Enhanced SIP Communication Services by Context Sharing

Manuel Görtz, Ralf Ackermann, Ralf Steinmetz
Multimedia Communications Lab (KOM)
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

Abstract

Communication plays a central in our society. It affects our private lives as well as business activities. Humans usually observe the environment and the communication partner with all their senses. The perceived information is evaluated to deduce the context of the targeted communication partner. However, distant interpersonal communication does not provide a priori knowledge of the called party's current situation, condition or mood – the callee's context. Current communication systems do not offer satisfying technical means to support context sharing between communication partners. The proposed solution in this paper is based on an enhancement of the Session Initiation Protocol (SIP) for IP Telephony systems. Different mechanism of sharing context among communication peers have been investigated, implemented and evaluated.

1 Introduction

Communication is an essential part of our culture. The behavior and the codex applied when interacting with others forms a part of our social knowledge. A series of implicit assumptions of the mutual situations and the potential communication partner is applied before approaching one another. The conclusions from the observation allow to form a reasonable assumption about the other's context. This context is considered for the further progress of the verbal conversation.

However, remote interpersonal communication differs significantly from face-to-face communication. The caller is deprived of the ability to sense the context of the communication partner. There is no exact a priori knowledge of the called party's current situation, condition or mood. The call initiator can only make assumption about the probable context. These assumptions are based on former experiences and expectations.

It is basically obvious that direct human face-to-face communication has advantages over a remote conversation using a technical medium. Context information and social conventions protect (to some extent) the privacy of a person. The initiator of the conversation is responsible to obey certain criteria and to decide whether to start or to wait.

The current generation of mobile phones, phones for the Intelligent Network (IN) and IP phones provide the identification of the caller to the called party. Additionally, the `Subject` header field in Session Initiation Protocol (SIP) can be used to specify the reason of a call. However, this feature is not widely used today. This information allows the callee to decide whether to accept the call based on its own context. Caller groups allow the user to define a set of addresses. A call from a caller within such a group is handled according to a predefined action. However, the problem that the calling party has no knowledge about the context of the called party still remains.

This paper investigates several approaches to provide technical means to share context information between communication partners. The purpose is to allow an adequate way to consider context information of the remote party. IP Telephony has been used as application scenario. IP Telephony is a particular interesting service in this context. Comprehensive integration abilities of user's computation and communication services is one of its promising properties. A SIP-based Open Source telephony system forms the development platform for the proposed extensions.

The rest of paper is structured as follows: Section 2 gives a detailed motivation based on an application scenario. Section 3 provides a short introduction of the essential components and concepts of the approach. The approach to share context information between communication peers and implementation issues of the system are described in Section 4. The paper is concluded with a summary in Section 5.

2 Motivation

Interpersonal communication can be distinguished between direct face-to-face and distant communication. Communication theory states that context information is an essential part of communication. This refers to the content of the verbal communication as well as to the environmental situation of the communication partners. During each conversation a specific context is built to resolve ambiguities [2]. The outer environmental context likewise influences the type and topic of the conversation.

Additionally, the use of context plays an important role at the start of a conversation. A communication partner observes the situation of other communication partners to perceive their current context. Multiple criteria are taken into account in this decision process. Social roles, content of the conversation, and current situation are examples for these criteria. Humans deduce a "good" starting point for the conversation based on the observed information.

Currently, no comprehensive technical means exist to comprehensively map this behavior onto interpersonal communication over distance. The proposed approach assumes that the use of context information will enhance communication processes. Especially, the aspect of *sharing* context information is considered in this paper. The context information is exchanged during the signaling phase of a communication setup.

Figure 1 shows an example that is used to illustrate the use of context sharing. Consider the following situation: A caller initiates a call to a callee who is currently in a meeting. Common questions when calling persons on their mobile phone are "Where are you?" and "Do I disturb you?". However, if the callee has been disturbed by the call it is already too late.

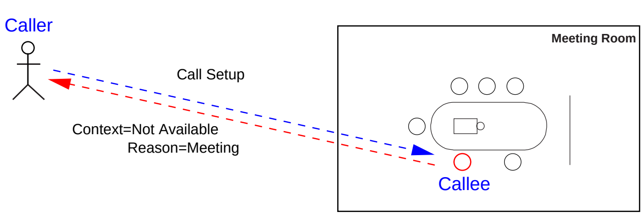

Figure 1. Context Information is shared during Call Setup

Knowing that the communication partner is in a context in which a call is potentially disturbing would allow the caller to better decide whether or not to initiate a call. In the case that the reason for the call is not important or even obsolete due to the callee's context the caller can select from a set of several options. The call might be shifted to a later point in time, a message might be sent via an asynchronous text-based channel such as e-mail, facsimile or SMS. Alternatively, a context-aware call completion service might be invoked, the call might be discarded or initiated to a different person.

2.1 Approach of Context Sharing in SIP-based Telephony Systems

The purpose of the proposed solution in this paper is to provide a technical equivalent to the human ability to observe the situation and perceive the current context of the communication partner. An IP Telephony system using the Session Initiation Protocol (SIP) is the aimed target platform. Context information is queried from the called party and displayed to the caller. Additionally, call control services are extended such that the context information can be used to offer context-aware communication services. However, only a fraction of the human capabilities of context acquisition and use is realized by the described approach. The main general contribution of this paper is the evaluation of several specific transport mechanisms for the use within a SIP environment. These mechanisms form an extension to the existing state of the art and have been investigated, implemented and evaluated.

2.2 Related Work

An initial approach for the sharing of context information has been presented in [18]. However, the described approach is limited to mobile phones using the Wireless Application Protocol (WAP). An additional drawback is the restriction to only a few possible pre-defined answers. These answers are presented to the user. A user can decide whether to proceed or to discard the call. However, the context information is not used to automatize a call processing. The caller has still to become active in triggering the next events.

3 Basic Concepts

In this section the basic concepts of context and the Session Initiation Protocol, that is used for signaling, are covered. It is meant to provide a common understanding of the used terminologies.

3.1 Context

The use of context information gives a number of advantages in communication as motivated before. Context-awareness is an enabling technology to build helpers that are disappearing from the users perception. This allows to create new classes of services.

In this document the focus is on *communication services* in the area of SIP-based IP Telephony. These services are either call control services or custom user services providing customized functionality to the user. In principle the following definition is used:

> A *service* σ is defined as a meaningful set of capabilities provided by an existing or intended set of systems to all who utilize it [20].

3.1.1 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. [11] investigates the formalization of context in the area of Artifical Intelligence (AI). Contexts are abstract objects in a domain and statements can be made "about" them. They are often *rich* objects like situations and cannot be completely described. The main question of *what context is* cannot be answered in single and final statement. Instead, various notions of context each for its specific application will be found useful [12]. Throughout this paper the following definition of context adapted from [7] is used. This definition is widely accepted in the area of context-aware computing.

> *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.

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 lets an application have a good estimation of the user's social situation at a specific point in time.

3.1.2 Context Usage

The utilization of context information requires several processing steps. These are differentiated in the phases *acquisition*, *synthesis*, *dissemination* and *use*. A similar principle can be found in the area of multi sensor fusion [4].

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 physical world.

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

Two types of sensors can be distinguished [17]. *Physical sensors* are electronic hardware components that measure physical parameters in the environment. Information gathered from a host process (e.g. current time, GSM cell, etc.) is considered as originating from *logical sensors*. The sensor S is 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 $S\colon t \rightarrow X$ returns a scalar, vector, or a symbolic value (X) [5]. The 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 source were used individually [8].

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 transmitted to the applications. The application uses the information internally or displays it to the user. The described mechanisms must fit into an IP Telephony setup based on the Session Initiation Protocol (SIP).

3.2 Session Initiation Protocol

This paper proposes a solution that works with a SIP [16] based IP Telephony system. Without loss of generality the principles can be transfered to other signaling protocols such as H.323. SIP is an application layer signaling protocol. Its main purpose is setting up, modifying, and tearing down multimedia sessions. The text-based signaling messages provide call routing information and media descriptions. They are transfered in transactions that requests that trigger responses.

SIP follows the "fast in core, smart at the edges" principle. This means that a substantial part of the distributed system intelligence is located in the end-systems. A SIP telephony setup incorporates several SIP entities. A *User Agent* (UA) is the interface to the user, initiates calls and keeps track of the call state. The User Agent (UA) is the equivalent to the traditional telephone device.

SIP servers are intermediate systems, that handle the application-level control of the signaling routing, and are equivalent to SCPs (Service Control Points) and SSPs (Service Switching Points) in the IN. The Proxy server and the Registrar are the common SIP server in IP Telephony setups. Both server are often co-located. SIP servers form an entity that is especially suited for system enhancements, because all signaling messages pass through them. We extend them in our approach.

3.2.1 Event Notification

The SIP event framework [14] is an extension to the SIP core specification. Applications subscribe to event sources using the SUBSCRIBE method. When a specific event occurs a NOTIFY message is sent to the consumer application. The event concept is part of the SIP Instant Messaging [6] and Presence [15] architecture. It is well suited to also transport context information. The utilization of the SUBSCRIBE/NOTIFY mechanism is part of our approach.

4 Building Blocks for the Approach

This section describes our approach for a generic mechanism that enables context sharing. Several mechanisms to acquire context information are described in this section. Different methods for transmitting context information between SIP entities have been investigated. Finally, the prototype setup which combines the presented mechanisms in a SIP environment is explained.

4.1 Context Sharing

Two methods to exchange context information can be distinguished. The context sharing can be performed directly between the individual clients. Alternatively, the exchange can be done between a client and a server that relays or utilizes the information. Both methods are investigated in this paper.

Figure 2 shows the principle procedure of a direct context sharing between two SIP User Agents (UA). A *query/response* mechanism is used to exchange the context information. The UAs must have the capability to acquire and transmit their own context. This method is especially interesting for supporting voice conversation setup in a peer-to-peer mode and for the autonomous use with mobile hand-held devices.

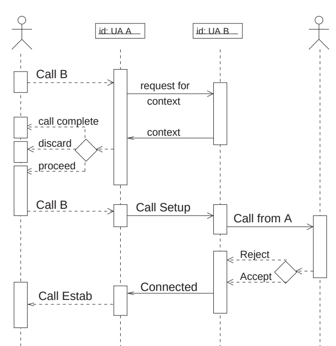

Figure 2. Context Sharing Using a Request/Response Mechanism

An alternative communication mechanism is the utilization of *subscriptions* and *notifications*. Figure 3 depicts such a mechanism with a User Agent and a *Context Server*. A Context Server acts as a proxy entity that possess the context information of User Agents. Whenever the context server becomes aware of context changes it announces them.

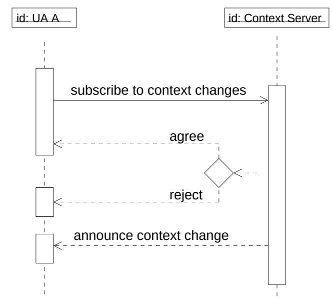

Figure 3. Context Sharing in a Client/Server Fashion

A User Agent is not the only entity that is suitable to response to a context request. The SIP proxy is an applicative component, especially if the Context Server functionality is co-located with a SIP signaling proxy. The proxy intercepts the incoming messages which query the callee's context. Responses are sent out by the proxy on behalf of the UA. The Context Server/Proxy serves as an integration component and is depicted in Figure 4. The context information sources, shown on the left side, transmit their data to the server. The mode of the transmissions depends on the capabilities of the sensor. Operation properties such as the update frequency or bandwidth constrains, require a suitable communication strategy.

Several devices, such as Bluetooth senders, 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 for vendor specific data types and communication mechanisms. Each source should possess self-describing capabilities, which are notated with e.g. Web Service Description Language (WSDL) descriptions. Data is encoded in an extended Presence Information Data Format (PIDF) syntax [19] to ensure compatibility with existing presence information systems.

4.2 Context Acquisition

Automated context acquisition is a prerequisite for context-aware applications. To show the feasibility of the

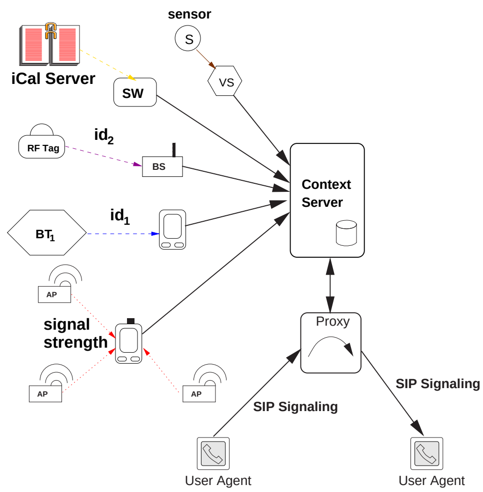

Figure 4. A Context-aware Communication Service System Using a Multitude of Context Information Sources

proposed approach, location information as well as calendar entries from iCal-compliant applications have been used as context sources for the system. However, the system is not limited to these two types of context information.

Several indoor location sensing systems exist. The Active Badges system [22] is considered as the pioneer work in this area. The badges communicate with their infrastructure via infra-red (IR) transmissions. Thus, a line-of-sight is needed to work properly. Within our work different indoor location sensing systems have been set up and evaluated to show the feasibility of such approaches. The investigated concepts cover location sensing using a Wireless LAN infrastructure, the utilization of Bluetooth and a location system with infra-red beacons and radio frequency tags.

RADAR [3] is a radio frequency based approach that utilizes a IEEE 802.11 WLAN infrastructure. The location is estimated by evaluating the signal strengths measured between the client and multiple visible access points. Our experiments have shown that a room-scale accuracy can be achieved by comparing the measured values with a before prepare signal strength map of the floor. The evaluation [10] 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 [9]. A PDA with a Bluetooth interface receives the beacon signaling sent 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. Devices 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.

Both approaches provide primarily location sensing information on a room scale. A commercially available location sensing system [1] with infra-red tags and badges have been installed in order 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 needed for instance to detect if a person is a listening visitor of a meeting or is active as a speaker.

4.3 Infrastructure Elements

The utilization of central infrastructure components such as a SIP Proxy/Context Servers provides several beneficial properties. A *trapezoid* routing is typically found in SIP telephony system. This means that outgoing and incoming SIP proxies are traversed by the signaling. The SIP proxies are often combined with the SIP Registrar entity. The routing via a proxy supports user mobility and multiple registered devices belonging to one user. Therefore, it naturally fits in the signaling process to locate the Context Server in the signaling path.

Additionally, if the user has switched off the device, the server can still answer incoming requests about the user's current context. A central instance is useful if several context sources in the environment transmit their data to such a server and not to all individual end-devices.

Even though the properties for the use of a central entity seem advantageous over the peer-to-peer concept both are considered useful. They are complementary concepts necessary to form a complete system. Such a system must support all kinds of setups and user or administrator needs.

4.4 Prototype setup

The proposed extensions have been implemented as part of our context-aware communication services architecture. The Open Source software project VOCAL [21] forms the platform for SIP-based IP Telephony. The Vovida Open Communication Application Library (VOCAL) provides a fully-fledged SIP system with a variety of SIP server, administrative entities and a command line User Agent (UA). The UA is of special interest because the suggested extension can be relatively easy integrated into this device.

The internal architecture of the VOCAL UA is based on Finite State Machine (FSM). The behavior is described by a set of *states* and *operators*. The operators subscribe to specific states of the internal structure. Events, such as incoming messages or key strokes trigger the operators. The operator that first matches to the trigger is executed and the next state is entered. During an operation several actions, such as sending messages or storing status information, are performed.

A cut-out of the complete FSM of the UA is shown in Figure 5. It shows the call setup procedure of UA on the caller side as a state machine diagram. The extensions are inserted in this process. Before the call setup is performed the context is queried. After the user entered the address (SIP URI) of the callee the operator *OpQueryContext* is called. The actual process depends on the method chosen for the context query. On receipt of the response the state *Context* is entered. In this state the UA waits for the user input on how to process further. If the user wants to proceed with the call setup the next state of the regular setup procedure is entered. Otherwise the call initialization is canceled.

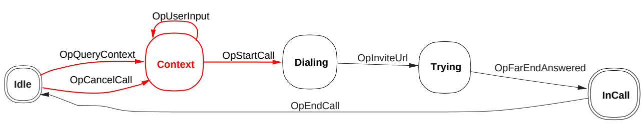

Figure 5. Extended Call Setup FSM

4.4.1 Context Query using OPTIONS Methods

The analysis of the existing methods defined within the core specification of the Session Initiation Protocol (SIP) [16] lead to the use the OPTIONS method for querying context information. It is a request that queries another SIP entity about its capabilities. A SIP server responds to an OPTIONS request that it supports all methods without REGISTER and it allows SDP descriptions. The principle message exchange is shown in Figure 6. The initial OPTIONS request is answered by a 200 OK response. The response contains the available capabilities of the answering entity. This method is usually send before the actual call initiation phase. Therefor it is well suited from its intention to query also other information from the communication partner.

The original OPTIONS method was extended by an additional header field. It is used to indicate that the requestors seeks to find out the current context of the receiver. The policy to ignore unknown header fields let the message traverse unmodified SIP entities. A UA with the extensions will respond with its capabilities as described in the SIP standard. A modified UA with the proposed extension however will parse the header field in the OPTIONS method header. Upon recognizing the `Context` header field the client will include the current user's context into the reply.

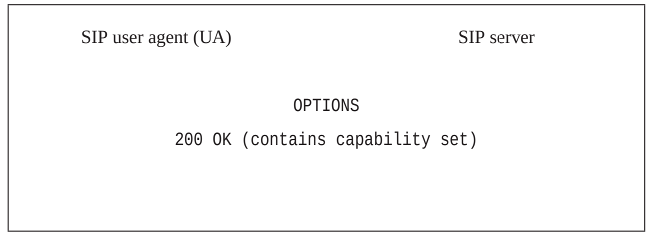

Figure 6. Options Operation

The replied SIP 200 OK response carries the context information. The body of the response can contain any kind of format as long it is specified in the `payload-type` header field. For our prototype the context information is a simple ASCII string. The `payload-type` field is set accordingly to the content. The initiator displays the context of the body to the user. The User Agent interacts via the command line with the user who decides whether to proceed with the regular invite procedure or to discard this process.

The main benefit of extending the existing mechanism is the high reuse of existing code in the SIP entities. Only very few changes have to be made to the internal processing and structure. The original invite process has to be changed such that after a call setup first an OPTIONS is created and sent to the addressee of the call.

During the construction of the method the new header field is inserted. The sequence of operations to create and send the method does not have to be modified. On the receiving side the operator that performs the message parsing has to be extended. On detection of the `Context` header field a new operation is executed, which retrieves the actual context of the user. This is currently done from a database.

Additionally, the building process of the response message is also modified. The context information is added into the body of the message instead of the capabilities. The sending of the response follows the regular process. Based on the type of payload the receiver of the response message enters a new state. Before the state is entered the content of the body is displayed to the user. Two alternatives ('proceed' or 'discard') are presented for decision. Depending on the user's choice one of the regular operations is executed. Additionally, services such as a context-aware call completion can be activated.

A drawback of this approach is the implicit use of an existing SIP method. A receiver of the request which is not capable of handling the context request will send a normal response. The requestor will have to analyze the content of this response. The decision whether the response contains the context information or the standard capabilities is based on the content. The parsing and analyzing takes time and processing power. An explicit identification would be preferable.

4.4.2 Context Query using CONTEXT Methods

A new method that directly refers to the query for context information would enable an explicit signaling. Therefore, we defined a new CONTEXT method. The method provides a similar functionality as described for the OPTIONS method. However, SIP entities unaware of this method will reply with a 405 Methods Not Allowed error response. This unambiguously indicates that the called party cannot provide the required information.

A new state has to be inserted into the FSM of the UA. The new state is entered after the user has entered the caller's address. The constructor for the new CONTEXT method is invoked and the message is sent. Afterward, the requesting side waits until it receives a 200 OK or a 405 Methods Not Allowed response.

On the receiving side a new operator was inserted in the UA to enable listening for incoming CONTEXT methods. Upon receiving this method the content is parsed and the queried information is gathered. As described for the extended OPTIONS method the information is received from a database. A 200 OK response message is issued with the context information. Since the initial sender of the CONTEXT method is explicitly awaiting the context in a response message the incoming response can be parsed without ambiguities.

The use of new method for context sharing has a slightly higher implementation effort. It also enlarges the number of SIP methods. However, the handling of SIP entities that are unaware of the new method is similar to the extended OPTIONS message. Methods unknown to the entity are not processed but routed. The benefit of the new CONTEXT method is the explicit meaning which eases the internal handling at the receiver side.

4.4.3 Event Mechanism using SUBSCRIBE/NOTIFY

Both, the extended OPTIONS and the new CONTEXT method approach use SIP in-band signaling. The SIP event mechanism provides an alternative approach to share context information. The methods SUBSCRIBE and NOTIFY offer the necessary functionality in a SIP environment. After an event as specified in the SUBSCRIBE message occurred, a NOTIFY is sent to the subscriber. Figure 7 shows a message sequence of a subscription and notification.

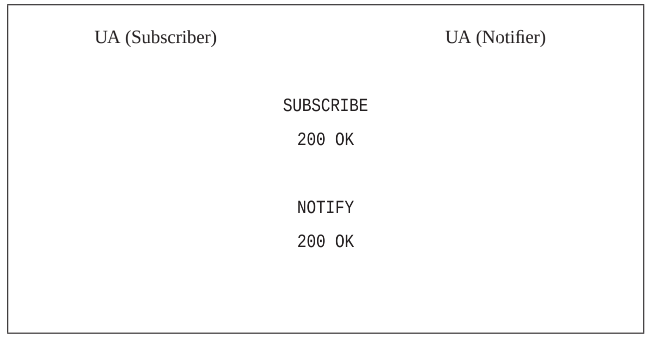

Figure 7. Subscribe/Notify Operation

The notifier is obliged to send the actual status of the event to a subscriber immediately after the subscription has been acknowledged. A synchronous behavior can be achieved this way. The subscription mode is especially interesting when used in conjunction with a Context Server. The user application subscribes to certain context changes. The server handles multiple open requests for context changes.

Moreover, this behavior is also suitable for asynchronous services which do not necessarily require the context information immediately. A context-aware call completion service can utilize the subscription mechanism to be informed when a user becomes available again. The criteria for availability is in this case not restricted to the state of the phone device but depends on the user's state.

The event framework does not provide any predefined event types. It is left to the application designers to define the necessary *event packages*. These packages should be registered with the IANA. However, for our prototype the proprietary event package name 'context' has been used. The defined package allows to query for different levels of context information. Additionally, a user name can be specified to interrogate the context of a particular user from a Context Server. The header field `Event` specifies the event package the requestor is subscribing for. The receiving application has to parse the `Event` header field and to create an subscription if the event type is supported.

After a context change has taken place a *SubscribeManager* cycles through all active subscriptions. The subscriptions that match are further processed. NOTIFY messages are sent to the subscribers with the according information. The body contains the information in the specified MIME format of the event package. The benefit of the event mechanism is the loose coupling with the existing applications. However, it is not well suited for a prompt request/response mechanism.

4.5 Privacy Issues

The shared context information convey very private information to the requestor. It is a legitimate claim of the user to control its context information. In distant communication the caller often asks (implicitly) for the user's context. The possible answers depend on the current situation of the user and the relationship to the questioner. Each caller is categorized into a specific member group and receives the answer in an appropriate detail level. The following example is meant to illustrate this. The question for the actual location of the user is answered to trusted communication partners with a detailed information about the room number in a building of a company in a certain country. For a semi-trusted caller the context information conveys just the information about the country the user is currently in. An untrusted caller will only get the information that “the user is not available”.

The technical equivalent should provide a policy system to support a similar concept of context hierarchy levels. Authentication mechanism to assure that only legitimated users get a certain level of information are used. A variety of authentication schemes are available. The SIP core standard describes an authentication mechanism for SIP using an adapted HTTP authentication scheme. A receiver of a request message can challenge the sender to authenticate itself. A proxy server receiving the CONTEXT request sends a 401 Proxy Authentication Required message back and the requestor answers with an new CONTEXT request but includes the required credentials. This mechanism can be used for all three suggested approaches.

Another sensible area is the possibility to use the proposed approaches for *surveillance* of the user. The request for the context is performed without notifying the called party. It is an intention of the design not to disturb the user. However, a periodic (silent) request for the callee’s context draws a good picture of the callee’s habit. Especially, for employees this would offend asserted labor law. Requests with an excessively high frequency of inquiries should be detected and trigger an alarm. Methods known from Intrusion Detection System provide valuable mechanisms.

Eavesdropping is also a possible threat to the exchange of context information. Cryptographic methods provide protection of the sensible context data. SIP messages can be protected with S/MIME in an end-to-end fashion. Extensions to the S/MIME protection scheme will allow to encrypt messages between end-systems and proxy servers as well as between proxies on a hop-by-hop basis [13].

5 Summary & Conclusion

Interpersonal communication face-to-face over distance is an integral part of our culture. Humans learn conversation techniques and conventions since very early ages. Context information about the situation of the communication partner as well as about the content are proven and implicitly used part of communication. Current technical means do not provide an adequate equivalent for this. The proposed approach in this paper tries to narrow the gap in the application domain of communication services. Different methods to share context information in SIP-based IP Telephony have been analyzed, implemented and evaluated.

References

[1] Vista Seek – The IP-based Local Position System. http://www.ivistar.de.

[2] P. Auer. Kontextualisierung. *Studium Linguistik*, pages 22–47, 1986.

[3] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *IEEE INFOCOM*, pages 775–784, Tel-Aviv, Israel, Mar 2000. IEEE Computer Society Press.

[4] M. Bedworth and J. O’Brien. The omnibus model: A new architecture for data fusion? In *Proceedings of the 2nd International Conference on Information Fusion (FUSION’99)*, Helsinki, Finnland, July 1999.

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

[6] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle. Session Initiation Protocol (SIP) Extension for Instant Messaging. RFC 3428, Dec. 2002.

[7] A. K. Dey. *Providing Architectural Support for Building Context-Aware Applications*. PhD thesis, Georgia Institute of Technology, December 2000.

[8] W. Elmenreich. *Sensor Fusion in Time-Triggered Systems*. PhD thesis, Technische Universität Wien, nov 2002.

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

[10] M. Goertz, A. Perez, R. Ackermann, A. Mauthe, and R. Steinmetz. Location Sensing using RADAR. Technical Report TR-KOM-2003-09, Multimedia Communications Lab, Darmstadt University of Technology, Sept. 2003.

[11] R. Guha. *Contexts: A Formalization and Some Applications*. PhD thesis, Stanford, 1991.

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

[13] K. Ono and S. Tachimoto. Requirements for End-to-middle Security for the Session Initiation Protocol (SIP). Internet Draft, Oct. 2003. Work in progress.

[14] A. B. Roach. Session Initiation Protocol (SIP)-Specific Event Notification. RFC 3265, June 2002.

[15] J. Rosenberg. A Presence Event Package for the Session Initiation Protocol (SIP). Internet draft, July 2003. Work in Progress.

[16] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. *RFC 3261*, June 2002.

[17] A. Schmidt, M. Beigl, and H.-W. Gellersen. There is more to context than location. *Computers and Graphics*, 23(6):893–901, 1999.

[18] A. Schmidt, A. Takaluoma, and J. Mntyjrvi. Context-aware telephony over wap, 2000.

[19] 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.

[20] TINA-Consortium. TINA-C Glossary of Terms, 1997.

[21] Vovida Networks, Inc. Vovida vocal system. http://www.vovida.org/.

[22] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992.