Context-aware Communication Services: A Framework for Building Enhanced IP Telephony Services

Manuel Görtz, Ralf Ackermann, Johannes Schmitt, Ralf Steinmetz
Multimedia Communications Lab
Darmstadt University of Technology
Merckstr. 25, 64283 Darmstadt
Tel. +49 6151 166150 Fax. +49 6151 166152

{Manuel.Goertz, Ralf.Ackermann, Johannes.Schmitt, Ralf.Steinmetz}@KOM.tu-darmstadt.de

Abstract— **Communication is an essential part of our daily life. A multitude of devices enable users to communicate everywhere and anytime. One drawback resulting from this ability is the expectation of a caller to always reach the callee. As a consequence the possibility to restrict "availability for communication" becomes a necessary feature too. Users demand efficient filtering mechanisms to control incoming calls according to their current context. Communication services should become more user centric and consider context information to adapt to the most suitable behavior.**

This paper investigates the use of context information to enhance existing SIP call control services and services created with the the Call Processing Language (CPL). These *context-aware communication services* are the proposed approach to cope with the demand for a user centric control of incoming calls. The proposed solutions have been implemented as an extended SIP call control service and an extension to the existing CPL syntax. Currently, location information forms the primary source of context information. Different indoor location sensing systems have been evaluated. Finally, two different kinds of service types have been chosen for evaluation as representatives for the variety of service creation approaches especially in a SIP environment.

I. Introduction

Communication has become a sine qua non in today's life. It is a commodity in our private life and a requirement for nomadic business models. Current telephony functionality is mainly provided by Intelligent Network (IN) systems. These systems operate on a single purpose circuit switched network with a limited number of dedicated access points. The services in the Intelligent Network have evolved from a simple connection for verbal communication to a small, but carefully designed set of supplementary services.

IP Telephony uses packet switched networks, that are shared with other communication applications such as e-mail or instant messaging. IP networks open up a multi-vendor market and provide a place of competition for innovative products, services, and business models. The shared networking platform facilitates the integration of real-time voice communication with other IP based applications.

A rising amount of mobile hand-held devices assist the user's (demanded) mobility. The attainment to communicate (nearly) everywhere also has a downside – the expectation that one is always available. Depending on the user's actual context the user wants to control its availability. In several situations (e.g. in a theater) even the ringing of a device is inappropriate. Therefore, users need the possibility to control the behavior of their devices. This should be accomplished by a helper that implicitly fulfills this task on behalf of the user disappearing from the user's perception.

In this paper the use of context information to parameterize the execution of communication services is introduced. The user's view on the handling of the call is moved into focus. Therefore, services must be highly customizable and the specification of the user's requirements must be intuitive to express. To clearly distinguish between services provided by the telephone network, such as call control and supplementary services, and services that offer tailored functionality to the user, we define the term *Custom User Services*. These services are typically set up and parameterized by the user itself instead of being the outcome of a professional development process. This distinguishes them from traditional telephony services which only allow limited parameterization by the user.

The proposed approach shows the usability of *context-aware communication services* on the basis of two different kinds of services. An end-system service executed in a SIP User Agent (UA) and a third-party call control service based on a CPL script have been chosen as typical representatives. The extensions of both service types have been implemented in our IP telephony system running in a lab environment. The implementation shows their feasibility in supporting the service creation process.

The rest of paper is structured as follows: Section II defines the scope and challenges of our application scenario. Section III provides an introduction of the essential components and concepts needed for the realization of our approach. The approach to consider context information for parameterizing communication services as well as architectural and implementation issues of the system and the prototype are described in Section IV. The paper is concluded with a summary and an outlook in Section V.

II. Investigated Environment and Scope

Communication theory provides insights on the properties of human face-to-face communication. It states that context information is an essential part of communication [1]. A specific context is built during each conversation to resolve ambiguities. Additionally, the communication between humans always happens in a specific situation and a particular environment – a certain context. Currently, no comprehensive technical means to use context information to enhance communication processes exist.

The scenario depicted in Figure 1 provides an intuitive example of a context-aware communication service. The purpose of the service is to support the controlling of incoming calls by *filtering*. The *callee* in this situation attends a meeting. During the meeting the callee does not want to be disturbed by any incoming calls. The only exceptions are calls from the project leader, an important customer and calls related to some sort of (personal) emergency. All other calls should be diverted to a colleague who does not attend the meeting. This behavior should only be active while the callee is in the meeting.

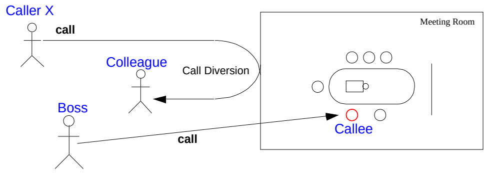


Fig. 1. Context-aware Call Diversion

The user decides to transfer this policy into a service that provides the required functionalities. The call diversion service is the basic service to achieve this goal. Currently this service can only be attributed by the three conditions *unconditional*, *busy*, and *no-reply*. The approach described in this paper extends the condition parameters to satisfy the user's needs to handle the shown scenario. The call diversion services needs to be parametrized by the addresses of the incoming calls and by the context the user is in at the moment of the call. If the conditions are met the call is diverted to the callee. Otherwise the call will be passed through to the specified colleague.

Automated detection of the actual context is essential for offering context-aware services to the user. The switching of the context does not have to be performed by the user itself. Only this way it can actually disappear from the perception of the user This is a major requirement and a distinction to the profiles used in current mobile phones.

It seems that the detection of the location is sufficient to determine the user's context. Location is a primary context information and gives a good indication about the user's situation [2]. However, it often does not provide enough information to resolve possible ambiguities. Context information is a superset of the location information.

The scenario shown in Figure 1 might relate to a farewell party of colleagues. In another case the callee who listened in the meeting has changed its role and became a speaker. In the latter case the handling described is not valid anymore. It is in general inappropriate to accept calls as a speaker. However, during a farewell party taking place in the meeting room calls can be routed to the callee.

A. Related Work

Considering context information for human-to-human communication has not been investigated to its full extend. However, the work on a location based call transfer service at ORL [3] can be seen as an early archetype. The application is limited to just one service and to only one kind of context information – the location. Active Badges have been used to track the user 's location here.

III. Basic Concepts

The concepts of services and context are introduced in this section. Definitions are given to form a common understanding, because both terms exist in different areas and notations.

A. Communication Services

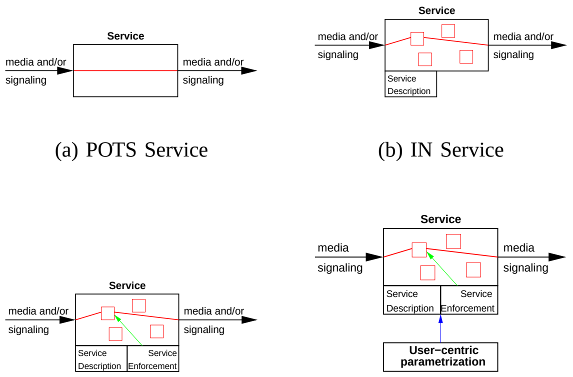
In this paper a generic model for services is introduced. The principle concept is to model services as a black box containing functionality that manipulates incoming and outgoing signaling and media streams. Such a model is also proposed in the Distributed Feature Composition (DFC) architecture [4]. It is used to show the evolution from communication services.

1) Evolution of Services in Telephone Networks: In the traditional telephony system (POTS) the service functionality might be seen as just a piece of wire connecting the incoming and the outgoing signaling and media streams. This kind of service abstraction is depicted in Figure 2(a) and is typically called *basic call*. The interaction with the user is limited to ring tones and the ability to accept the call by going on-hook or off-hook.

The Intelligent Network (IN) provides *call control services* such as call forwarding, call transfer or call completion. The users typically subscribe to these supplementary services. A model for this service is shown in Figure 2(b). The individual services shown an on/off behavior. The user has to explicitly enable or disable them. Some additional information is often provided to the user with the help of small displays that many ISDN phones have.

The current generation of mobile phones provides mechanisms to support the user in controlling incoming calls and personalizing the device. The enforcement of the service is hinted in Figure 2(c). *Profiles* form a container for individual settings can typically be activated for a certain situation. *Caller groups* allow to associate addresses to specific groups. An incoming call from an address within such a group can e.g. be indicated by a defined ring tone.

Even though there is quite a considerable amount of services that are recently evolving they all have one thing in common.



(a) POTS Service

(b) IN Service

(c) Mobile phone Service

(d) User centric Service

Fig. 2. Evolution of Communication Services

The user has to explicitly active and deactivate the services and is rather restricted in parameterizing them. Moreover, the services have only a small amount of control functionalities.

IP Telephony is a promising platform for the integration of services from different application areas. Most notably, application known from the Internet such as web, mail or instant messaging can be named here. The common IP platform eases the integration. Additionally, CPL provides a comprehensive mechanisms to create Custom User Services especially for IP Telephony systems using SIP.

2) Call Processing Language: The Call Processing Language (CPL) primarily addresses service parameterizations by untrusted developers or users [5]. CPL is strictly formalized and uses a decision graph technique. A Directed Acyclic Graph (DAG) is used to describe the control flows. This allows methods for analyzing worst-case paths and a guarantee for a well-defined termination. CPL has no variables, loops, or the ability to execute external programs.

B. Context

Context awareness is an enabling technology to build helpers for user. They are meant to disappear from the user's perception. This allows to create new classes of services.

1) Context Definition: Most people have a general idea about what context is. However, there are diverse (and often vague) notions about what the term actually describes. In the area of Artifical Intelligence (AI) contexts are abstract objects in a domain and statements can be made "about" them [6]. Contexts are often *rich* objects like situations and cannot cannot be completely described. The main question of *what context is* cannot be answered as a result of a unique conclusion. Instead, various notions of context each for its application will be found useful [7]. Throughout this paper the following definition of context adapted from [8] is used:

> *Context* is any information that can be used to characterize the situation of a subject and its interaction with optional objects. Objects are persons, places, or applications considered relevant to the subject.

The combination of several context values provides a very powerful mechanism to determine the current situation. Location, entity activity and time are typical context sources and are forming the *primary context*. Knowledge of the current location and time together with a user's calendar provides a good estimation of the user's current social situation. It is preferable that the user's context is detected automatically and used as an implicit input to the parametrization of services.

2) Context Usage: The utilization of context information requires several processing steps. These are the phases of *acquisition*, *synthesis*, *dissemination* and *use*. The principle procedure is shown in Figure 3.

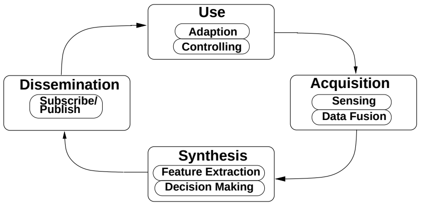


Fig. 3. The Context Cycle

The automatic context acquisition is a prerequisite in order to capture real world situations. This phase is characterized by the usage of a multitude of sensors. Sensors are used to capture the characteristics of the physical world.

> A *sensor* is a device that perceives a physical property. It transmits the result to a measurement. A sensor maps the value of some environmental attribute to a quantitative measurement.

Two types of sensors can be distinguished [2]. *Physical sensors* are electronic components that measure physical parameters in the environment. Information gathered from a host process (e.g. current time, GSM cell, etc.) are considered as *logical sensors*. The sensor S is a time dependent function $S\!:\!t \longrightarrow X$ that provide the system with a set of values (scalars, vectors, or symbolic values) which give a description of the context at that time [9]. A single output of a sensor might not produce sufficient information. *Sensor Fusion* is the combination of sensory data or data derived from sensory data such that the resulting information is in some sense better than would be possible when these sources were used individually.

The context synthesis process assesses significant features of the context. This process uses the sensor information as an input and creates an abstract representation of the captured situation. Finally, the context information has to be disseminated to a receptor which stores or uses the information. The proposed approach describes the whole process from the context acquisition to the use of context information to parametrize communication services.

IV. Approach

The observations on the behavior of human communication and the evolution of communication services lead to the approach to propose context-aware communication services.

This section gives a theoretical model on these services as well as information obtained from implementation and evaluation.

A. Context-aware Communication Services

Context-aware communication services are meant to actually provide the functionalities that satisfy the demands described by the motivation example in Figure 1. The introduced service model is extended by the application of context information. The extended model is shown in Figure 4. The context progress block is subdivided into the functionalities context acquisition and context synthesis.

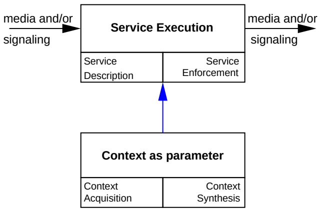


Fig. 4. Model of a Context-aware Communication Service

The context information must be acquired from the environment using sensors as described. In the context synthesis process sensor data and additional information is processed to determine the most likely context. Techniques like sensor fusion or voting are used in this process. The context notion is transmitted to the service execution process. In the service enforcement block the context information is used to influence the execution of the service functionality according to a specified behavior.

Two concrete approaches to provide communication services are proposed in this paper. One approach targets at mobile devices with communication facilities. This can be a PDA with a SIP or H.323 soft client [10] or a next generation UMTS mobile phone using SIP. The second proposal utilizes a 3rd-party call control concept with a proxy server. This proxy server acts on behalf of the user, e.g. if the user is not available. Additionally, the proposed extensions are also useful for extending the Language for End System Services (LESS) an XML-based scripting language [11].

The approach proposed in this paper exemplarily shows solutions for both types of services. The standard call diversion service has been extended with a new condition attribute. The extension give the service the ability to take context information into account. The execution of the call diversion can be based on this condition. The divertion of an incoming call can also be handled by a third-party call control component. A server component which is able to execute CPL scripts was chosen. The CPL language semantic was enhanced by new tags. These tags allow the user to specify in which contexts the CPL action is to be executed. Both extension have been implemented in our IP Telephony environment.

B. Context-aware Communication System

The approach has been used to design a prototype setup. The principle setup is shown for a server and a client using a Context Server. Both examples are chosen because they represent typical application setups. Figure 5 depicts a third-party call control scenario. It includes the Feature Server hosting the users' CPL scripts. On an incoming call the according CPL script is executed and the necessary context information is queried from the Context Server.

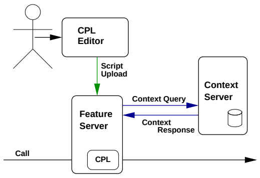


Fig. 5. The Feature Server communicates with the Context Server

In another scenario the Context Server is required by a SIP User Agent. The provided context information is used by the service process itself. The user has to configure the service via the command line with a SIP User Agent. Figure 6 depicts this use case.

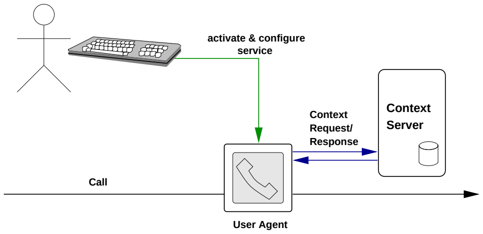


Fig. 6. A User Agent Queries the Context Server for Information

1) Context Server as an Integration Component: The *Context Server* serves as an integration component and is shown in Figure 7. The context information sources on the left side, transmit their data to the server. The mode of the transmissions depends on the capabilities of the sensor. Additionally, operation properties such as the update frequency or bandwidth constrains, require a suitable communication strategy.

Several devices, such as Bluetooth sender, RF/IR-Badges or iCalendar-compliant applications act as potential context information sources. The variety of devices allows to support intrinsic and extrinsic context acquisition. The low-level information sources such as temperature or light sensors are encapsulated by *virtual sensors* (VS). These provide an abstraction to the vendor specific data type and communication. Each source should possess self-describing abilities, which are realized by a Web Service Description Language (WSDL) description. To be interoperable the data will be encoded in an extended Presence Information Data Formal (PIDF) syntax [12].

At present the Simple Object Access Protocol (SOAP) [13] provides the common transport mechanism between the individual components. Currently, the SOAP messages are transported over HTTP, but other transport mechanisms like jabber [14] can be used in the future. Four different transport types, such as request/response or notification are available.

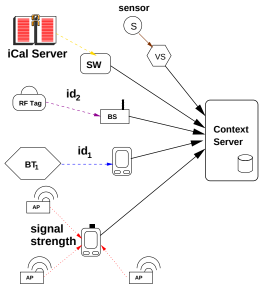


Fig. 7. A Context-aware Communication Service System Using a Multitude of Context Information Sources

Additionally, SOAP support is available for various programming languages and many different operating systems. Alternatively, other interprocess communication mechanism can be utilized as well.

2) Context Acquisition using Indoor Location Sensing and Calendar entries: The described system concentrates on location as a primary context information. Additionally, entries from an iCal-compliant calendar application are used as well. However, the system in principle is not limited to these two types of context information. Location sensing was restricted to indoor usage. Several indoor location sensing systems exist. The Active Badges system [3] is considered as the pioneer work in this area. The badges communicate with their infrastructure via infra-red (IR) transmissions.

Different indoor location sensing systems have been built and evaluated to show the feasibility of such approaches. The chosen approaches reflect the necessity to underline the applicability for different kind of end devices. A location sensing application based on a IEEE 802.11 Wireless LAN infrastructure was implemented following the core ideas of RADAR [15]. A room-scale accuracy can be achieved by comparing the measured values with a before prepared signal strength map. The evaluation [16] resulted in a detection rate of approximately 85% if three or more Access Points (APs) are visible for the measuring device. The rate drops to roughly 60% if only two APs were detected. A PDA running Linux and patched drivers for the WLAN cards were used as the prototype device. A similar approach was undertaken using a Bluetooth environment [17]. The location is determined by comparing the 48 bit ids of the BT access points with a lookup-table containing the corresponding symbolic location information for every identifier. An advantage of both approaches is that they use commodity wireless network technologies existing already in most office-like environments. Device such as PDAs and notebooks are often equipped with WLAN and Bluetooth interfaces. Therefore, no additional hardware cost for location sensing on the client side arise.

Both approaches provide primarily location sensing information on a room scale. A commercially available location sensing system [18] with infra-red tags and badges was installed to allow a more fine grain detection of positions. The tags have been placed near selected positions, such as a presentation wall. This information is useful to decide wether the person is listening to a talk or presenting it.

C. Extending CPL for Context Usage

The proposed goal is the extension of the existing CPL syntax. The extension allows to consider the user's context. New CPL tags had to be developed which uses the context as a condition for switching between different behaviors. We have inserted new CPL tags for high-level contexts such as `<AtHome>`, `<Working>` or `<Travel>`. The CPL execution process queries for the actual context. A Context Server as described above is a possible source for this information. The query can be a SQL statement for the database or a Web Service request messaging using SOAP.

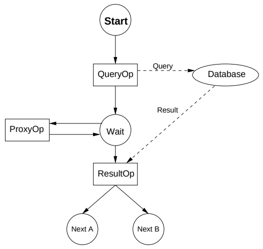


Fig. 8. Sequence of the Context Query in the CPL-Engine

New operators have been added to the existing CPL engine of the Vocal architecture. Figure 8 shows the sequence of operations for a context query. Two operators are of particular interest. The *query-operator* sends the query to the database running on the Context Server. Additionally, there is a *result-operator* which evaluates the result. The result-operator will fork to the according next states. To handle timeouts a wait-state has been inserted between the query-operator and the result-operator. The wait-state is active until the result-operator receives the result from an external data source. During this time the wait-state calls the proxy-operator. This operator can be used to signal messages e.g. 100 Trying to the caller to give a feedback that the process is still running. The result-operator will call the next matching state after receiving the result.

1) CPL Engine: The Vocal IP Telephony system has already some implemented features such as call blocking, call screening or call forwarding. These basic-features can be easily controlled by the web interface. The CPL engine allows extensions to add new functionality. The according tags have to be specified for the existing XML-DTD of CPL. This specifications allow the verification of CPL scripts.

Figure 9 depicts the initialization process of the Feature Server. Existing DTD files are automatically parsed on startup of the Feature Server by the *FeatureBuilder* component. For each user an individual file exist. The builder process creates the necessary state machines for each user. The individual state machines are kept in memory and are linked to a predefined port number.

2) Extension of the CPL Engine: A small set CPL tags is defined in the core specification. These tags allow to

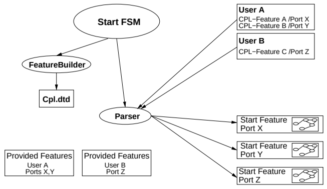


Fig. 9. Initialization Process of the VOCAL Feature Server

express the basic functionality to control calls. To extend the syntax with new and enhanced functionality it is necessary to create new CPL tags. For creating new features it is mainly needed to edit or create the four objects: CPLFeatureBuilder, CallProcessingLanguage, CPLInterpreter and CPLOp*XX*.

The *CPLFeatureBuilder* object has been extended according to the changes in the CPL DTD-file. These objects are instantiated if the new tag is parsed during the build of the state machines. The *CallProcessingLanguage* object is responsible for executing the build of the state machines every time the corresponding CPL tag is reached by the parser. Additionally, in the *CallProcessingLanguage* object variables which are needed to get the parameters from the CPL tag are defined.

To build a state the *CallProcessingLanguage* calls the corresponding process in the *CPLInterpreter* object. The *CPLInterpreter* also defines the type in which the new CPL tag will be inserted to the state machine. A simple tag inserts only an additional state. A branch tag creates an additional forking operators with several next states. Additionally, there are several features for incoming or outgoing calls. If an incoming/outgoing call arrives at the feature server, the server will search for an active feature which belongs to the called/calling user. This process is depicted in Figure 10. If a corresponding state machine is found in the memory the process will interpret the state machine of this feature.

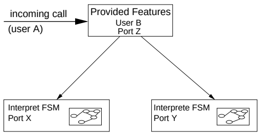


Fig. 10. Execution of FSM withing the Feature Server

The added functionality is implemented in an own operator named *CPLOpXX*. The operator object has to be inherited from the *CPLOperator* class. It will be executed if the state of the corresponding CPL tag is reached within the state machine.

V. Summary, Conclusion & Outlook

A variety of devices allows to communicate everywhere and every time. On the other hand there is a rising demand for an efficient and intelligent way of controlling communication requests. The approach in this paper proposes the consideration of the user's context. A new class of *context-aware communication services* have been introduced. Their applicability and advantage is investigated in the context of the Session Initiation Protocol (SIP). Two different types of services have been enhanced accordingly. The quality of the services depend on the sensed context. Sensing accuracy is found sufficient. The internal processing logic of a standard call diversion service has been modified suitably. Additionally, the syntax of the Call Processing Language (CPL) has been extended by a new class of tags. These tags allow to evaluate context information as conditions. The experiences gathered while implementing the proposed extensions show that the chosen software platform provides a beneficial base. Knowledge from creating one context-aware call control service or a new tag for CPL can be projected onto further extensions. The effort after the initial endeavor stays appropriate for each new feature. A common vocabulary for context information and a unification of the data format for all information sources will yield additional advantages for the development of innovate and user centric context-aware communication services. An extended CPL-editor will guide the user through the creation process.

References

[1] P. Auer, “Kontextualisierung,” *Studium Linguistik*, pp. 22–47, 1986.

[2] A. Schmidt, M. Beigl, and H.-W. Gellersen, “There is more to context than location,” *Computers and Graphics*, vol. 23, no. 6, pp. 893–901, 1999. [Online]. Available: citeseer.nj.nec.com/schmidt98there.html

[3] R. Want, A. Hopper, V. Falcão, and J. Gibbons, “The Active Badge Location System,” *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, January 1992.

[4] M. Jackson and P. Zave, “Distributed feature composition: A virtual architecture for telecommunincation services,” *IEEE Transactions on Software Engineering*, vol. 24, no. 10, pp. 831–847, Oct. 1998.

[5] J. Lennox and H. Schulzrinne, “Call Processing Language Framework and Requirements,” *RFC 2824*, May 2000.

[6] R. Guha, “Contexts: A Formalization and Some Applications,” Ph.D. dissertation, Stanford, 1991.

[7] J. McCarthy and S. Buvač, “Formalizing Context (Expanded Notes),” *Computing Natural Language*, 1997.

[8] A. K. Dey, “Providing architectural support for building context-aware applications,” Ph.D. dissertation, Georgia Institute of Technology, December 2000.

[9] R. R. Brooks and S. Iyengar, *Multi-Sensor Fusion: Fundamentals and Applications*. New Jersey: Prentice Hall, 1998.

[10] R. Ackermann, M. Goertz, M. Karsten, and R. Steinmetz, “Prototyping a PDA based Communication Appliance,” in *Proceedings of Softcom 2001, Split*, Oct. 2001. [Online]. Available: http://www.kom.e-technik.tu-darmstadt.de/publications/abstracts/AGKS01-1.html

[11] X. Wu and H. Schulzrinne, “Programmable End System Services Using SIP,” in *IEEE International Conference on Communications (ICC)*, New York, May 2003.

[12] H. Sugano, S. Fujimoto, G. Klyne, A. Bateman, W. Carr, and J. Peterson, “ Presence Information Data Format (PIDF),” Internet Draft draft-ietf-impp-cpim-pidf-08.txt, May 2003.

[13] M. Hadley, N. Mendelsohn, J.-J. Moreau, and H. F. Nielsen, “Soap version 1.2 part 1: Messaging framework, W3C Recommendation,” June 2003.

[14] “Homepage jabber - get the message,” *http://www.jabber.org/*.

[15] P. Bahl and V. N. Padmanabhan, “RADAR: An in-building RF-based user location and tracking system,” in *IEEE INFOCOM*. Tel-Aviv, Israel: IEEE Computer Society Press, Mar 2000, pp. 775–784. [Online]. Available: citeseer.nj.nec.com/bahl00radar.html

[16] M. Goertz, A. Perez, R. Ackermann, A. Mauthe, and R. Steinmetz, “Location Sensing using RADAR,” Multimedia Communications Lab, Darmstadt University of Technology, Tech. Rep. TR-KOM-2003-09, Sept. 2003.

[17] M. Goertz, R. Ackermann, A. Mauthe, and R. Steinmetz, “A Protype Setup for Location-Aware Personal Communication Services,” in *Evolute Workshop 2003*, Nov. 2003.

[18] “Vista Seek – The IP-based Local Position System,” http://www.ivistar.de.