Enhanced Presence Tracking for Mobile Applications*

Marko Luther, Sebastian Böhm, 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 metamodel 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 their (changing) environments. The MobiLife project [Aftelak *et al.*, 2004] aims at bringing 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 adapts 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 data from encapsulated data sources or other CPs. The internal working of a CP is usually hidden and may include context aggregation, caching, prediction, reasoning etc. Smart constellations of CPs allow the inference of high-level situational information out of tiny bits of lower-level context information from heterogeneous sources. CPs and CCs are put in relation by the registration and discovery facilities of a Context Broker (CB), 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

To achieve interoperability between CMF components from diverse domains a metamodel for context representation is standardized by the MobiLife Context Representation Framework (CRF). This model prescribes a standard for a CP advertisement, a context query that specifies the desired output of a CP and a context element, which represents the elementary piece of context information. A context advertisement specifies the CPs configuration options, the types of context parameters it can deliver, as well as the type of entities (users, devices, rooms, vehicles, etc.), that are supported. A context query describes the type of context information requested from a certain CP and relates to specific entity. 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 semantic-based integration of diverse context sources through linking parameter types of the metamodel to the MobiLife context ontologies at will.

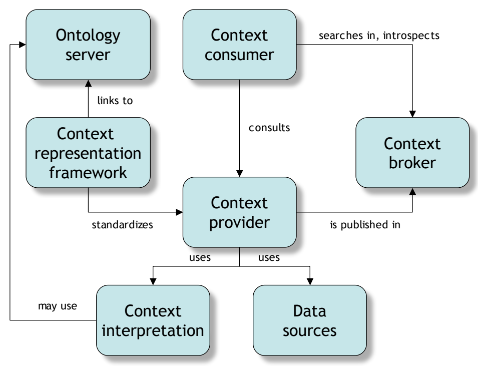

Figure 1: Context Management Framework

*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 semantic. 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 [Brickley and Miller, 2005]. However, we extended the RDF-based 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. 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.

Usually, the information stored in an address book comprises not only a simple contact database, but in addition offers the possibility to 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, the 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 (gathered from 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 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 is able to enrich spatial information (e.g. from cell id or latitude-longitude to street address) and features a history function. Additionally, it applies location clustering to determine places frequently visited by the user. The meaning and name of location clusters are linked to T-box concepts describing virtual locations such as “Office”. These virtual locations are then reported to the personal agenda resulting in an automated activity log.

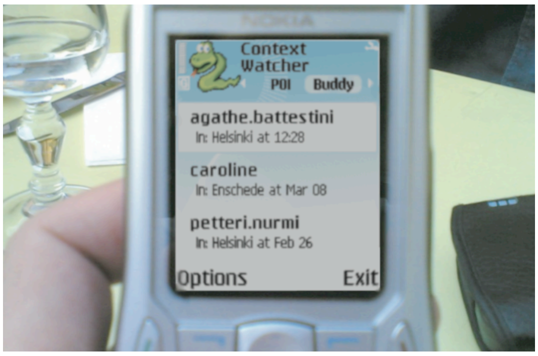

Figure 2: ContextWatcher

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 to different CPs (via GPRS) gathering rich presence information (such as speed, location, activity, weather, etc.) to share it with the user’s buddies. The level of detail is controlled by user-defined policies that make references to user specified social relations. For example, a user might specify that his family can access the full address of his current location, while restricting the access for his colleagues or friends to a different level of detail such as the closest city or the current 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, M. Klemettinen, and S. Steglich. MobiLife: applications and services for the user-centric wireless world. In *Proc. of the IST Summit*, Lyon, 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*, Dresden, 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 *Proc. of the 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, Galway, Ireland, 2004.

Enhanced Presence Tracking for Mobile Applications

Appendix: Demo Explanation

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

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

We demonstrate how personal information can be made accessible by linking to standard PIM applications, using a Java front (Figure 3). Apart from being able to organize existing high-level personal information in a T-Box structure, this component features ontology-based reasoning to derive entailed knowledge. Consequently, one can easily simulate certain scenario use-cases by selecting attending persons, the location and the time using pull-down menus. Afterwards, the simulated situation is displayed in the result window according to the classification obtained from 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 show the ContextWatcher application (Figure 2), our enhanced presence tracker running on a mobile phone. It accesses various local (i.e., GPS, sport, etc.) and remote CPs accessible within the network (location, map, FOAF, agenda, weather, etc.), gathering and distributing rich presence information (based on the policy settings). It reads cell information directly from the phone, and submits this data to a remote CP that in turn tries to enrich it with spatial information (latitude-longitude, or even address) based on previous GPS measurements. If the user is in a well-known or frequently visited location the phone will displays the corresponding Abox element (e.g. home or office). Furthermore, the system 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.

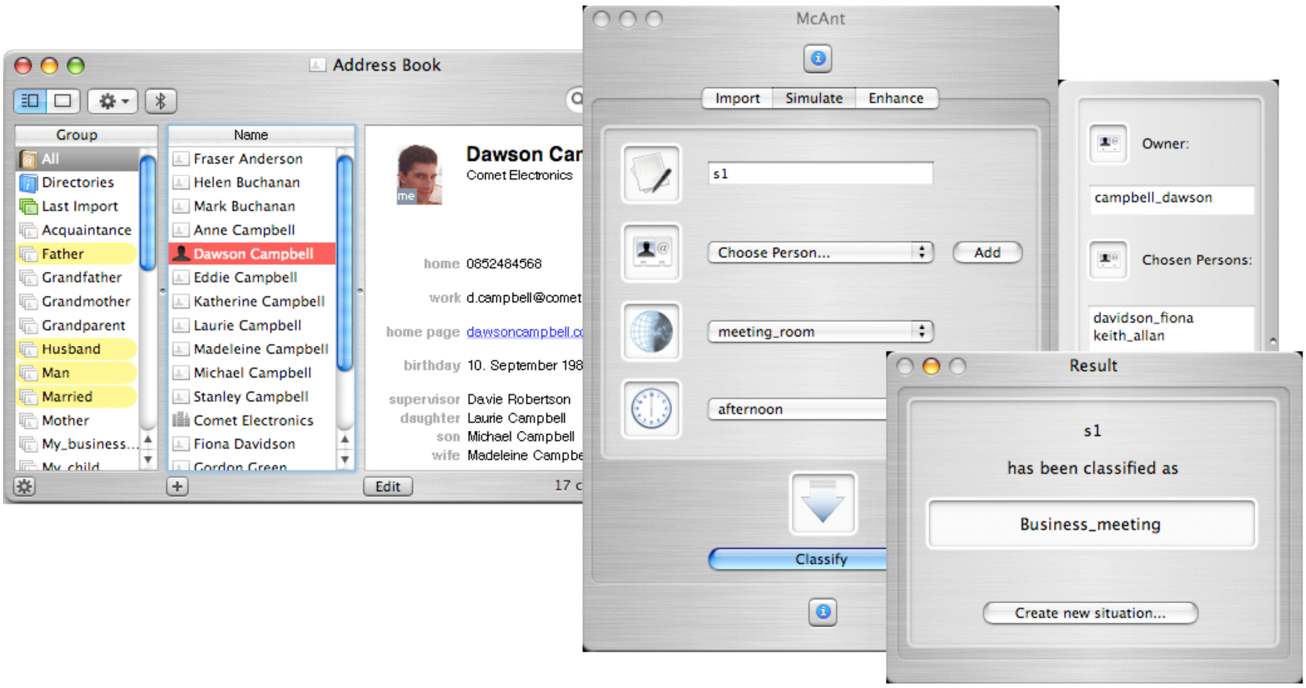

Figure 3: Knowledge integration on the personal desktop.