Scalable Computing: Practice and Experience
Volume 6, Number 2, pp. 71–81. http://www.scpe.org
ISSN 1895-1767

CONTEXT SENSING, AGGREGATION, REPRESENTATION AND EXPLOITATION IN WIRELESS NETWORKS

ALOIS FERSCHA, SIMON VOGL AND WOLFGANG BEER*

Abstract. As the use of wirelessly connected embedded mobile devices grows in vast quantity, their situative use and context sensitive behavior becomes an essential feature. For the implementation of context awareness in mobile appliances, the need for the efficient gathering, representation and delivery of context information evolves. This paper describes issues related to context sensing, representation and delivery, and proposes a new approach for context based mobile 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 showcases of time and event triggered context sensing in wireless environments.

Key words. Mobile Computing, Context Awareness, Spatial Proximity Sensing, Event Trigger, Time Trigger, Identification, WLAN, RFID, RDF;

1. Introduction. An 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 [26]. 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 [26]. On the other hand this also means that it is not necessary to build upon a globally accessible network, but to e.g. use peer to peer communication where appropriate. Context computing [15] [19], i.e. the collection, transformation, interpretation, provision and delivery of context information [5][14][15] is the key to future development of smart environments and applications [4] [11]. Context information in a context aware application is captured via a collection of sensors [3], mapped to a representation of the real world ("world model" [19], and used to control the real world via a set of actuators. Recent research work related to "world modelling" 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 [26][29][10]. This distinction is able to fulfill the needs, for abstract real world object base classes sufficiently [17].

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) [31] 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 [34] [13]do have to provide a special object model and a formal semantic, to support adequate modelling and reasoning of object context representations [33].

Wireless sensor networks are emerging as a potentially suitable setting to implement context based applications. We discuss issues of embedded networked environments as sensor actuator systems in Section 2. 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 are 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 4). A demonstration scenario of our framework is developed in Section 5, conclusions are drawn in Section 6.

---

*Johannes Kepler University Linz Department for Practical Informatics, Altenbergerstrasse 69, 4040 Linz Austria. ferscha@soft.uni-linz.ac.at, vogl@soft.uni-linz.ac.at, beer@ssw.uni-linz.ac.at

2. Networks of Senors and Actuators. The vision of networked embedded systems, like e.g. an embedded Internet, in the future pervasive computing landscape is dominated by the ubiquity of a vast manifold of heterogeneous, small, embedded and mobile devices, the autonomy of their programmed behaviour, the dynamicity and context-awareness of services and applications they offer, the ad-hoc interoperability of services and the different modes of user interaction upon those services. This is mostly due to technological progress like the maturing of wireless networking, exciting new information processing possibilities induced by submicron IC designs, low power storage systems, smart material, and motor-, controller-, sensor- and actuator technologies, envisioning a future computing service scenario in which almost every object in our everyday environment will be equipped with embedded processors, wireless communication facilities and embedded software to percept, perform and control a multitude of tasks and functions [16]. Since many of these objects will be able to communicate and interact with the network and each other, the vision of "context-aware" network appliances and network spaces [28] [43] [21]—where dynamically configured systems of mobile entities by exploiting the available infrastructure and processing power of the environment—appears close to reality. We can hypothesize that the individual utility of such embedded network services will be greatly increased if they were personalized, i.e. user centered and dynamically adapted to user preference, location aware, i.e. multimodal and multifunctional with respect to the environment, and time dependent, i.e. if they were time dynamic and exhibited timely responsiveness [14] [15] [42]. One impacting trend hence will be the evolution of context aware environments [23] [35] [22]—often referred to as "smart appliances" or "smart spaces"—that intelligently monitor the objects of a real world (including people), and interact with them in a pro-active, autonomous, souvereign, responsible and user-authorized way. People will be empowered through an environment that is aware of their presence, sensitive, adaptive and responsive to their needs, habits and emotions, as well as ubiquitously accessible via natural interaction [20].

Recent developments in the technology of sensors and actuators, processing devices, embedded systems and wireless communication technologies already accelerate the maturing of embedded network applications [36]. "Context-aware" services [22] [17], i.e. applications able to take into account information about context, such as the location of the user or users [7] [18], the time of the day, the light, sound or humidity conditions, the available communication bandwidth, etc., and, most importantly, automatically adapt to the changing conditions in which they execute, are in the process of being cooperatively developed in the software engineering, embedded systems, communication and hardware integration research communities.

To build context aware applications, the adoption of a world model (or "virtual world") representing a set of objects and their state in a physical (or "real") world (or at least the subworld essential for the specific application) is the common approach suggested in the literature [14]. What makes an application context aware is the ability to interact with objects in the real world, requiring adequate models and representations of the objects of interest, and the possibility to sense, track, manipulate and trigger the real objects from within the world model. Several frameworks for such world models have appeared recently, the most prominent ones identifying persons, things and places as the primary abstract classes for real world objects [29][10]. People living in the real world, acting, perceiving and interacting with objects in their environment are represented in a "virtual world" by "virtual objects" or "proxies". Proxies of persons, things and places are linked to each other in the virtual world, such that this "linkage" is highly correlated with the "linkage" of physical persons, things and places in the real world. A context-aware application now monitors the state and activity of the real world objects via set of sensors, coordinates the proxies according to the rules embodied in the application, and notifies, triggers or modifies the physical world objects via a set of actuators.

The next generation of networked applications will evolve at the very edge of todays global networks (like e.g. the Internet) by wirelessly networked, embedded appliances, characterised by the possibilites of (*i*) ubiquitous access, (*ii*) context awareness, (*iii*) intelligence, (or "smartness") and (*iv*) natural interaction. Ubiquitous access here refers to a situation in which users are surrounded by a multitude of interconnected embedded systems, which are mostly invisible and weaved into the background of the surrounding, like furniture, clothing, rooms, etc., and all of them able to sense the setting and state of physical world objects via a multitude of sensors. Sensors, as the key enablers for implicit input from a "physical world" into a "virtual world", will be operated in a time-driven or event-driven way, and actuators, as the generic means for implicit output from the "virtual" to the "physical world", will respond to the surrounding in either a reactive or proactive fashion. The way how humans interact with the computing environment will thus fundamentally change towards implicit, ubiquitous access [42].

Context awareness refers to the ability of the system to recognise and localise objects as well as people

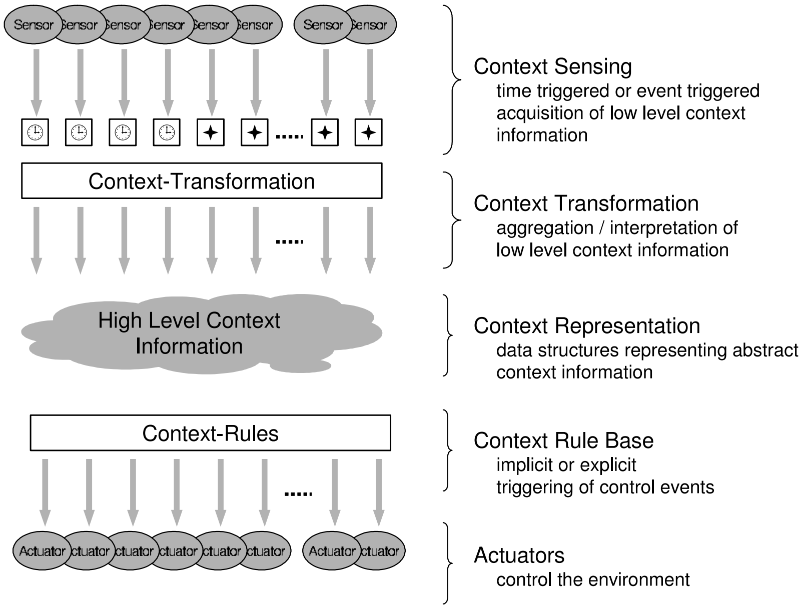

Fig. 2.1. *Context Information Life Cycle*

and their intentions. The context of an application is understood as “any information that can be used to characterize the situation of an entity”, an entity being “a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves” [15]. A key architecture design principle for context-aware applications will be to decouple mechanism for collecting or sensing context information and its interpretation, from the provision and exploitation of this information to build and run context-aware applications [22]. To support building context-aware applications, software developers should not be concerned with how, when and where context information is sensed. Sensing context must happen in an application independent way, and context representation must be generic for all possible applications.

Intelligence refers to the fact that the digital surrounding is able to adapt itself to the people that live (or artefacts that reside) in it, learn from their behaviour, and possibly recognise as well as show emotion. Natural interaction finally refers to advanced modalities like natural speech- and gesture recognition, as well as speech-synthesis which will allow a much more human-like communication with the digital environment than is possible today.

3. Context Sensing in Smart Environments. Embedding processors into daily-use appliances lets users access an ever growing variety of information about the physical world [35], while at the same time a variety of “digital traces” can be collected upon monitoring the users activities. A software framework allowing for an ubiquitous collection of context information via sensors and the deployment of context information via actuators is the challenge we approach in the sequel. We develop a context information framework that addresses dynamic “physical world” changes 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.

3.1. 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.

3.2. 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 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.

4. 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 [31]. 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.

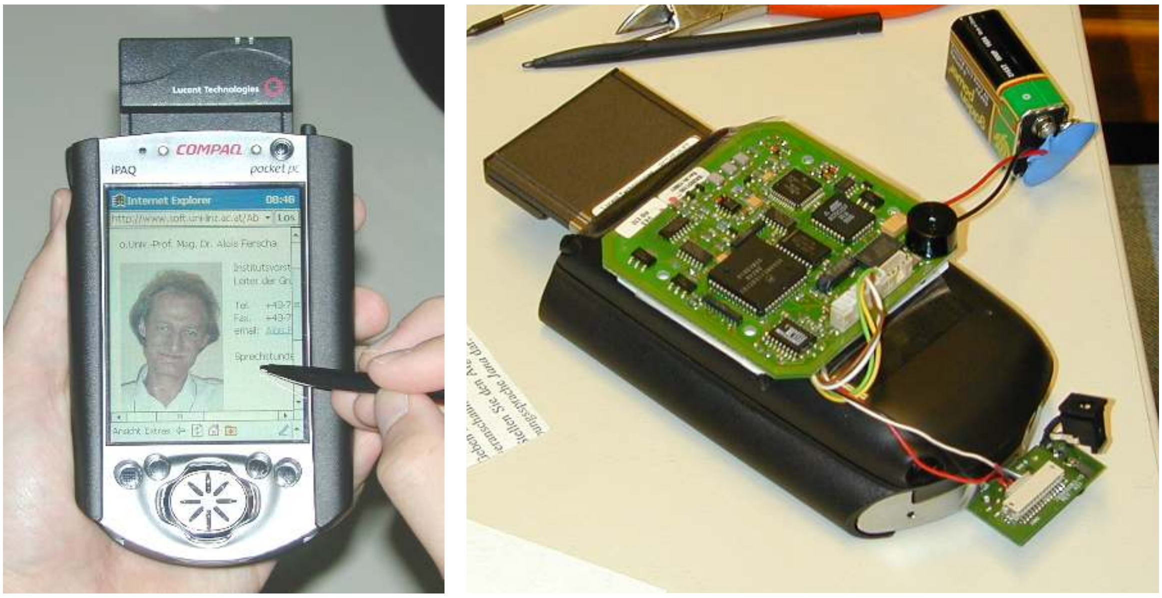

FIG. 3.1. *Combining of WLAN and RFID Capabilities in a Commercial PDA*

2. 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.
3. 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 use of RDF and RDFS as a standard context information representation and delivery format is that a promising research effort (like the "Dublin Core" [10], "On to Knowledge" [34] and "Semantic Web") is under way to establish a standard vocabulary for semantic data description. `UBIC` can thus refer to standard for properties like "email", "name", "date", or "creator", etc. In `UBIC`, resource properties are used to describe context information. Basic sets of context information properties are used for the representation of location, containment, ownership and person to thing relations, together with a history of these context information properties. The following list of basic context information properties is used in our context sensing framework, distinguishing basically three types of objects:

- Place: 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 information on whether 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).

We can summarize that in `UBIC` an application specific abstraction of the real world is generated from three generic classes for persons, things, and places. The reification of physical world objects and their relation among

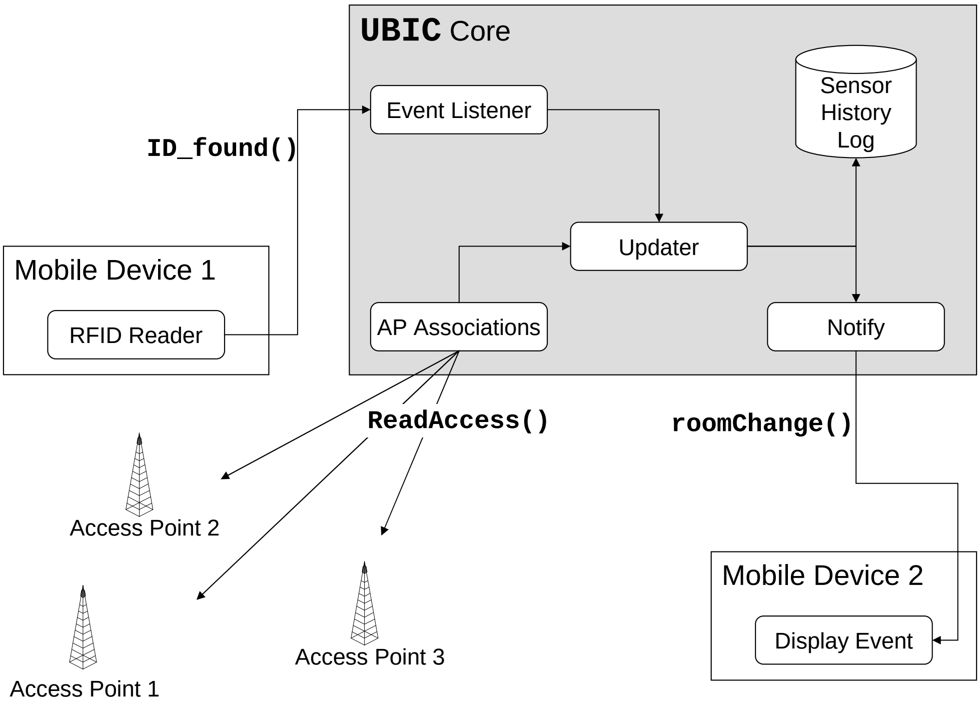

Fig. 3.2. *System Overview*

each other is expressed in RDF. The person object represents the concepts about a person that are necessary to link their physical properties and activities to the virtual world. The thing object encapsulates the basic abstraction for objects relevant to the application and has instances like shirt, bag, etc. The place object is used to represent the essential features of a physical location like an office room or a waiting lounge. Places, things and persons may be related in a manifold of ways.

5. Application Scenarios.

5.1. Context-Aware Suitcase. With our first application scenario we aim at the construction of a context-aware suitcase [17], as shown in Figure 5.2. The primary intent is to make an ordinary suitcase aware of the presence of things it contains, that it has (ever) contained, the location where it resides at the moment (and all the different locations its has ever been to), etc., and to meaningfully use this context information to provide services.

In a first step, we model the context of the suitcase in terms of RDF based abstract representation as described above. The physical appearance of the suitcase is expressed as an instance of the abstract object class `place`, the owner of the suitcase as an instance oft he abstract class `person`, and things to be carried with the suitcase as instances of the abstract class `thing`. The contextual interrelatedness among those object instances is described by a set of (bilateral) object relations expressed in RDF. The whole scenario is expressed in terms of RDF statements, i.e. denoted as subject, predicate and object triples. A subject is represented by a resource, that could be a URL, a simple HTML file, a URI reference etc., a predicate is represented by a property describing a resource or the relation among resources, and an object is represented by a property value. As an example, the statement that the person `ferscha` owns the thing `suitcase` is expressed in RDF as in Figure 5.1, which sketches quite a few relations: The `owner` relation expresses ownership of a real world object by another, the `contains` and `is_in` relations expresses geometrical containment of an object within another one, the `contained` and `was_in` relations trace the history of containment into RDF bags, the `containable` attribute defines whether an object may contain another one, the `controllable` attribute allows to prevent or enable the modification of an object RDF by the object itself, etc. The unique ID associated with every real

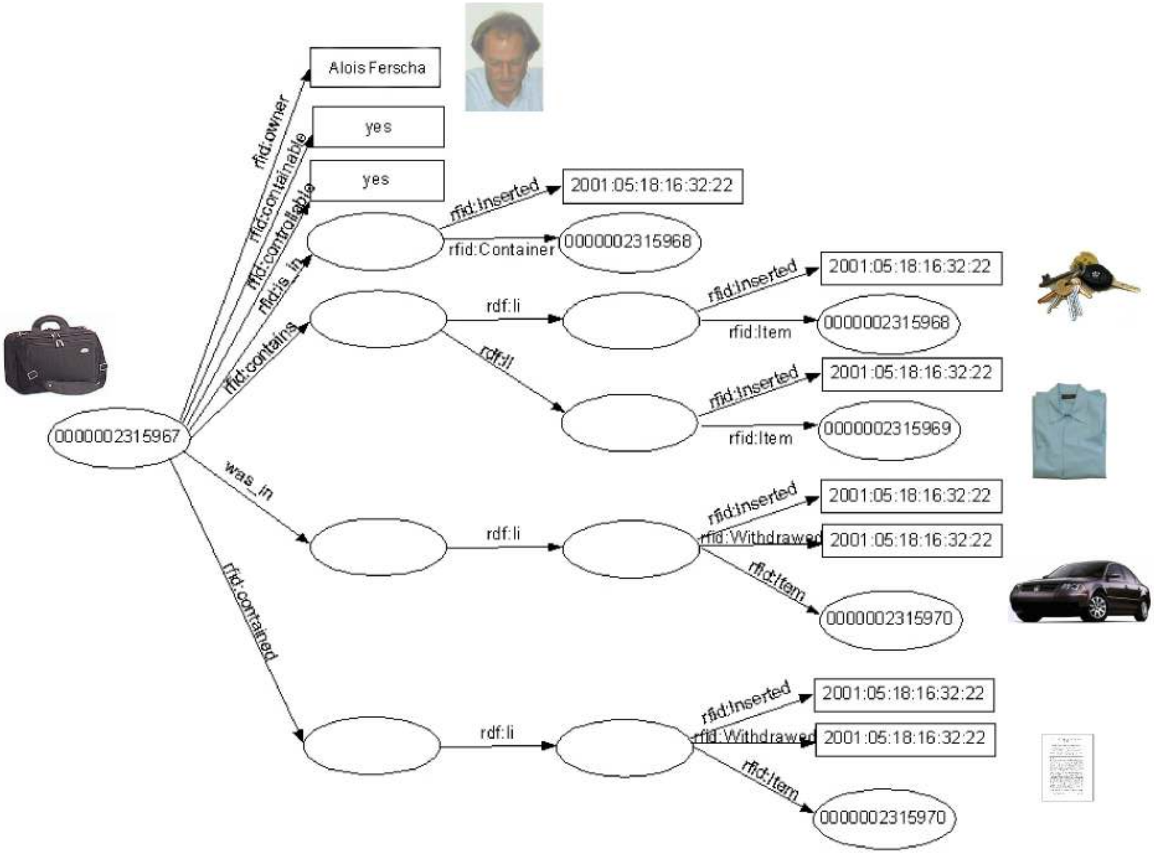

Fig. 5.1. *An RDF Statement Describing an Ownership Relation*

world object is the ID encoded in its RFID tag. It is supposed to be sensed by an appropriate sensor which triggers a script to update the involved object RDFs (Inserting e.g. the shirt into the suitcase would cause an RFID reader to identify the shirt tag and automatically update (among others) both the shirts RDF relation `is_in`, as well as the suitcases RDF relation `contains` by cross-referring URIs.)

Object instances can be created and interrelated at run-time, relations can be updated according to dynamic changes of context (like time, location, etc.) via time-triggered or event-triggered sensors. Upon change of context, like the insertion of a thing into the suitcase, or the movement of the suitcase to a different place, inheritance mechanism resolve transitivity of object relations like the `is_in` relation for objects `contained` in a (geographically) moving (physical world) object.

For building the physical demonstration prototype, an embedded single board computer has been integrated into an off-the-shelf suitcase (see Figure 5.1), executing a standard HTTP services on top of a TCP/IP stack over an integrated IEEE802.11b WLAN adaptor. A miniaturized RFID reader is connected to the serial port of the server machine, an RFID antenna is integrated in the frame of the suitcase so as to enable the server to sense RFID tags contained in the suitcase. A vast of 125KHz magnetic coupled transponders are used to tag real world objects (like shirts, keys, PDAs or even printed paper) to be potentially carried (and sensed) by the suitcase. The suitcase itself is tagged and possibly sensed by readers integrated into home furniture, car or airplane trunks, conveyor belts etc. so as to allow for an identification and localization at any meaningful point in space of the application.

5.2. Context-Aware Office. 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 [18]. So it is possible to map any sensed tag identification number to a specified location. Figure 5.3 shows how RFID tags can specify locations and map this locations to a virtual environment.

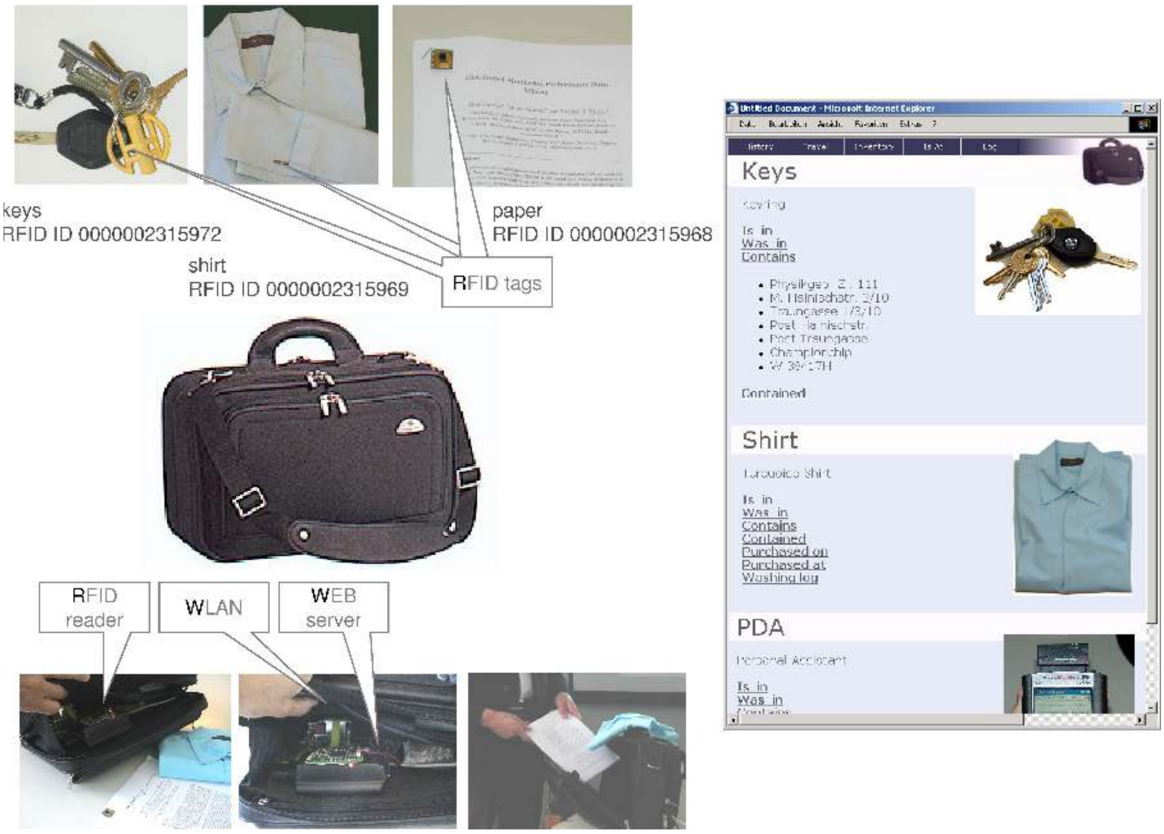

FIG. 5.2. *Context-Aware Suitcase as an Application Instance of the UBIC Framework*

In this application of the `UBIC` framework context sensing is reached by tracking the WLAN 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.

6. Conclusion. Besides the demand for new middleware paradigms and concepts that encompass software issues, devices, users, interaction, and dynamic changes of context, new requirements for the automated configuration and operation of embedded networked applications have emerged. Central to those is the notion of “context”, preliminarily defined as any information that can be used to characterize the state or situation of an entity and the spectrum of its behaviors. Software architectures and frameworks for context-aware applications [19] thus must be concerned with (*i*) abstract representations of context, (*ii*) sensing, collecting, interpreting and transforming of context information and (*iii*) disseminating and making appropriate use of context information to steer the application. The methodological approach used in this paper was to employ standardized Web-metadata modelling (like SGML, XML, RDF) to integrate an arbitrary set of sensors (electrical, chemical, magnetic, optical, acoustic etc.) with an arbitrary set of actuators (like data processors, controllers, motors, filters, etc.) within a single framework. We have presented the distinction among event based and time triggered context sensing, and have shown how these two sensing mechanisms work together in practical applications. Additionally this work has demonstrated how to represent context sensing information with the XML based RDF and RDFS semantic data description standard.

Given today’s global networks as an already existing backbone communication infrastructure, ubiquitous access to this infrastructure still demands technological solutions for the “spontaneous” discovery and configuration of devices and services, the selection of meaningful services offered by autonomous software components, the automatic adaptation to mobile and sporadic availability, the interoperability across manufacturers, platforms and services and the scalability with respect to the number of involved activities and entities. On top of

Fig. 5.3. *Mapping of RFID Tags to Locations in a Wireless Campus Office*

ubiquitous access, context-awareness and issues of intelligence (learning, memorizing, planning, forgetting) and knowledge processing are essential for the provision next generation context-aware applications. To this end—as a first approach for future `UBIC` applications—we are trying to exploit patterns in the context history of context aware, `UBIC` appliances and spaces, so as to be able to serve even anticipated future contexts. As opposed to "re-active" context awareness in the present implementation of `UBIC`, future versions will comprise a context prediction engine (based on statistical inferencing from the context history) and thus implement "pro-active" context awareness.

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. Bahl and V. N. Padmanabhan, *RADAR: An in-building RF-based user location and tracking system*, in Proceedings of IEEE INFOCOM 2000, pp. 775784, Mar. 2000.

[3] V. Bellotti, M. Back, W. Edwards, R. Grinter, A. Henderson, and C. Lopes , *Making Sense of Sensing Systems: Five Questions for Designers and Researchers*, Proc. CHI 2002, CHI Letters, 1 (1), 415-422, Minneapolis, USA, 20-25 April, 2002, ACM.

[4] St. Benford, H. Schnadelbach, B. Koleva, B. Gaver, A. Schmidt, A. Boucher, A. Steed, R. Anastasi Ch. Greenhalgh, T. Rodden, and H. Gellersen, *Sensible, sensable and desirable: a framework for designing physical interfaces*, Technical Report Equator-03-003, Equator, February 2003.

[5] P. J. Brown, J. D. Bovey, X. Chen, *Context-Aware Applications, From the Laboratory to the Marketplace*, IEEE Personal Communications, 1997 Vol. 4, Nr. 5.

[6] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann , *Scalable coordination for wireless sensor networks: Self-configuring localization systems*, in Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA '01), July 2001.

[7] N. Bulusu, J. Heidemann, and D. Estrin, *GPS-less low-cost outdoor localization for very small devices*, IEEE Personal Communication, vol. 7, pp. 2834, October 2000.

[8] M. Brunato and K. K. Csaba, *Transparent location fingerprinting for wireless services*, in Proceedings of Med-Hoc-Net 2002, 2002.

[9] S. Card, J. Mackinlay and G. Robertson, *A Morphological Analysis of the Design Space of Input Devices*, ACM Transactions on Information Systems 9 pp. 99-122 (1991).

[10] D. Caswell, P. Debaty, *Creating a Web Representation for Places*, Proc. of the Ninth International World Wide Web Conference, 2000.

[11] B. Clarkson, K. Mase, and A. Pentland *Recognizing user's context via wearable sensors*, Proc. of the IEEE International Symposium on Wearable Computing (ISWC'00), pp. 69-76, 2000.

[12] A. Dix, T. Rodden, N. Davies, J. Trevor, A. Friday, and K. Palfreyman , *Exploiting Space and Location as a Design Framework for Interactive Mobile Systems*, ACM Transactions on Computer-Human Interaction (TOCHI) 7(3), September 2000, pp. 285-321.

[13] Dublin core initiative, `http://dublincore.org`

[14] 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.

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

[16] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, *Instrumenting the world with wireless sensor networks*, in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, vol. 4, (Salt Lake City, UT), pp. 20332036, May 2001.

[17] A. Ferscha, *Towards a Context-Aware Embedded Internet*, Proceedings of Informationstagung Mikroelektronik 2001, pp.445-456, ÖVE Schriftenreihe Nr. 26, Wien, 2001.

[18] A. Ferscha, W. Beer, W. Narzt, *Location Awareness in Community Wireless LANs*, Informatik 2001, Tagungsband der GI/OCG-Jahrestagung, Vienna, Austria, September 2001, pp.190-195, Österreichische Computergesellschaft, Wien 2001.

[19] A. Ferscha, *Contextware: Bridging Physical and Virtual Worlds*, Reliable Software Technologies - AE2002, Lecture Notes in Computer Science 2361, Springer Verlag, Berlin, pages: 51-64, 2002.

[20] W. Gaver, J. Beaver and S. Benford , *Ambiguity as a resource for design* , in Proceedings of CHI 03, pp. 233-240, Fort Lauderdale, Florida, April 2003.

[21] H.W. Gellersen, M. Beigl and H. Krull, *The Mediacup: Awareness Technology Embedded in an Everyday Object*, Proc. of First International Symposium on Handheld and Ubiquitous Computing (HUC99), Karlsruhe, Germany, September 1999, LNCS 1707, Springer-Verlag, pp. 308-310.

[22] H.W. Gellersen, M. Beigl and A. Schmidt, *Sensor-based Context-Awareness for Situated Computing*, Proc. of Workshop on Software Engineering for Wearable and Pervasive Computing, Limerick, Ireland, June 2000.

[23] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster , *The anatomy of a context-aware application*, in Proceedings of MOBICOM 1999, pp. 5968, Aug. 1999.

[24] M. Hassan-Ali, K. Pahlavan, *A New Statistical Model for Site-Specific Indoor Radio Propagation Prediction Based on Geometric Optics and Geometric Probability*, IEEE Transactoins on Wireless Communications, vol. 1, no. 1, pp. 112-124, Jan. 2002

[25] W. B. Heinzelman, A. Chandrakasan, and H. Balakrishnan , *An Application-Speci.c Protocol Architecture for Wireless Microsensor Networks* , IEEE Transactions on Wireless Communications, vol. 1, pp. 660-670, 2002.

[26] Hewlett Packard, *CoolTown Appliance Computing: White Papers*, `http://cooltown.hp.com/papers.htm`

[27] J. Hightower, G. Borriello, and R. Want, *SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength*, University of Washington, Technical Report: UW-CSE 2000-02-02, Feb. 2000.

[28] L.E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl and H.W. Gellersen, *Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts*, Proc. of UBICOMP 2001, Atlanta, GA, USA, September 2001.

[29] 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*, in Proceedings of the WWW'2000, 2000.

[30] S. Kumar, D. Shepherd, F. Zhao, *Collaborative signal and information processing in microsensor networks*, IEEE Signal Processing Magazine, vol. 19, March 2002.

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

[32] S. Lindsey and C. S. Raghavendra, *PEGASIS: Power-Efficient Gathering in Sensor Information Systems*, Proc. IEEE Aerospace Conference, vol. 3, pp. 1125 -1130, 2002.

[33] A. Omicini, F. Zambonelli, M. Klusch, R. Tolksdorf (Eds.), *Coordination of Internet Agents. Models, Technologies, and Applications*, Springer Verlag, Berlin 2001.

[34] OntoKnowledge, *On To Knowledge Organisation*, `http://www.ontoknowldge.org`

[35] 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

[36] G. Pottie and W. Kaiser, *Wireless integrated network sensors*, Communications of the ACM, vol. 43, pp. 5158, May 2000.

[37] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, *The cricket location-support system*, in MOBICOM 2000, pp. 3243, Aug. 2000.

[38] H. Regenbrecht and M. Wagner, *Interaction in a Collaborative Augmented Reality Environment*, CHIi 2002 Extended Abstracts, 504-505, 2002.

[39] T. Roos, P. Myllymaki, H. Tirri, P. Misikangas, and J. Sievanen, *A probabilistic approach to wlan user location estimation*, International Journal of Wireless Information Networks, vol. 9, July 2002.

[40] C. Savarese, J. Rabaey, and J. Beutel, *Locationing in distributed ad-hoc wireless sensor networks*, in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, vol. 4, (Salt Lake City, UT), pp. 20372040, May 2001.

[41] A. Savvides, C. C. Han, and M. B. Srivastava, *Dynamic fine-grained localization in ad-hoc wireless sensor networks*, in Proceedings of the International Conference on Mobile Computing and Networking (MobiCom) 2001, (Rome, Italy), July 2001.

[42] A. Schmidt, *Implicit Human-Computer Interaction through Context*, Personal Technologies 4(2 and 3), June 2000, pp. 191-199.

[43] A. Schmidt, and K. Van Laerhoven, *How to Build Smart Appliances?*, IEEE Personal Communications 8(4), August 2001.

[44] R. Want, A. Hopper, V. Falcao, and J. Gibbons, *The active badge location system*, ACM Transaction on Information Systems, vol. 10, pp. 91102, Jan. 1992.

[45] Y. Zou, K. Chakrabarty, *Energy-Aware Target Localization in Wireless Sensor Networks*, in Proc. IEEE International Conference on Pervasive Computing and Communications (PerCom 2003), pp. 60-70, March, 2003.

Edited by: Zsolt Nemeth, Dieter Kranzlmuller, Peter Kacsuk, Jens Volkert
Received: April 14, 2003
Accepted: June 5, 2003