UBIQUITOUS CONTEXT SENSING IN WIRELESS ENVIRONMENTS

Alois Ferscha, Simon Vogl, Wolfgang Beer
Johannes Kepler University Linz
Department for Praktical Informatics, Altenbergerstrasse 69, 4040 Linz Austria.
ferscha@soft.uni-linz.ac.at, vogl@soft.uni-linz.ac.at, beer@ssw.uni-linz.ac.at

Abstract: The immanent and pervasive use of mobile devices, especially in wireless environments, raises issues about the context awareness and sensitivity of applications. As the use of embedded mobile devices grows in vast quantity, the need for efficient context gathering, representation and delivery evolves. With regard to this situation this work describes problems concerning context sensing, representation and delivery and proposes a new approach for context based computing: Time and event triggered context sensing for mobile devices and an abstract, application and platform independent, representation of context information is introduced. The paper presents different showcases of time and event triggered context sensing in wireless environments.

Keywords: Context Computing, Spatial Proximity Sensing, Event Trigger, Time Trigger, Identification, WLAN, RFID, RDF;

1. INTRODUCTION

Today, the encouraging development of mobile computing and communication devices enables a new vision for future ubiquitous computing environments. A great variety of different electronic devices will be embedded to our environment and to articles of daily use [1]. The intention is, to create intelligent self organizing environments, composed of a multitude of embedded systems. These systems should interact with people in a more natural – and thus more convenient – way than it is the situation today. In the past few years different research efforts have dealt with "smart spaces", environments, exploiting new hardware technologies, like submicron IC design, reliable WLAN communication, low power storage systems, new sensor and actuator technologies and smart materials [11][9]. With wireless communication technologies it is possible for embedded devices to communicate with the user and with each other, as well as to gather information about their local environment. The sensing of information about the local environment is important in that way, as it tells about the existing infrastructure surrounding a certain device. The sensing of information about e.g. the location (or geographical position) of a mobile device could minimize the infrastructure ultimately demanded to provide those services [9]. On the other hand this also means that it is not necessary to build upon a globally accessible network, but to use peer to peer communication. Context computing [6], i.e. the collection, transformation, interpretation, provision and delivery of context information [2][5][6] is the key to future development of smart environments and applications. Recent research work has shown that a distinction among the abstract classes of *person*, *thing* and *place* is useful, when real world objects are mapped to objects in virtual environments [9][10][3]. This distinction is able to fulfill the needs, for abstract real world object base classes sufficiently [7]. In this work we present a generic context information representation framework for the person-thing-place world view, and develop context gathering mechanisms based on time and event triggered context sensors (Section 2). As an abstract context representation mechanism the *Resource Description Framework* (RDF) is adopted in Section 3. We discuss the basic RDF definition as well as an appropriate RDF Schema (RDFS) as a means to provide a vocabulary and structures for expressing the context information gathered from different sensors. Ontologies do have to provide a special object model and a formal semantic, to support adequate modeling and reasoning of object context representation [12].

Particularly with this work we address the issues of automated context sensing and context information transformation, raised when nomadic users roam in a dynamically changing environment. In these cases of time varying object relation in the real world, mechanisms for an automated update of the world model is of critical importance. With the choice of RDF we also address the need to present context information in an application and platform independent way and present effective methods to handle the context information updates, even in cyclic linked context information structures (Section 3). A demonstration scenario of our framework is developed in Section 4, conclusions are drawn in Section 5.

2. CONTEXT SENSING IN SMART ENVIRONMENTS

Embedding processors into daily-use appliances lets users access an ever growing variety of information about the physical world [14]. A persons digital extensions that accompanies it has already become a rich data source. RFID-enabled shopping cards offering access to skiing areas, spas, hotel bars etc. allows tracking, with respect to the presence of the user within the spatial proximity of a sensor (RFID reader [8]) – which is usually fixed in the geography.

Despite this wealth of digital traces, a persons activity in the real world already leaves behind in a digital realm, it is hard to merge these sources into one framework for easy access and processing. In the sequel we develop a context information framework that tries to overcome these barriers by offering a flexible, extensible architecture based on two concepts: (i) the identification of (real world) objects, irrespective of the sensing technology, and (ii) the distinction of two types of context information sources: continuous context information streams and occasional context events.

The foundation for our context sensing network is the ability to digitally identify objects via various identification and addressing methods. Each active artifact (objects playing an active role with respect to our framework) needs its own unique ID, that makes it identifiable and allows further tracking and logging of activities.

Integrating different technologies implies that we have to cope with varying addressing and identification schemes. We deal with this issue by assigning name spaces to the families involved and format these identifiers like URIs, with a leading name-space tag. As an example, consider a mobile device known to the framework as `ip:140.78.95.11`, i.e. a PDA that is linked wirelessly with our campus network and recognized via its name-space tag `ip:`, or one of our rooms identifies as `rfid:0800B9F1AFB1` – here a unique RFID tag number – recognized via the name-space tag `rfid:`. This way, it is easy to integrate new sensors into our framework, covering barcodes (`ean:90018211212`), ISBN-numbers, or any other type of fingerprint.

As far as input data is concerned, we distinguish among two fundamentally different types of information, that need to be treated in different ways: (i) events potentially occurring at certain occasions in the real world and (ii) continuously occurring events describing context data streams. Consequently, two different context sensing mechanisms are needed: (i) watchdog mechanism monitoring the environment for the occurrence of events and their immediate notification to the framework, and (ii) the probing of real world state information continuously over time and the filtering of these streams with respect to data volume and importance to the framework. We refer to the latter as *continuous (context) sensing*, to the former as *event based (context) sensing*.

2.1 Continuous sensing

Continuous data sources provide information about the current state of the real world, e.g.: indicators like temperature, light conditions, link quality, etc. This type of data is typically sampled at fixed time intervals. Alternatively, it may be sampled upon request. In a web-based environment, this data can be provided by a simple HTTP server script. This is an easy way to retrieve the current value. Persistent connections can be used when values should be read at fixed intervals, to minimize connection overhead (HTTP), a method commonly in use for WebCams, status monitors and the like. One of our continuous data sources provides a list of WLAN devices, their MAC and IP numbers, in combination with the name of the WLAN-access point they are associated with. This provides a compact overview of active devices and their location based on access point influence radii.

2.2 Event based sensing

The other data source we use does not deal with system states, but rather with their changes. Events occur whenever some change is detected – e.g.: in the real world by analyzing sensor data, or in a software component executing the world model framework. Due to their dynamic nature, these events cannot be read conveniently over standard HTTP mechanisms at the point of their occurrence. Instead, events that occur have to be actively propagated to the interested parties by calling event handler scripts on their web servers. Like in simulation environments, we can generate events that are derived from continuous data sources by defining specific triggers, like threshold values, or query strings that match different situations we are interested in.

In our test setting we use RFID readers that are connected to mobile devices as event-generating data sources. Whenever a transponder enters the spatial proximity of the electro-magnetic field of the RFID-readers, its unique ID is read and posted to an event listener service, on the server machine executing the context framework which we call *ubic*.

The ubic framework is exclusively based on standard internet technologies like IP-addressing and HTTP. In its current state of implementation ubic relies on fixed server addresses and thus on the availability of an access network, which is, in our case, a campus WLAN. Together with the concept of an (RFID-)tagged environment, ubic can seamlessly combine identification as well as positioning and tracking of the real world.

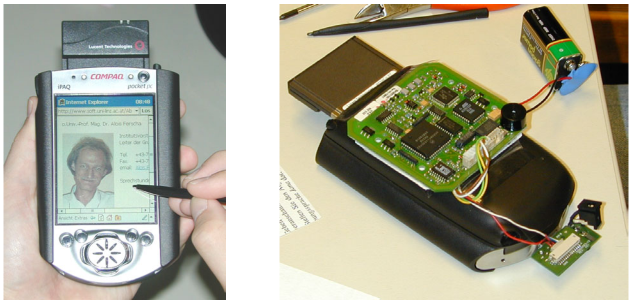

Figure 1. Combining of communication and RFID capabilities

Figure 1 shows how a mobile device reads a tag that is associated with a place (e.g. an office-room) and transmits the tag ID to the ubic framework. This triggers the updater process that checks the relations defined for the tagged item as well as the device that caused the event, updates all concerned data and sends this information to the persistent history as well as a notification process. That informs all interested parties that a room change has occurred.

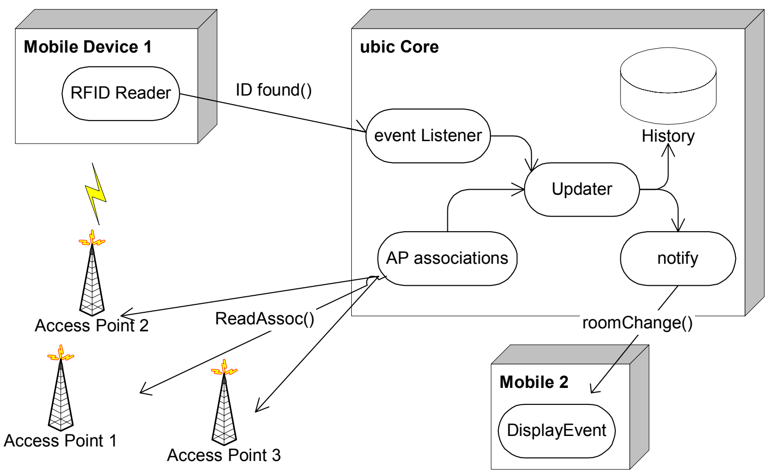

Figure 1. System overview.

3. ABSTRACT REPRESENTATION OF CONTEXT INFORMATION

To proliferate context information in a timely due manner and general format – irrespective of its purpose of use or application – a representation of context information based on the resource description framework (RDF) is proposed, modeling the artifacts *person*, *thing* and *place* as RDF resources. In combination with RDF Schema (RDFS), RDF provides a powerful syntax for semantic knowledge modeling [15]. RDF and RDFS are based on the XML Web data exchange format. The basic RDF model consists of three object types, which form subject, predicate and object triples, called RDF Statements. These three object types are:

1. *Resources*: Every object described in RDF is called a resource, that could be a simple HTML page, data like pictures and graphics on the web or, like in this work, real objects which are not directly embedded into the internet. In this work many objects of the real world, distinguished into person, thing and place present their context information written in RDF.

1. *Properties*: Properties are specific attributes which describe a resource and its relation to different resources. Every property has its own characteristics, like values which are permitted and values that are not permitted. The RDF basic syntax and model specification does not address this specific property characteristics. For this purpose the RDF Schema specification is needed.

1. *Statements*: The subject (resource), predicate (property) and the property value (object) triple, build an RDF-Statement. The property value could be a literal, another resource or an URI to another resource.

RDF has a simple, but powerful, model and syntax definition and is therefore a good choice for representing and delivering context sensing information. Furthermore, it is simple to import or export RDF statements (subject, predicate and object triples) from any kind of database. Another important aspect, concerning the usage of RDF and RDFS as a standard context information representation and delivery format is, that organizations like the *"Dublin Core"* [3], *"On to Knowledge"* [13] and *"Semantic Web"* try to establish standard vocabulary for semantic data description. This improves this work insofar, that we can use standard definitions for properties like *email*, *name*, *date*, or *creator*.

In this section we also try to establish a formal description of resource properties, which are able to describe basic sets of context information properties. These basic sets of context information properties handle the representation of location and containment awareness, ownership and attachments from person to thing and a history of this context information properties. The following list of basic context information properties is used in our context sensing framework, distinguishing between the three basic types of objects:

- *Places*: Places hold information tags about the location (*location*) and references to a set of objects which are actually inside the place (*contains*). In order to track changes in the contains list it is necessary to store any action in a history tag (*contained*). Entries of the contained property list have two time stamps, called *startTime* and *endTime*, specifying the period of time when the entry was part of the *contains* property list.
- *Person*: A person holds information tags about its actual location inside the scenario (*isIn*) and a list of objects which are owned by the person (*ownerOf*). Additionally the person holds a list of locations where it was before (*wasIn*).
- *Thing*: A thing holds the information, if it is able to contain other objects (*canContain*), as for example a bag or a bagpack is able to. In the case that it can contain other objects, it also holds a *contains* information tag and a *contained* tag to track the objects. Furthermore a thing has an information tag about its owner (*owner*).

4. APPLICATION SCENARIO

In the sequel we develop an example for an object of the type place that shows how to integrate the context information, mentioned above, into a RDF description:

4.1.1 Context-aware Suitcase

As one example application, we mention a context-aware luggage [7] shown in *Figure 2*.

Figure 2. Context aware suitcase

An ordinary suitcase has been equipped with an embedded PC, that is linked to the Internet over a IEEE802.11b WLAN connection. Attached to this computer is an RFID-reader (with a matching antenna) that surveys the interior of the suitcase. The application logic is integrated into a web-server running on this machine. Any tagged object that is put into or removed from this 'active' storage results in an event that is passed on from the reader to the listener script on the web server, where the necessary state transformations are executed. This example covers only a subset of our RDF-namespace, as only objects are interesting: The suitcase itself and the objects it may contain. This small subset of our framework nevertheless demonstrates its benefits with respect to the temporal logging of state changes.

4.1.2 Electronic object tagging

Our second scenario covers a more complex setup: a campus WLAN is used for a rough localization of persons roaming the university by tracking the access points their mobile devices are associated with. We superimpose fine-grained position information derived from tagged environments. Offices have been covered with myriads of RFID tags, giving us the possibility to track devices in 10cm ranges [8]. So it is possible to map any sensed tag identification number to a specified location, as it was already described in section 2. *Figure 3* shows how RFID tags can specify locations and map this locations to a virtual environment:

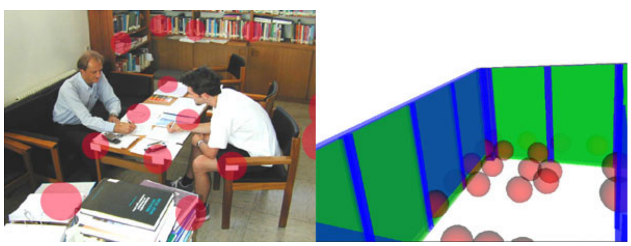

Figure 3. Mapping of RFID tags to specified locations

In this application of the ubic framework context sensing is reached by tracking the access points and reading their association tables. This scenario intends to combine event and time triggered context sensing and to integrate the context sensing information into a wireless location awareness framework. For exact spatial proximity information the RFID technology was used and in order to get location spheres context information the access point association tables were used.

5. CONCLUSION

Future internet use and the development of embedded wireless devices surrounded by smart digital environments, will demand for new methods concerning sensing and presentation of context information. The distinction between event based and time triggered context sensing was mentioned and how these two sensing mechanisms work together in practical applications. Additionally this work presents the possibility to represent context sensing information with the XML based RDF and RDFS semantic data description standard. For future applications the possibility of evaluating the historic context sensing information will be a great advantage and enables methods to personalize current internet services.

6. REFERENCES

[1] D. ARNOLD ET.AL.: Discourse with disposable Computers: How and why you will talk to your tomatoes. *USENIX Workshop on Embedded Systems*, 1999.

[2] P. J. BROWN, J. D. BOVEY, X. CHEN: Context-Aware Applications: From the Laboratory to the Marketplace. *IEEE Personal Communication*s, 1997 Vol. 4, Nr. 5.

[3] D. CASWELL, P. DEBATY: Creating a Web Representation for Places. *Proceedings of the Ninth International World Wide Web Conference*, 2000.

[4] DUBLIN CORE INITIATIVE, http://dublincore.org

[5] A. K. DEY, G. D. ABOWD: Toward a Better Understanding of Context and Context-Awareness. GIT, *GVU Technical Report* GIT-GVU-99-22, June 1999.

[6] A. K. Dey: Understanding and Using Context. Personal and Ubiquitous Computing, *Special Issue on Situated Interaction and Ubiquitous Computing*, 5(1), 2001.

[7] A. FERSCHA: Towards a Context-Aware Embedded Internet. *Proceedings of Informationstagung Mikroelektronik 2001*.

[8] A. FERSCHA, W. BEER, W. NARZT: Location Awareness in Community Wireless LANs. *Proceedings of the Informatik.*

[9] Hewlett Packard: CoolTown Appliance Computing: White Papers. http://cooltown.hp.com/papers.htm

[10] T. KINDBERG, J. BARTON, J. MORGAN, G. BECKER, I. BEDNER, D. CASWELL, P. DEBATY, G. GOPAL, M. FRID, V. KRISHNAN, H. MORRIS, J. SCHETTINO, B. SERRA, M. SPASOJEVIC: People, Places, Things: Web Presence for the Real World. *WWW'2000*, 2000.

[11] MIT Media Lab: Smart Rooms. http://ali.www.media.mit.edu/vismod/demos/smartroom/

[12] A. OMICINI, F. ZAMBONELLI, M. KLUSCH, R. TOLKSDORF (EDS.): Coordination of Internet Agents. Models, Technologies, and Applications. *Springer Verlag*, Berlin 2001.

[13] On To Knowledge Organisation, http://www.ontoknowldge.org

[14] J. PASCOE, N. RYAN, D. MORSE: Issues in developing context-aware computing. In H-W.Gellersen, editor, Handheld and Ubiquitous Computing, *Nr. 1707 in Lecture Notes in Computer Science*, Heidelberg, Germany, September 1999. Springer-Verlag

[15] O. Lassila and R. R. Swick, Resource Description Framework (RDF): Model and Syntax Specification. Recommendation, World Wide Web Consortium, Feb. 1999.