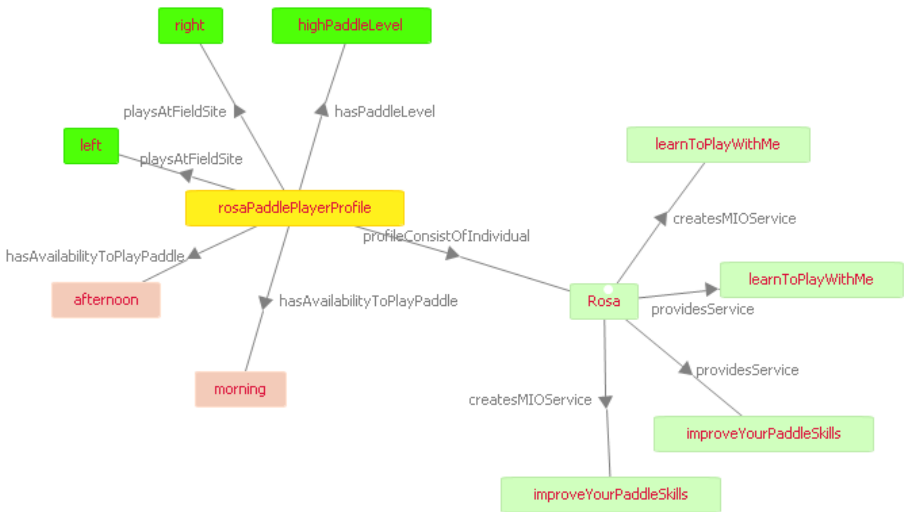

Fig. 8. Excerpt of the mIO! paddle use case.

7 Conclusions

This paper presents a context ontology network including details about its methodological development and focusing on the reuse of different types of knowledge resources such as ontologies, ontology design patterns and non-ontological resources. The main purpose of this context ontology, called mIO! network ontology, is to represent the user context to be able to process and use it to configure, discover, execute and enhance different services that the user may be interested in.

The paper provides an overview of the ontology network, by means of a general conceptual model and a set of modelling details about some parts of the ontology network.

Finally, an extension of the ontology network for a concrete use case is presented. This point includes some notions of how we have reused the mIO! ontology network to develop an application ontology for a concrete use case and the resulting ontology network.

After the development of the mIO! ontology network, we have realized (a) the usefulness of following a methodology to guide ontology development, (b) the advantages of reusing knowledge resources, in particular ontology design patterns, ensuring the use of good practices in the ontology development, and (c) the difficulty of reusing context ontologies within a new ontology development because of the different purposes and requirements for which the ontologies are developed.

Our next steps aim to carry out the third iteration of the mIO! ontology network development to obtain the final version of the ontology that fulfils all the defined requirements. This third version will be evaluated not only according to ontology requirements but also from a user perspective and a modelling perspective. Also, we aim to provide guides and examples of how to reuse the mIO! ontology network to adapt it for a concrete use case or application.

The real benefit of using ontologies for context information in pervasive computing environments lies in the interoperability of different devices. This benefit will not become effective before there is a widely-accepted standard context ontology. We think that the approach introduce in this paper is a good starting point for the future work to establish such a standard.

Acknowledgments. This work has been supported by the Spanish project *mIO!* (CENIT-2008-1019). We would like to thank Elena Montiel-Ponsoda for her comments and revisions.

References

1. Cadenas, A., Ruiz C., Larizgoitia I., García-Castro R., Lamsfus C., Vázquez I., González M., Martín D., Poveda M. *Context Management in Mobile Environments: a Semantic Approach*. Workshop on Context, Information and Ontologies (CIAO 2009). 1 June, 2009. Heraklion, Greece.
2. Cantera-Fonseca, J.M., Lewis, R. *Delivery Context Ontology*. W3C Working Draft. June 2009.

3. Chen, H., Finin T. *An Ontology for Context Aware Pervasive Computing Environments*. Special Issue onOntologies for Distributed Systems, Knowledge EngineeringReview, 2003.
4. Chen, H., Finnin, T., and Joshi, A.. *The SOUPA Ontology for Pervasive Computing*. In Ontologies for Agents: Theory and Experiences, 233-258. 2005.
5. Dey, A. K. 2001. *Understanding and using context*. Personal and Ubiquitous Computing Journal, 5(1), 5–7.
6. Gómez-Pérez, A., Fernández-López, M., Corcho, O. *Ontological Engineering*. November 2003. Springer Verlag. Advanced Information and Knowledge Processing series. ISBN 1-85233-551-3. 2003.
7. Gruninger, M., Fox, M. S. *The role of competency questions in enterprise engineering*. In Proceedings of the IFIP WG5.7 Workshop on Benchmarking - Theory and Practice, Trondheim, Norway, 1994.
8. Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M.H., Tran, L. *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0*. W3C Recommendation. http://www.w3.org/TR/CCPP-struct-vocab/. January 2004.
9. mIO! Project Technical Report. *E7.1.2: Escenarios mIO!*. March 2010.
10. Poveda, M. *Master Thesis. Metodología NeOn Aplicada a la Representación del Contexto*. Spain. Universidad Politécnica de Madrid. To appear in September, 2010.
11. Presutti, V., Gangemi, A., David, S., Aguado de Cea, G., Suárez-Figueroa, M.C., Montiel-Ponsoda, E., Poveda, M.. *NeOn D2.5.1: A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies*. NeOn project. http://www.neon-project.org. February 2008.
12. Preuveneers, D., Van Den Bergh, J., Wagelaar, D., Georges, A., Rigole, P., Clerckx, T., Berbers, Y., Coninx, K., Jonckers, V., De Bosschere, K. *Towards an Extensible Context Ontology for Ambient Intelligence*. 2004.
13. Strang, T., Linnhoff-Popien, C., Frank, K. *CoOL: A Context Ontology Language to enable Contextual Interoperability*. In 4th Int'l Conf. on Distributed Applications and Interoperable Systems, pp. 236–247, Nov 2003.
14. Suárez-Figueroa, M.C. *PhD Thesis: NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse*. Spain. Universidad Politécnica de Madrid. June 2010.
15. Suárez-Figueroa, M.C., Brockmans, S., Gangemi, A., Gómez-Pérez, A., Lehmann, J., Lewen, H., Presutti, V., Sabou, M.. *NeOn D5.1.1: NeOn Modelling Components*. NeOn project. http://www.neon-project.org. March 2007.
16. Suárez-Figueroa, M.C., Gómez-Pérez, A. *First Attempt towards a Standard Glossary of Ontology Engineering Terminology*. 8th International Conference on Terminology and Knowledge Engineering (TKE 2008). 18-21 August 2008. Copenhagen.
17. Suárez-Figueroa, M.C., Gómez-Pérez, A., Villazón-Terrazas, B. *How to write and use the Ontology Requirements Specification Document*. In Proceedings of the 8th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE 2009). On the Move to Meaningful Internet Systems: OTM 2009. (Robert Meersman, Tharam Dillon, and Pilar Herrero) (ISBN: 978-3-642-05150-0). LNCS 5871. Volume: Part II Pp: 966-982. Vilamoura, Algarve-Portugal. November 2009.
18. Wang, X., Zhang, D. Q., Gu, T., Pung, H. K. *Ontology based context modeling and reasoning using OWL*. Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference, pp. 18-22. 2004.