Enhanced Presence Tracking for Mobile Applications*

Marko Luther, Sebastian Boehm, Matthias Wagner
DoCoMo Communications Laboratories Europe
Munich, Germany

Johan Koolwaaij
Telematica Instituut
Enschede, The Netherlands

Abstract

We present an early prototype for semantic-based monitoring of mobile users from our work in the IST project MobiLife. The system enables rich presence tracking in combining Web Service technology with an XML-based meta model to capture presence data that is linked to semantic descriptions formulated using OWL ontologies.

1 Introduction

Many people can no longer imagine daily life without a mobile telephone. The mobile society is now making way for the next generation technology (3G and beyond). These networks are becoming more intelligent as the service providers are now developing personalized and context-aware services that adapt to the users preferences and his (changing) environment. The MobiLife project [Aftelak *et al.*, 2004] aims to bring these advances in mobile applications and services within the reach of users and groups in their everyday life. One of the projects main focus is the design of a general framework that supports the provisioning of services that are relevant to a user in a given context and adapt their functionality accordingly. Thereby context is regarded as almost any piece of information available at the time of interaction.

2 MobiLife Context Management

The MobiLife Context Management Framework (CMF) [Floréen *et al.*, 2005] provides efficient means of presenting, maintaining, sharing, protecting, reasoning, and querying context information. The key component of the CMF is a generic consumer/producer model given by Context Providers (CPs) and Context Consumers (CCs) as depicted in Figure 1. A CP is a software entity that exposes interfaces to provide context information for consumption by CCs, produced by computing on context elements from encapsulate data sources or other CPs. The internal working of a CP is usually hidden, but may include context aggregation, caching, prediction, reasoning etc. This allows to build smart constellations of context providers that can infer high-level situational information out of tiny bits of lower-level context information from heterogeneous sources. The role of the Context Broker (CB) is to put CPs and CCs in relation by providing registration and discovery facilities for CPs and acting as a single point of entry for CCs.

Technically, the CMF is realized using Web Services as platform-independent application environment. All messages are described in an XML schema, whereas the interfaces are described using the Web Services Description Language (WSDL).

2.1 Context representation

A meta model for context representation is standardized by the MobiLife Context Representation Framework (CRF) to achieve interoperability between CMF components from diverse domains. The metamodel prescribes a standard for a CP advertisement, a context query that could be requested from a content provider, and a context element, the elementary piece of context information. A context advertisement specifies the CPs configuration options, the types of context parameters it can deliver, and for which entities (users, devices, rooms, vehicles, etc.). A context query describes what type of context information is requested from which CP, relating to specific entities. A context element composes metadata such as a time stamp with the actual context value.

We use ontologies to provide a unified machine-interpretable domain vocabulary. This allows for a semantics-based integration of different context sources through linking different parameter types of the metamodel to the MobiLife context ontologies at will.

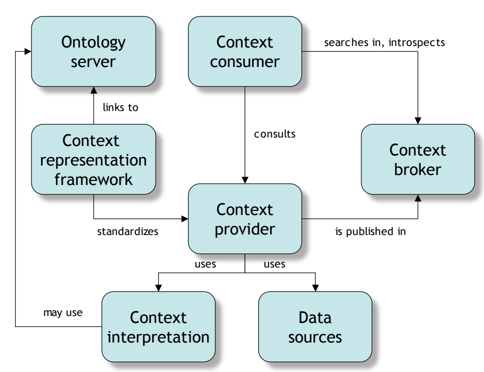

Figure 1: Context Management Framework

*This work has been carried out in the MobiLife project, which is partly funded by the European Union under grant IST-2004-511607.

2.2 Context ontologies

The MobiLife context ontologies define the basic contextual categories and the relations among them. This common high-level structuring of context information enables its integration and consolidation on a semantical basis. Furthermore, the axiomatic descriptions of context elements such as personal situations (i.e., Working, At_home, etc.) can directly be used by logical inference engines to realize high-level context reasoning [Luther *et al.*, 2005]. It is important to note that we do not propose the ontologies as the main representation format for all aspects of context modeling, as ontologies are are limited to the formulation of qualitative aspects and are generally weak in handling large amounts of date efficiently.

The MobiLife context ontologies consist of several modules written in OWL-DL, describing general vocabulary on temporal and spatial concepts, agents as well as devices. They are informed by the vCard standard, the iCalendar representation and the FOAF (Friend-of-a-friend) format (based on RDF) [Brickley and Miller, 2005]. However, we extended the FOAF contact records by vocabulary expressed in OWL-DL for the precise modeling of complex social relations along the lines of [Mika and Gangemi, 2004].

3 Accessing Qualitative Context Sources

The existence of qualitative data is one prerequisite for ontology-based reasoning support. Often an otherwise necessary mapping process from low- to high-level contextual data can be made dispensable, by making use of already existing sources of high-level context information like the data stored in personal information management (PIM) applications. We implemented a prototype that allows for the experimentation with logical context reasoning based on qualitative information from commonly available sources. It has been developed as a Java application and is linked to a back-end reasoning engine, the OS X address book and calendar applications as well as the MobiLife ontologies.

The information stored in an address book often comprises not only a simple contact database but in addition offers the possibility describe relationships among contact entries. Due to the relationship definitions, individuals that are linked with object properties are created and classified to appropriate T-box concepts like "Family" or "Colleague". Similarly, information managed using the calendar manager iCal is accessed and added to the ontology. This time the high level information gained consists of events, which can be associated with people attending this event and the location where it takes place. According to the concepts defined in the schedule ontology, the reasoning system classifies these event instances with respect to their properties – based on the kind of event, its attendees as well as recurrence settings.

4 Presence Tracking

A number of CPs have already been implemented providing information such as location (provided by a GPS device connected via Bluetooth or computed from the cell id communicated by the network provider), local weather information (based on address information), FOAF provider, personal and group agenda (by wrapping a Mozilla calendar server) and

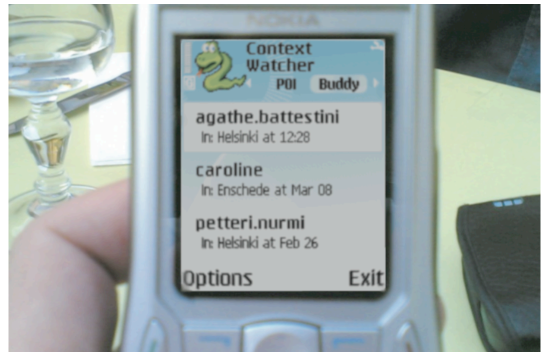

Figure 2: ContextWatcher

body conditions such as heart rate (by accessing Suunto sport sensors such as a T6 watch, a foot pod or a heart rate belt). Some of those CPs provide extended interfaces and are interrelated. For example, the location provider can enrich poor location information (e.g. from cell id or latitude-longitude to street address), stores a history of locations, applies location clustering to determine places frequently visited by the user, and attaches meaning and names to location clusters by linking to T-box concepts describing virtual locations such as "Office", and reports those to the personal agenda creating an automatic activity log.

All current MobiLife CPs can be accessed using the ContextWatcher application, a mobile CC that runs on the Python for Series 60 platform on Nokia phones. The application connects with different CPs (via GPRS) to gather rich presence information (such as speed, location, activity, weather, etc.) and share it with the user'ss buddies. The level of detail is controlled by user-defined policies that make references to his social relations. For example, a user might specify that his family can access the full address of his current location, while access to his location for his friends is restricted to the closest city and for his colleagues to the country. The underlying ontology model is used to interpret concepts such as "Family" adequately (e.g., covering the father, all sisters and other family members).

References

[Aftelak *et al.*, 2004] A. Aftelak, A. Häyrynen, and M. Klemettinen. MobiLife: Applications and services for the user-centric wireless world. In *Proc. of IST Summit*, 2004.

[Brickley and Miller, 2005] D. Brickley and L. Miller. FOAF vocabulary specification. W3C, April 2005.

[Floréen *et al.*, 2005] P. Floréen, M. Przybilski, P. Nurmi, J. Koolwaaij, A. Tarlano, M. Wagner, M. Luther, and et al. Towards a context management framework for MobiLife. In *Proc. of the IST Summit*, Germany, June 2005.

[Luther *et al.*, 2005] M. Luther, B. Mrohs, M. Wagner, S. Steglich, and W. Kellerer. Situational reasoning – a practical OWL use case. In *Proc. of ISADS*, April 2005.

[Mika and Gangemi, 2004] P. Mika and A. Gangemi. Descriptions of social relations. In *Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, Galway, Ireland, September 2004.

Enhanced Presence Tracking for Mobile Applications

Appendix: Demo Explanation

The live software demonstration will show the main building blocks of the presence tracking system.

First, we will provide a detailed insight into the context meta model including the context ontologies with a focus on our OWL FOAF model that supports the inference of additional social relations based on DL reasoning and rule-base FOAF data mining.

Second, we demonstrate with a Java front end to our FOAF Context Provider, how personal information can be made accessible by linking to standard PIM applications (see figure 1 below). Apart from being able to organizing existing high-level personal information into a T-Box structure, this component features ontology-based reasoning to derive entailed knowledge. For a simple example, having defined transitive relationship properties enables automated role completion for individuals. This means, that if two persons have been defined as colleagues of the owner, these two persons are also colleagues themselves. Furthermore, through its simple user interface, one can easily simulate certain scenario use-cases by selecting attending persons, location and time using the available pull-down menus. The simulated situation is displayed in the result window afterwards according to the classification derive using the attached reasoning system. This way, the use of ontologies in terms of reasoning about the users' context can be demonstrated in a graspable manner, revealing the decisive entities that lead to the classification result.

Finally, we demonstrate the ContextWatcher application (Fig. 2 of the demo paper), the enhanced presence tracker (shown in the figure above) running on a mobile phone. It accesses the various local (i.e., GPS, sport, etc.) and remote Context Providers running in the network (location, map, FOAF, agenda, weather, etc.) via GPRS, gathering and distributing (based on the policy settings) rich presence information. In concrete, it has the following functions. It reads cell information directly from the phone, and submits this data to a remote location provider (over GPRS) that in turn tries to enrich this cell information with location information (latitude-longitude, or even address) based on previous GPS measurements and if connected via Bluetooth to a GPS receiver it will update this this cell database with new measurements. It shows when the user is in a well-known or frequently visited location (e.g. home or office) and reports about accidental meetings with buddies. It integrates with a calendar provider that is used to store the deduced activities of a user and his meetings with buddies, as well as to learn about the semantics of a spotted location cluster. It provides a buddy list, to which others users can be added, and after mutual agreement context information can be exchanged with buddies in a direct and unobtrusive manner. It provides some extra services like weather conditions, Points of Interests (POI) or map services based on the available contextual data.

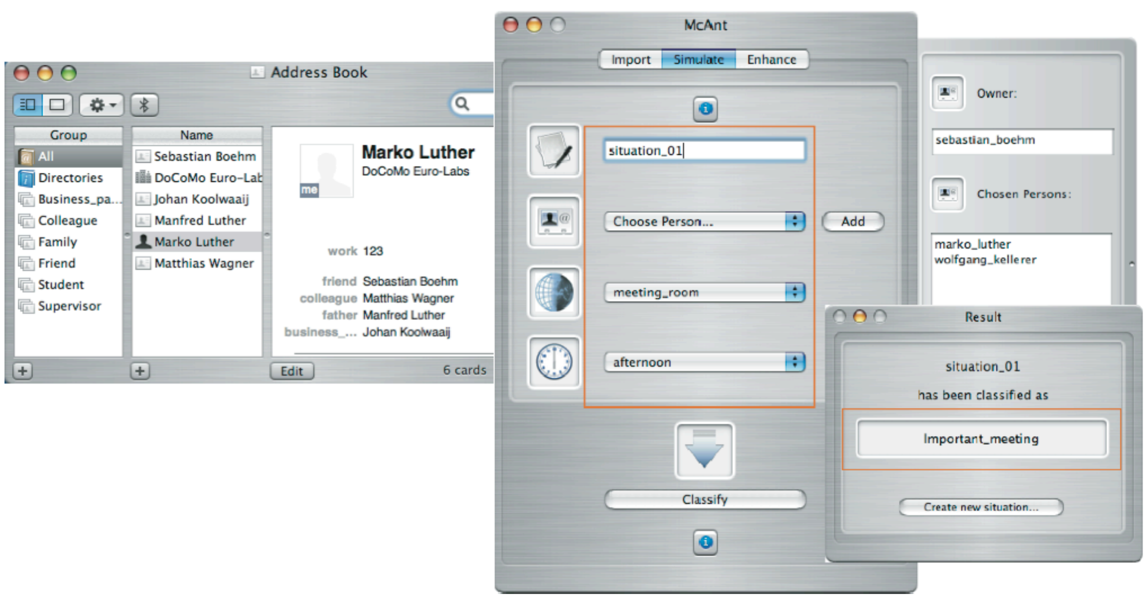

Figure 1: Knowledge integration on the personal desktop.