# A context information manager component for dynamic environments

Jérôme Euzenat[1] , Jérôme Pierson[2], Fano Ramparany[2]

[1]INRIA Rhône-Alpes, 655 avenue de l'Europe, Montbonnot St Martin, 38334 Saint-Ismier, France
Jerome.Euzenat@inrialpes.fr
[2]France Telecom R&D, 28 chemin Vieux Chêne 38240 Meylan, France
{Jerome.Pierson, Fano.Ramparany}@francetelecom.com

**Abstract.** *In a pervasive environment, heterogeneous devices need to communicate in order to provide services adapted to users. We have developed an extensible context model using web semantic technologies and a context information manager component that enable the interaction between context information producer devices and context information consumer devices and as well as their insertion in an open environment.*

## 1. Introduction

In a pervasive computing environment, various basic services can be provided by smart devices (sensors, actuators, human-computer interface…). To provide more elaborate services, they have to act together and cooperate. Then they can offer an infinite number of services. It would be better if the devices could adapt their behaviour to the user, his/her preference and his/her task than if users have to find the specific service he wants among all the smart devices.

This idea requires the perception of the environment in which devices and users interact. There are pieces of information that can be considered common to all services. In particular, spatial and temporal location as well as information related to the physical environment in which services are made available [1, 2]. These elements are part of the context in which applications operate. We are here concerned with context-aware applications, i.e., application whose behavior is determined to some extent by the context.

Our goal is to design a context management system general enough for being used by different pervasive computing applications, specific enough for encompassing already existing services and application, and flexible enough for supporting the dynamic addition of new devices.

First we introduce our proposition for a distributed architecture to manage context information (§2), then we will define a context representation (§3) which is independent of applications and an architecture enabling their evolution. The openness of the system will lead to deal with heterogeneous representations that will have to be reconciled before being used. For that purpose, we will take advantage of solutions developed for the "semantic web".

## 2. A context information manager component

Context is the set of information (partly) characterizing the situation of some entity [3]. The notion of context is not universal but relative to some situation, task or application [4, 5]. Pervasive Computing applications retrieve context data directly or indirectly from sensors, which are grounded in the physical environment. We propose an architecture in which applications won't need to directly connect to each sensor available and where adding a new sensor won't require all applications to be recompiled and redeployed.

Designing architecture for hosting context-aware services, suggests the development of a context management service for providing other services or devices context information [6, 7, 11]. In our approach, each device (or his proxy if it cannot embed enough computing resource) or service embeds a context management component for maintaining context information for its own use or for the benefit of others (Fig. 1).

This component provides mechanisms for helping context-aware devices to request context information from context sensitive devices. For this purpose we design a protocol to enable devices to identify a service, know what kind of context information it could provide and interact with it in order to get access to this information. Thus the context management component provides few methods. A first method allows identifying devices that are available in the environment. The identifier can then be used to contact the device. Alternatively, it could be used to get a more detailed description of the device (e.g. in case the identifier is a URI pointing to a network location where a description of the identified object is stored). A second method identifies the class (in OWL terminology) of the device. In theory, this class should be accessible from the network and once its definition is found, it provides a detailed description of the device. A third method provides device's description (or rather that of context information they provide) information in a OWL like language (aka OWL-S). A fourth method is used to post queries to the devices and to get the context information returned.

Thus any device is able to: find out in its environment, services that are able to provide information relevant to its own context, get features of services that have been found (for example, measurements precision), connect to the selected service to get the information sought.

We need a language to describe the context model of heterogeneous devices so that these device can interact in a dynamic environment.

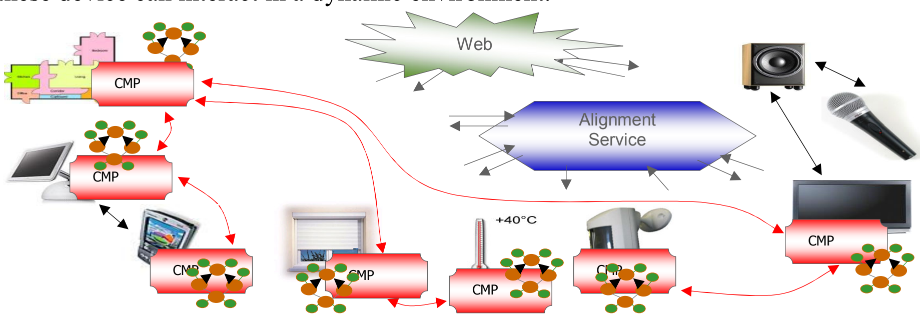


Fig. 1: Each device embeds a context manager component (CMP) and a semantic description of his context.

## 3. Context model

We first introduce the semantic web technologies that we use to model and represent context information. Then, we present the simple context model that we will use for targeting pervasive computing applications. These are based on semantic web languages.

The ground language for the semantic web is RDF (Resource Description Framework [8]). It enables expressing assertions of the form subject-predicate-object. The strength of RDF is that the names of entities (subjects, predicates or objects) are URIs (the identifiers of the web that can be seen as a generalization of URLs: http://www.w3c.org/sw). This opens the possibility for different RDF documents to refer precisely to an entity (it is reasonable to think that a URI denotes the same thing for all of its users).

The OWL language [9], has been designed for expressing « ontologies » or conceptual models of a domain of knowledge. It eases the interpretation of RDF graphs concerning this domain. OWL provides schemas to define classes of objects and predicates and to declare constraints applying to them (i.e., that the « output » of a « thermometer » is a « temperature »).

The context model that we will use at that stage is very simple: a context is a set of RDF assertions. Interoperability is guaranteed through considering that context-aware devices are consumer and producer of RDF. However, this is not precise enough and devices might want to extract only the relevant information from context sources. For that purpose, a language like RDQL [10] will be useful for querying or subscribing to context sources. In order to ask the relevant queried to the adequate components, it is necessary that components publish the OWL classes of objects and properties on which they can answer.

The languages developed for the semantic web, and particularly RDF and OWL, are adapted to context representation in pervasive computing and particularly to the representation of dynamically evolving contexts for two reasons: these languages are open: they implement the open world assumption under which it is always possible to add more information to a context characterization; and they have been designed to work in a networked way.

If we can add components at any time, it is not clear that they are easily usable. Indeed, there is no reason, a priori, components available, new applications and new sensors are really compatible. Fortunately, using the knowledge representation techniques that are integrated in OWL language it is possible to introduce new devices in the environment by extending the ontology, through specifying a new concept or a property. The applications must be as general as possible to describe the information they need whereas the context management system must be as precise as possible on what information it makes available. This approach enables the most specialized applications to take advantage of them. The essential point is to have sufficiently generic ontologies to cover the various concepts implied in pervasive computing applications [12 ].Unfortunately this is not always the case and agreeing on standard universal and self contained context ontology is not a reasonable assumption. This raises the issue of matching context information produced and applications context information requirements. There are three alternative approaches to address interoperability in pervasive computing environments: (i) A priori

standardisation of ontologies, (ii) setting up mediators among ontologies and (iii) a dynamical ontology alignment service. These three approaches are not incompatible and might even be jointly used.
We propose to set up one (or more) ontologies alignment service(s) (Fig. 2). The goal of such service is to help agents (context managers in our case) to find a matching between different ontologies. These services provide mechanisms for finding out ontologies close to a given ontology, archiving (and retrieving) past alignments, dynamically computing matching between two ontologies and translating queries and responses to queries between context managers that use different ontologies.

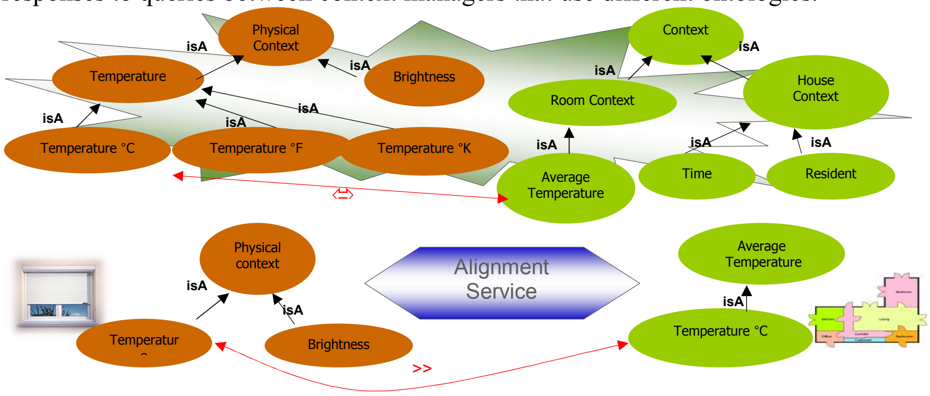


Fig. 2: To find correspondence between his model and the context information provider, the window service ask to an alignment service to translate his model to an another device model.

## 4. Conclusion and perspectives

Our objective is to address specifically the dynamic aspects of context management. This problem has been tackled by providing a distributed component based architecture and by using semantic web technologies. Components enable the addition, at any moment, of new devices that can provide information about the context of applications. The use of RDF and OWL ensures interoperability between components developed independently by taking advantage of both the open character of these technologies and the work on ontology alignment. The proposed approach relies on a minimal commitment on basic technologies: RDF, OWL, some identification protocol
We are currently developing a demonstrator of this technology. Ontology alignment is also available and will be integrated in the system. Our future work will focus on developing the decentralized architecture described above. We will develop a toolkit for developers of pervasive applications which help them deploy a distributed context management system. This toolkit provides a component to manage (search, diffuse and update) context information.

## 5. References


1. R. Guha, Contexts : a formalization and some applications, PhD thesis, Stanford university (CA US), 1991 (Technical Report STAN-CS-91-1399-Thesis et MCC ACT-CYC-423-91).
2. O. Khriyenko, V. Terziyan, Context description framework for the semantic web, Proceedings Context 2005 Context representation and reasoning workshop, Paris (FR), 2005
3. A. Dey, D. Salber, G. Abowd, A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, *Human-Computer Interaction* 16:97-166., 2001
4. P. Dourish, Seeking a foundation for context-aware computing, *Human-Computer Interaction*, 16(2-3), 2001.
5. M. Chalmers, A Historical View of Context, *Computer supported cooperative work* 13(3), 223-247, 2004.
6. A. Dey, Understanding and using context, *Personal and ubiquitous computing* 5(1):4-7, 2001.
7. J. Coutaz, J. Crowley, S. Dobson, D. Garlan, Context is key, *Communications of the ACM* 48(3):49-53, 2005.
8. G. Klyne, J. Carroll, Eds., Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation, 2004 http://www.w3.org/TR/rdf-concepts/
9. M. Dean, G. Schreiber Eds, OWL Web Ontology Language: Reference, W3C Recommendation, 2004. http://www.w3.org/TR/owl-ref/
10. A. Seaborne, RDQL — A Query Language for RDF, W3C Member submission, 2004. http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/
11. J. Crowley, J. Coutaz, G. Rey, P. Reignier, Perceptual components for context aware computing, Proceedings International Conference on Ubiquitous Computing, Göteborg (SW), pp. 117-134, 2002.
12. X. H. Wang, D. Q. Zhang, T. Gu, H. Keng Pung, Ontology based context modeling and reasoning in OWL, Proceedings 2nd International conference on pervasive computing and communication Workshop on Context Modeling and Reasoning (CoMoRea), Orlando (FL US), 2000