[GASt04b] *Manuel Goertz, Ralf Ackermann, and Ralf Steinmetz;* **The Digital Call Assistant: Determine Optimal Time Slots for Calls;** Second International Workshop on Multimedia Interactive Protocols and Systems (MIPS 2004), Grenoble, November 2004. S.

The Digital Call Assistant: Determine Optimal Time Slots for Calls

Manuel Görtz, Ralf Ackermann, and Ralf Steinmetz

Multimedia Communications Lab
Department of Electrical Engineering and Information Technology
Darmstadt University of Technology
Merckstr. 25, D-64283 Darmstadt, Germany
{Manuel.Goertz, Ralf.Ackermann, Ralf.Steinmetz}@KOM.tu-darmstadt.de
http://www.kom.tu-darmstadt.de

Abstract. Communication plays a key role in today's businesses. Reaching a communication partner often has become a time consuming task. A multitude of potential communication channels with individual addresses forces a callee to guess an appropriate device at the right time. Under these circumstances additional information about a high probability to reach the calling target at a specific point in time enhances efficiency in communication. The decision when to call and the choice of the communication channel can be based on these information. This paper presents a *Digital Call Assistant* to determine an optimal time slot to place a call. The proposed approach combines calendar events and context information. The combination of these two information sources allows the creation of call plans which provide a list of possible time slots for communication with another user. A trust concept will assure that these sensible data will only be shared among trusted peers. Pending call requests and open call slots are presented to the user. The proposed planning application is going to form a novel part in our context-aware communication service framework.

1 Introduction

The current working requirements in companies have lead to a change in the daily schedule of employees. In many cases working behavior has become nomadic. The working environment is no longer bound to a specific place and time. Nevertheless, employees are typically still integrated in the company by means of communication over networks.

A variety of electronic appliances supports the demands for this kind of mobility. This has lead to the observation that an individual usually carries a multitude of electronic helpers, such as cell phones or PDAs. These companion devices are typically designed for a specific purpose. At each point in time the user has access to a certain set of communicating devices. Additionally, the choice of active communication entities depends on the user's current context. Context is in this case determined by a superset of aspects such as location, situation or environment.

Communication is an essential part of our culture and a requirement for today's business models. Thus, communication must be as efficient as possible for all involved communication partners. However, the caller often faces the problem of how and when to optimally reach the callee. Based on assumptions about the communication partners current context the most suited communication type can be chosen. The callee's context is guessed from typical daily routines and history information.

However, this approach is characterized by a high uncertainty. There is no exact a priori knowledge of the called party's current situation, condition or mood. In many case multiple attempts have to be made to reach the communication partner at an adequate point in time. Finding an appropriate moment will increase the efficiency of the caller and additionally decrease the possibility of disturbing the communication partner.

A typical example is an ongoing meeting which the communication partner B attends. The caller (A), who is not aware of the meeting and its duration, will need to re-try the call attempts. A call on the cell phone might disturb B if the phone is not switched off. Some telephone systems provide a call-completion service. A caller can activate this service. When the callee becomes available again this is signaled to the requester. However, if A is not available then, B might now activate a call-completion. This would lead to a loop of mutual activated call-completion requests and unsuccessful call attempts.

The Digital Call Assistant proposed in this paper tries to provide a solution to this *reachability problem*. Both entities, the caller and the callee, negotiate a point in time which is promising for a call. The negotiation is done between agents that act on behalf of the users. The planning algorithm can be parameterized with a number of requirements. Context information is used to react on dynamic changes in the user's schedules. Thus, the assistant saves time for both communication partners.

The rest of the paper is structured as follows. Section 2 introduces necessary concepts and provides definitions. The proposed planning algorithm is shown in Section 3. The overall system design of the Digital Call Assistant is covered by Section 4. The paper is finally concluded with a summary and an outlook.

2 Concept and Components

The approach proposed in this paper relies on the utilization of context information and calendar events. This section provides the basic concepts and definitions of both information types in order to establish a common understanding.

2.1 Context

Context forms an important concept in the proposed planning algorithm. Individual users are surrounded by their context. This is expressed by the very generic definition that can be found in [1]:

> *Context*: That which surrounds, and gives meaning to, something else.

Actions of users are performed in a specific context. Contexts are often *rich* objects like situations and cannot be completely described [2]. McCarthy states that the main question of *what context is* cannot be answered as a result of a unique conclusion [3]. Several definitions exist in the area of computer science, too. The following definition adapted from [4] has been chosen as notion for context in this paper.

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 that are considered relevant to the subject.

In general applications are considered *context-aware* if they use context to provide relevant information and/or services to the user. The relevancy depends on the user's task [5]. Main usage scenarios of context in such applications have been identified in [6]. These applications automatically *adapt* their behavior according to discovered context (using active context), or *present* the context to the user using reductions of all possible information and/or *store* the context for the user for later retrieval (using passive context).

Context Usage The main prerequisite of a context-aware system is the process of context sensing. This process spans several processing steps. The following steps have been identified: The phases of *acquisition*, *synthesis*, *dissemination* and *use*. The principle procedure is described in the *context cycle* shown in Figure 1. The Context Cycle model follows the principle of the Omnibus Model for multi-sensor fusion [7].

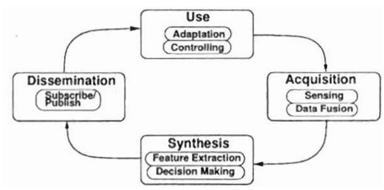


Fig. 1. The Context Cycle

The automatic acquisition of context information is a prerequisite to appropriately model real world situations. A common way is the utilization of a multitude of sensors. Sensors are used to observe the physical world.

Two types of sensors can be distinguished [8]. *Physical sensors* are hardware components that measure parameters in the environment. They provide the information on electronic level, typically as analog output or as digital signals. *Logical sensors* are components that provide information that is not directly sensed from the environment but represents aggregated information about the observed world. Information sources can be a clock as a sensor that offers time and a server offering the current exchange rate. Logical sensors most often supply information as digital signal over a common interface such as a serial data connection or an HTTP-connection.

Each sensor S can be seen as a time dependent function that provides the system with a set of values which give a description of the context at a specific time. The function $S\colon t \longrightarrow X$ returns a scalar, vector, or a symbolic value (X) [9]. The output of a single sensor may often not produce sufficient information for the desired purpose. The concept of *sensor fusion* describes the combination of sensory data or data derived from sensory data. The assumptions is that the resulting information is in some sense better than it would be possible when 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. 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 form the *primary context*.

Knowledge of the current location and time together with a user's calendar gives a good estimation of the user's current social situation. It is preferable that a user's context is detected automatically. Finally, context information has to be disseminated to a context consumer which stores or uses the information. Applications use the context information as an implicit input for e.g. parametrization of functional blocks.

2.2 Calendar Information

Usage of an electronic calendar is a common procedure that structures daily routines and reminds of important dates. Employees usually have access to a company-wide group calendar in addition to a private calendar. Calendar entries are commonly composed of a descriptive text, a start and end time, a categorization and a location (e.g. room number) information. Enterprise business has become considerably dependent of calendaring and scheduling.

Inter-enterprise scheduling can be achieved by central group calendars or by inviting employees to dates. However, the calendaring applications often use proprietary formats. Sharing information across the Internet thus may raise interoperability problems. The Internet Calendaring (iCal) [10] approach tries to provide a common format to exchange information between dissimilar calendaring and scheduling applications. Therefore, it forms an adequate basis for the purposes discussed in this paper.

3 Approach

In this section the proposed approach to tackle the addressed reachability problem is shown. The *planning algorithm* is a first step towards a Digital Call Assistant for communication processes. The solution approach uses user calendar events to determine the best point in time to call someone. Additionally, context information is used to reflect the dynamics of user's activities.

3.1 Components

The basic components of the algorithm are explained next. Each user possess a number of resources, such as communicating devices. The set of communication devices varies with the time and the context of the user. A specific type of such a device (A) is capable of establishing a communication with a certain set of other device types.

$$A \multimap\!\!\circ \{A, C, D\}$$

Each user controls two scheduling lists. The *call-in plan* contains a schedule of what resources for communication are available at what time. The complementary *call-out plan* comprise the time slots for possible call request with the according communication resources. The format of the list follows the iCalendar specification. The core definition has been extended by additional necessary information.

3.2 Planning Algorithm

The principle sequence of the planning procedure is shown in Figure 2. The scenario comprises Context Servers and Calendar applications that are co-located with the planning applications. The sequence can be divided into the subsequences *initialization*, *call*, *planning*, *updating* and *running*.

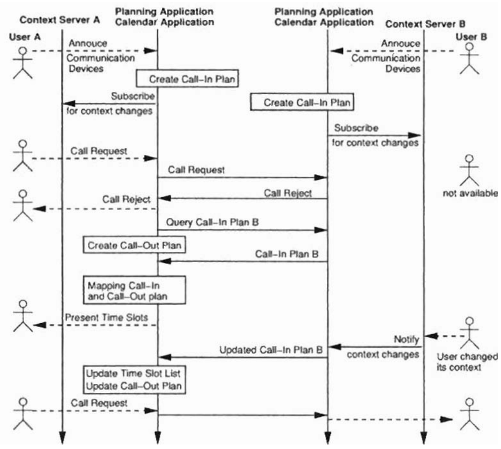


Fig. 2. Sequence of the Planning Algorithm

Initialization phase The users subscribe to the planning application. During the subscription the users announce their communication resources to the application. The communication resources can be categorized regarding their capabilities. These facilities are taken from a compatibility list that describes what device is able to communicate with which other device.

The planning application queries an iCalendar compliant calendar application about the current calendar entries of the subscribed users. The number of entries can be limited by obtaining an arbitrary time horizon, e.g. a time range of 12 hours. Next, the calendar events are analyzed and a static call-in plan is generated. This plan serves as a coarse structure.

The call-in plan entries are derived from the predefined preference list associating calendar categories such as *meeting*, *travel* or *lunch* with possible available communication entities. Each calendar event is augmented with the available communication channels. Additionally, the free-times of the user are also associated with the available communication devices. The entries are supplementary associated with a context. This association allows the dynamic adaption of the time range of the call plan entries.

The call-in and call-out plans are stored as a directed acyclic graph (DAG). Each entity E controls a set of resources R. Each of the resources has a list of time slots where communication with this resource is possible. These time slots are denoted v_n. A time slot must at least contain its start time t_s and end time t_e, denoted as $v_n = (t_{s_n}, t_{e_n})$.

In order to support the dynamics of the user's situation and the resulting changes in their daily schedule context sensing techniques are used. The planning application binds the calendar events with the according context whenever it is possible. This allows the application to adapt to the current user's context. To be aware of context changes the planning application subscribes at the Context Server for the required contexts.

Call phase After the initialization phase the user specifies its call intension. Therefore, the user provides the planning application the parameters such as the callee's address, importance of the conversation and the type of communication. A number of communication types have been defined for the proposed solution. These types are used to classify the communication events. Additionally, the attributes can be used to express the intention of the communication. An event can be augmented by more than one attribute. These information allow the called party to prepare scheduled calls. The following attributes are distinguished.

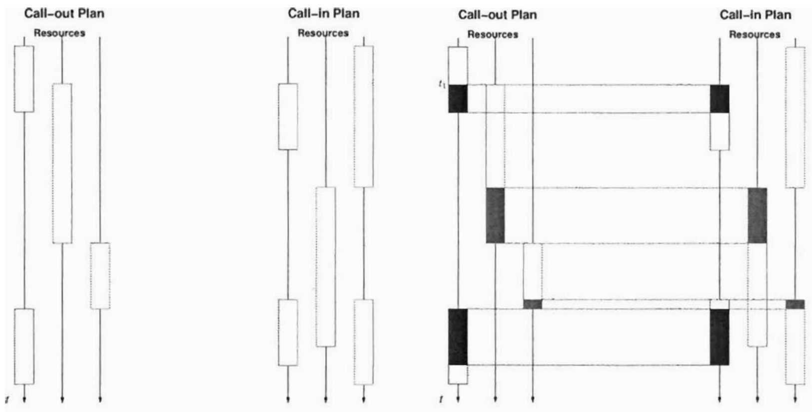
asynchronous Describes communication via e-mail or facsimile.
synchronous Attribute for real-time interactive communication such as phone calls.
uni-directional A one-way exchange of information that do no require any direct feedback, such as calls to answer machines.
bi-directional The transmitted messages is answered by the callee.
informative The content is purely informative. This kind of message is often transmitted in a uni-directional or asynchronous fashion.
urgent The communication request has to be processed with the highest priority.
private Classifies the conversation as private.
public The content of the message is declared as public.

The planning application interacts with a communication instance that initiates the call on behalf of the user. Such a 3rd party call control entity could e.g. be a back-to-back user agent (B2BUA) in a SIP environment. In case that the communication request was not successful, e.g. the called party was busy or not available, the planning application will start its algorithm. The algorithm will try to find a suitable time slot for the next call attempts.

Planning phase The planning phase is the core of the algorithm. In the first step a static call-out plan of the calling user is created. This step could also be done in the initialization phase. The planning application then queries on request the calling target for a current call-in plan. The corresponding planning assistant responds with the call-in plan. The exchange is similar to the sharing of context information shown in [11].

A suitable trust and privacy concept is needed to secure this very sensible information. The concept of trust and context has been investigated in [12] Additionally, the shared information should be available in a hierarchical structure. Such that persons belonging to a specific trust group will only receive the according information detail level.

Figure 3(a) shows the call-out plan of the caller on the left side. On the right side of Figure 3(a) the call-in plan of the callee is depicted. Both partners possess three communication devices shown as resources in the figure. After receiving the other party's plan the *mapping algorithm* is executed. The algorithm determines possible time slots for efficient communication. A temporal ordered list of these time slot is returned. These time slots are shown as overlapping areas in Figure 3(b). The beginning of the next possible time slot is marked as t_1. The example assumes exclusively disjunct resources.



(a) Call-In Plan and Call-Out Plan

(b) Possible Time Slots after Matching Algorithm

Fig. 3. Matching of Resources in Call-In Plan and Call-Out Plan

Updating phase During pending call attempts the schedule of the call-in and call-out plan can change. When the user's context changes this information is signaled to the Context Server. If the context of the user and the context information bound to the calendar events are not in consistence the planning application is notified. A mismatch might occur if a scheduled meeting lasts longer than previously expected or the user is back earlier from lunch.

Upon receiving such a change the call-in plan and call-out plan of the user are updated. Thereafter, the update is signaled to all subscribed planning applications. The utilization of a context sensing infrastructure is an important concept since users are usually not diligent in updating their calendar entries manually. The context-awareness of the application allows for the automatic adaption of the scheduled plans.

Running phase The caller has a list of pending call requests. If the planning algorithm has marked a call requests as *promising* the user can initiate this call request. If the call is still unsuccessful the algorithm will be run again. The phases *call* and *updating* are usually executed multiple times during the use of the application.

3.3 Matching Algorithm

Two algorithms are shown in more detail. The *matching algorithm* (Alg.1) compares two nodes v_n, v'_n of the graphs representing the call-in and call-out plans. If the two nodes have an overlapping region, the algorithm will return the maximum overlapping region. Otherwise an empty set is returned.

Algorithm 1 The Matching Algorithm

function MATCH(v_1, v_2)
    $s \leftarrow \min(t_{s_1}, t_{s_2})$
    $e \leftarrow \max(t_{e_1}, t_{e_2})$
    if $e > s$ **then**
        return (s, e)
    else
        return null
    end if
end function

Algorithm 2 describes the building of a time slot list, that contains all identified overlapping regions in temporal order. The resource graphs are compared node by node. A drawback of the algorithm is the exponential runtime of $O(n^2)$. It occurs since all entries of the call-in plan have to be matched with all entries in the call-out plan.

However, an exhaustive search is not necessarily required. Each first match of $v \in R$ with $v' \in R'$ is saved as a pivot element. The next node of R does not need to examine all nodes prior to the pivot element. Additionally, no element has to be checked after a missed matched since the nodes in R are ordered regarding their time. This strategy significantly decreases the number of operations.

4 The Digital Call Assistant System Design

The planning algorithm introduced in Section 3 will be part of our context-aware communication infrastructure. The communication platform is based on an Open Source IP Telephony system [13] using the Session Initiation Protocol (SIP) [14]. However, the proposed approach is in no way limited to just SIP as communication protocol. The overall system design is depicted in Figure 4. It comprises the components Context Server, Calendar Server, Planning Application and a Communication Server.

Algorithm 2 Building Time-Slot List

Require: R and R' as ordered list regarding start time

function MAPPING(R, R')
    $i \leftarrow 0$
    for all $v \in R$ **do**
        for all $v' \in R'$ **do**
            if $t_s \geq t'_e$ **then**
                next
            else if $t_e \leq t'_s$ **then**
                next
            else
                $T[i] \leftarrow$ MATCH(v, v')
                $i \leftarrow i + 1$
            end if
        end for
    end for
    return T
end function

4.1 The Digital Call Assistant System Setup

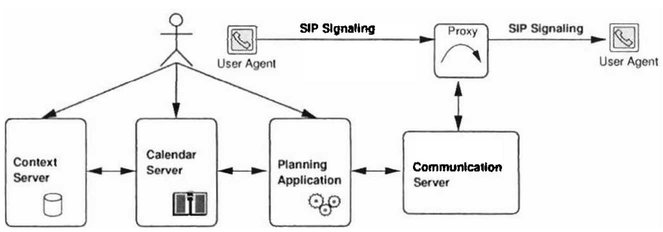


Fig. 4. The Digital Call Assistant System

4.2 Context Provision Infrastructure

The prototype system uses a *Context Server* as an integration component and context source for other context-aware applications. The planning application is a consumer of the context. The planning algorithm requires the current context to parameterize or modify the schedules of the participants. The context server is shown in Figure 5. A multitude of context information sources transmit their data to the server. The data will be encoded in an extended syntax of the Presence Information Data Format (PIDF) [15].

Low-level information sources such as temperature or light sensors are encapsulated by *virtual sensors* (VS). The purpose of the these virtual sensors is the provision of an abstraction to the vendor specific data type and communication. Typical context information sources are devices, such as Bluetooth sender, RF/IR-Badges or iCalendar-compliant applications.

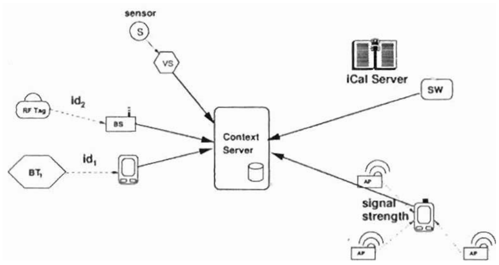


Fig. 5. A Context-aware System Using a Multitude of Context Information Sources

The context server provides the context information via a Web Service interface. The planning application can address the requests for context data using the Simple Object Access Protocol (SOAP). The client application can use a request/response mechanism to query the necessary information. Additionally, a subscribe/notify mechanism can be used to be informed when a context change has taken place.

Location Information In order to test the proposed call planning approach context information needs to be available. Therefore, an indoor location sensing system has been implemented. Different indoor location sensing systems have been built, installed and evaluated to show the feasibility of our approach. The investigated concepts in our lab cover location sensing using a Wireless LAN infrastructure, the utilization of Bluetooth and a location system with infra-red beacons and radio frequency tags.

A location sensing application based on a IEEE 802.11 Wireless LAN infrastructure has been implemented. This approach follows the core ideas of RADAR [16]. The intended granularity of the system was the detection of the room a user is currently in. The detection was achieved by comparing the measured signal strength values with an a-priori prepared signal strength map of the floor. The evaluation [17] 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 available. 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 [18]. A PDA with a Bluetooth interface receives the beacons send out by the base stations. The 48 bit address of each base station provides a tag for identification. The actual location is obtained from a lookup-table that holds a corresponding symbolic location information for every identifier. An advantage of both approaches is that they use commodity wireless network technologies. WLAN and Bluetooth already exist in most office-like environments. Device such as PDAs and notebooks are often equipped with Wireless LAN and

Bluetooth interfaces. Therefore, no additional hardware cost for location sensing on the client side is needed.

4.3 Extending and Using the iCalendar Format

The core specification [10] of the iCalendar defines a set of data fields for shared and distributed electronic calendaring. A MIME content type was chosen as calendar format. The following calendar components are defined: VEVENT, VTODO, VJOURNAL, VFREEBUSY and VTIMEZONE. For the purposes of the presented Digital Call Assistant the existing fields are used whenever possible. Where needed the content is interpreted according to our needs. The MIME content is suitable to be transmitted with a variety of protocols such as SMTP or HTTP.

The transport protocol of the events is derived from the iCalendar Transport-Independent Interoperability Protocol (iTIP) [19]. The protocol provides basic methods to negotiate free time slots. However, the negotiation does not provide the necessary functionality needed for the proposed planning algorithm and adaption through context information.

The Calendar User Type (`CUTYPE`) provides a mechanism to specify the type of calendar user. The parameter can e.g. contain the types `ROOM` and `RESOURCE`. These parameters are used to convey information about the location of the user and the available communication resources. To mark a time slot as free or busy the `FBTYPE` parameter is used. It allows to set the time slot to `FREE` or various `BUSY` types. Additionally, individual notations can be defined. For interoperability these new notions have to be registered with the IANA. The following listing shows an exemplary data set for the Digital Call Assistant.

Listing 1.1. iCalendar data used for the Digital Call Assistant

```
BEGIN:VEVENT
DTSTAMP:20040613T220546Z
ORGANIZER:MAILTO:mgoertz@kom.tu-darmstadt.de
CREATED:20040613T215649Z
UID:DigitalCallAssistant
LAST-MODIFIED:20040613T220437Z
SUMMARY:Meeting
LOCATION:S3106/348
CLASS:PUBLIC
PRIORITY:3
CATEGORIES:Conference
CUTYPE:RESOURCE: sip-phone, mobile-phone
FBTYPE=BUSY:20040616T109000Z/20040616T103000Z
DTSTART:20040616T090000Z
DTEND:20040616T100000Z
TRANSP:OPAQUE
END:VEVENT
```

5 Conclusion

Efficient communication promises savings in time and money. The loop of unsuccessful calls and unsuccessful call-backs has to be avoided. Knowledge about time slots when the communication partner can be optimally reached is the key information for this purpose. The decision when to place the call can be based on this information. In this paper a Digital Call Assistant has been introduced. The assistant's planning algorithm determines time slots which are appropriate for communication. Call-in and call-out plans are the basis for the algorithm. These plans are derived from user's electronic calendar (iCalendar). Context information is used to continuously adapt the static schedule to the user's current situation. The call assistant is part of our context-aware communication service infrastructure. The presented algorithm is identifying the time slots straight forward. Enhancements (e.g. trust concept) to this system are left for further work.

References

1. Imperial College Department of Computing: Foldoc: On-line dictionary. (http://foldoc.doc.ic.ac.uk/foldoc/))
2. Guha, R.: Contexts: A Formalization and Some Applications. PhD thesis, Stanford (1991)
3. McCarthy, J., Buvač, S.: Formalizing Context (Expanded Notes). Computing Natural Language (1997)
4. Dey, A.K.: Providing Architectural Support for Building Context-Aware Applications. PhD thesis, Georgia Institute of Technology (2000)
5. Dey, A.K.: Context-aware computing: The cyberdesk project. In: AAAI 1998 Spring Symposium on Intelligent Environments, Palo Alto, CA, AAAI Press (1998) 51–54 http://www.cc.gatech.edu/fce/cyberdesk/pubs/AAAI98/AAAI98.html.
6. Chen, G., Kotz, D.: A survey of context-aware mobile computing research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College (2000)
7. Bedworth, M., O'Brien, J.: The omnibus model: A new architecture for data fusion? In: Proceedings of the 2nd International Conference on Information Fusion (FUSION'99), Helsinki, Finnland (1999)
8. Schmidt, A., Beigl, M., Gellersen, H.W.: There is more to context than location. Computers and Graphics **23** (1999) 893–901
9. Brooks, R.R., Iyengar, S.: Multi-Sensor Fusion: Fundamentals and Applications. Prentice Hall, New Jersey (1998)
10. Dawson, F., Stenerson, D.: Internet Calendaring and Scheduling Core Object Specification (iCalendar). RFC 2445 (1998)
11. Goertz, M., Ackermann, R., Steinmetz, R.: Enhanced SIP Communication Services by Context Sharing. In: 30th EUROMICRO Conference 2004. (2004)
12. Martinovic, I., Goertz, M., Ackermann, R., Mauthe, A., Steinmetz, R.: Trust and context: Two complementary concepts for creating spontaneous collaborative networks and intelligent applications. In: Proceedings of SoftCOM'04, International Conference on Software, Telecommunications and Computer Networks, Croatia/Italy (2004)
13. Vovida Networks, Inc.: Vovida vocal system. (http://www.vovida.org/)
14. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: Session Initiation Protocol. RFC 3261 (2002)
15. Sugano, H., Fujimoto, S., Klyne, G., Bateman, A., Carr, W., Peterson, J.: Presence Information Data Format (PIDF). Internet Draft draft-ietf-impp-cpim-pidf-08.txt (2003)
16. Bahl, P., Padmanabhan, V.N.: RADAR: An in-building RF-based user location and tracking system. In: IEEE INFOCOM, Tel-Aviv, Israel, IEEE Computer Society Press (2000) 775–784
17. Goertz, M., Perez, A., Ackermann, R., Mauthe, A., Steinmetz, R.: Location Sensing using RADAR. Technical Report TR-KOM-2003-09, Multimedia Communications Lab, Darmstadt University of Technology (2003)
18. Goertz, M., Ackermann, R., Mauthe, A., Steinmetz, R.: A Protype Setup for Location-Aware Personal Communication Services. In: Evolute Workshop 2003. (2003)
19. Silverberg, S., Mansour, S., Dawson, F., Hopson, R.: iCalendar Transport-Independent Interoperability Protocol (iTIP) Scheduling Events, BusyTime, To-dos and Journal Entries. RFC 2446 (1998)