Experiences from Developing a Context Management System Applied to Mobility

Baptiste Gaultier and Jean-Marie Bonnin

Institut TELECOM, TELECOM Bretagne, France
Université européenne de Bretagne
{baptiste.gaultier,jm.bonnin}@telecom-bretagne.eu

Abstract. Recent advances in electronic and automotive industries as well as in wireless telecommunication technologies have drawn a new picture where each vehicle became "fully networked". In order to provide IP connectivity to on-board devices, the IETF has proposed the NEMO (NEtwork MObility) protocol. In this approach, a new device, the Mobile Router (MR), will take place in vehicles. It has to manage mobility and takes advantage of the surrounding wireless technology diversity to offer connectivity and reachability for all nodes in the mobile network as it moves.

To be efficient, the MR has to take into account various contextual parameters regarding the management of wireless network interfaces and the routing of the flows. Exchanging such a contextual information can be achieved easily through basic polling and broadcasting mechanisms. However, systems involving more than one MR and systems having hot sensor plugging capabilities will require more advanced techniques for exchanging context information. In [2], we proposed to use a CMS (Context Management System) in order to process and exchange contextual information in a vehicular network. This paper describes our experience with the design and the implementation of a new CMS applied to mobility.

Keywords: Middleware, Context-awareness, Mobile network, NEMO.

1 Introduction

Wireless communications for ITS (Intelligent Transport System) is an enabling technology to improve driving safety, reduce traffic congestion, and support information services to vehicles. In fact, continuous connectivity can make their transportation time more pleasant (browsing the web and watching online videos) and efficient (consulting emails on the way to work or read online newspaper during a train travel).

A major step in this evolution was the introduction of a new embarked component responsible for managing all communications of the vehicle: the Mobile Router (MR). The NEMO (NEtwork MObility) Basic Support protocol designed by the IETF manages mobility and offers continuous and seamless IPv6 connectivity to on-board Mobile Network Nodes (MNN in fig. 1). The CALM architecture[1]

[1] Continuous Air interface for Long and Medium range (ISO TC204 WG16).

Y. Cai et al. (Eds.): Mobilware 2010, LNICST 48, pp. 292–298, 2010.

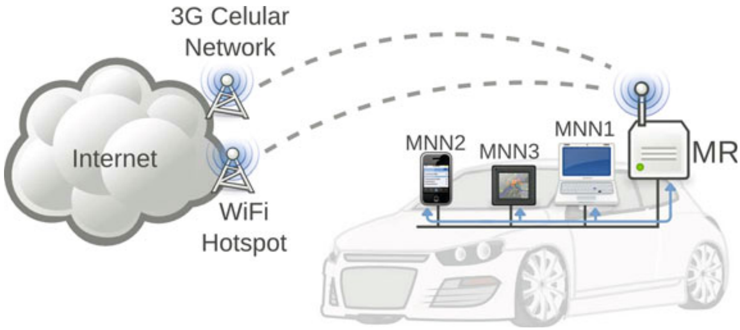

Fig. 1. Example of a Mobile Network

relies on IPv6 and NEMO to offer continuous communication between vehicles and the Internet infrastructure. This architecture introduces a distributed management plan which allows more advanced features such as simultaneous use of multiple wireless networks, per-flow routing, application adaptability to network conditions and advanced interface management.

In order to provide such services, CALM-compliant systems provide the ability to use the most appropriate access technology for information exchange. Selection rules are supported that include contextual information (e.g., current network environment, current position, current speed, etc.) which can be gathered from other devices or sensors available in the mobile network. User preferences and access technology capabilities can be taken into account in making decisions as to which access technology to use for a particular session, and when to handover between access technologies or between service providers on the same access technology. In [2], we came to a conclusion that a more convenient way to process and exchange these contextual information is to use a Context Management System (CMS).

The purpose of this paper is to share our experience in designing and implementing such a CMS applied to mobility.

The structure of the paper is as follows. Section 2 present the architecture we propose and analyzes a number of requirements for a distributed context management system suitable in a mobility context. Section 3 describes the implementation choices we made in order to provide a suitable context management system targeting adaptive applications in a mobility environment. Finally, Section 4 closes with our conclusions.

2 Requirements for a CMS Applied to Mobility

In a previous paper [2] we have shown that none of studied CMS based solutions is suitable for a mobility framework. [2] has also highlighted a set of important functionalities which are important in the design of a context management system applied to network mobility. The purpose of this section is to analyze these requirements in order to present and justify the design choices we have made.

The designed system aims at providing a convenient way to exchange and reason contextual information in a vehicular network.

2.1 General Design Requirements

In a vehicular network, we have many types of contextual information exchanged: network environment provided by the MR, current position, current speed,... which can be gathered from other devices or sensors available in the mobile network. Moreover, multiple mobile nodes can access to the shared context available. In order to deal with this kind of exchange we propose to use the *Context Server* Approach described in [1]. This distributed approach extends the *Middleware Infrastructure* [1] by introducing mechanisms to gather context from distributed sources. We propose to integrate the so-called *Context Repository* (see fig. 2) on the MR in order to facilitate concurrent multiple access to the context information. The usage of the *Context Server* approach has the advantage of relieving clients of resource intensive operations. This is important because embedded mobile nodes can be resource poor (GPS, smartphone ...).

Figure 2 shows the architecture and components of the solution we propose.

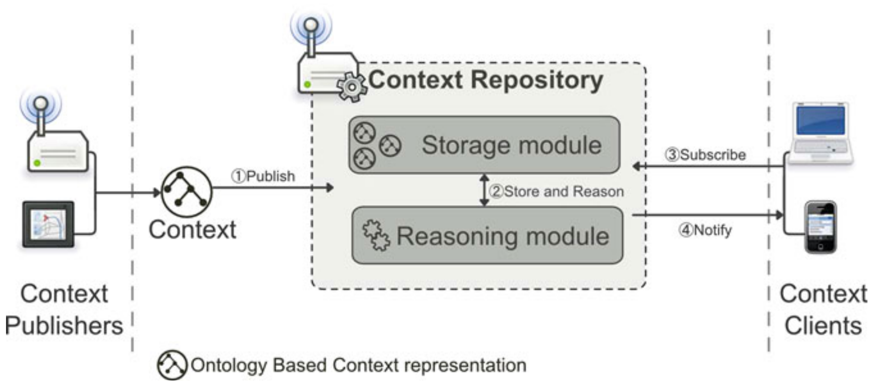

Fig. 2. Context Management System architecture and interactions

2.2 Context Discovery Requirements

In [2], we expressed the need to dynamically discover context providers and context repositories. Indeed, when a mobile node comes into the mobile network, he has to discover context information that match with the users requirements (e.g. network environment, localization, presence...). In our solution, we propose to use a centralized approach where context publishers have to publish their context to a repository called *Context Repository* (see fig. 2). In our context, we can imagine the following types of context advertised by producers in a vehicular network:

– Location data: a GPS or a smartphone with a localization capability can provide this kind of information and publishes these to *Context Repository*
– Network environment information: delay, bandwidth available, jitter can be computed by the mobile(s) router(s) and added to the context representation.

Moreover, rather than using a CMS exclusively for mobility management, it can be interesting to widen its purpose to offer context management for all the in-vehicle applications.

2.3 Distribution of Context Requirements

Context management systems adhere generally to the publish/subscribe paradigm where producers publishes information, and an event notification is sent to all authorized subscribers. All mobile nodes can be both context publisher and consumer (e.g. a smartphone can act as a context client wanting to know the network environment and a context publisher for transmitting location information). In general, the relationship between the publisher and subscriber is mediated by a service that receives publication requests, broadcasts event notifications to subscribers, and enables privileged entities to manage lists of applications that are authorized to publish or subscribe. The focal point for publication and subscription is a "node" to which publishers send data and from which subscribers receive event notifications. In our architecture this focal point is situated on the Mobile Router and its called *Context Repository* (see fig. 2).

2.4 Context Modeling Requirements

The context modeling refers to the requirement for formatting the context information. Information modeling offers a convenient way to define, store and reason this data in order to guarantee compatibility among the possibly heterogeneous devices (i.e. GPS, smartphones, sensors, computers...). The context modeling is particularly important in a vehicular network where mobile nodes are not a priori aware of each other. We propose to use an ontology based model in our solution to take advantage of their high and formal expressiveness. Moreover, ontologies enable reasoning and decision-making mechanisms which are very useful in deducing entailed context information from different sources of context (i.e. Vertical reasoning and/or Horizontal reasoning).

3 Implementation Choices

In addition to the general requirements described above, we have to address the following specific requirements we find out in our previous paper [2]:

3.1 Support for Discovery

When a mobile node is introduced into a vehicular network, its first task is to dynamically discover the network. A common approach in a managed network is to use DHCP (Dynamic Host Control Protocol) but the complexity to manage such a server based solution in a vehicular context led us to look for a solution without configuration or server. Therefore, we studied the IETF Zeroconf set of mechanisms which propose to solve the network discovery problems:

1. Allocation of IP network addresses for networked devices (link-local address auto-configuration)
2. Automatic resolution and allocation of computer hostname
3. Service discovery

In this paper, we use the last capability to discover the *Context Repository* (see fig. 2) offered by each MR as a service. This service can be advertised thanks to Zeroconf dynamic announcement mechanisms (DNS-SD [6]).

3.2 Context Exchange Mechanisms

As we said before, our architecture meets the publish-subscribe model. To implement this feature, we propose to use XMPP (Extensible Messaging and Presence Protocol [4]) and its publish-subscribe extension [3] which provides a convenient way to exchange context in the vehicular network. This protocol addresses problems we highlighted in [2]: First, publish-subscribe extension deals with the privacy and security issues. XMPP publish-subscribe extension contains a hierarchy of affiliations for the purpose of authorization and access control to the data (i.e. context information). Another interesting point is the ability of XMPP to offer fail-safe mechanisms (e.g. context repository database can be replicated and stored on multiple nodes).

3.3 Context Modeling and Reasoning

By introducing context-awareness in vehicular network, applications become increasingly complex and interconnected. This raises the need for context modeling. The conclusion of the evaluation presented in [5] show that ontologies are the most suitable model because of its high and formal expressiveness and the possibilities for applying ontology reasoning mechanisms. As we mentioned before, we want to express the concept of physical location which can be represented as 'location', 'place', 'position' etc. To be able to interpret and reason such a context information, a context model is needed to capture concept unambiguously.

We propose to use the Web Ontology Language OWL [8] which is a flexible, extensible, expressive and common language to describe a ontology. To represent the types of contexts described in Section 2, we are interested in the Delivery Context Ontology [7] which provides a formal model of the characteristics of the environment in which devices interact with the Web or other services. The Delivery Context includes the characteristics of the Device, the software used to access the service and the Network providing the connection among others.

3.4 Implementing the Architecture

The Architecture described in Figure 1 was designed and implemented in order to test the performance and the viability of the system. The *Context Repository* was developed in Python and tested on a desktop computer. The *Context Client* software that can both publish and subscribe context information was implemented in two programming language: Python and Objective-C. This client was then tested on a wide range of devices spanning from resourceful laptop (running Mac OS X) to smartphone (running iPhone OS). The *Context Client* provides also a context visualizer with a graphical user interface which allows a user or a developer to dynamically monitor and produce (publish) context information (shown in fig. 3, when deployed on both a smartphone and a laptop).

Fig. 3. The left picture depicts a network context viewed on an smartphone, while the right picture depicts a localization context on a laptop

4 Conclusions

In conclusion, this paper proposes a distributed context management system which provides exchanging and reasoning mechanisms. We have detected a number of both general and more specific requirements imposed by the mobility aspect. In this respect, we have proposed an approach which follow the publish-subscribe paradigm for forming loosely coupled entities and for advertising their context information. We argue that this approach fulfills the requirements we find out to a great extend. Furthermore, this architecture has been implemented, tested, and evaluated in real world applications, on both resourceful (laptops) and resource poor (smartphone) computer with significant success.

References

1. Baldauf, M., Dustdar, S.: A survey on context-aware systems. International Journal Of Ad Hoc And Ubiquitous Computing (2004)
2. Gaultier, B., Rayana, R.B., Bonnin, J.-M.: Context management systems applied to mobility. In: 9th International Conference on ITS Telecommunications RSM - Dépt. Réseaux, Sécurité et Multimédia (Institut Télécom-Télécom Bretagne, ITST (2009)
3. Millard, R.M.P., Saint-Andre, P.: Xmpp publish-subscribe extension. XMPP Draft XEP-0060 (2009), http://xmpp.org/extensions/xep-0060.html
4. Saint-Andre, P.: Extensible messaging and presence protocol (xmpp). RFC 3920 (October 2004), http://www.ietf.org/rfc/rfc3920.txt
5. Strang, T., Linnhoff-Popien, C.: A context modeling survey. In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England (2004)
6. Cheshire, M.K.S.: Dns-based service discovery. IETF working draft (September 2008), http://files.dns-sd.org/draft-cheshire-dnsext-dns-sd.txt
7. W3C. Delivery context ontology. W3C Recommendation (2009), http://www.w3.org/TR/dcontology/
8. W3C. Owl 2 web ontology language. W3C Recommendation (2009), http://www.w3.org/TR/owl-overview/