This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2007 proceedings.

An Intelligent Assistant for Context-Aware Adaptation of Personal Communications

Bassam El Saghir Noel Crespi
RS2M Department
GET / Institut National des Telecommunications
Evry, France
{bassam.el_saghir, noel.crespi} @int-evry.fr

Abstract— **Personal communications are facing many challenges created by mobility and convergence in today's communication networks. People often find themselves interacting with their devices in attention-constrained environments and deal with a bewildering variety of communication services, devices and access technologies. Although context awareness seems to be the best answer to these challenges, most of the already developed context-aware solutions did not make their way to the consumer market because they focused on context provisioning (acquisition and modeling) and fell short from proposing concrete architectures for context-based adaptation of user communications. In this paper, we discuss the basic requirements for communication adaptation and we define the main characteristics of a communication session. Then, we propose INCA (Intelligent Network-based Communication Assistant) by describing its architecture and its behavior according to a previously defined reference scenario.**

I. INTRODUCTION

The concept of personal communications is experiencing a radical evolution in comparison to what it was like in the early days of telephony. At the beginning, people were dealing with fixed phones which were tied to specific locations (rather than specific individuals) and communicating with others was merely a matter of lifting the handset and dialing the required phone number. This has changed, today, due to two factors which have deeply impacted the telecommunications industry: mobility and convergence.

Mobility has been mostly associated with the advent of wireless communications. Communication devices which were previously bound to specific locations are now following users almost everywhere, allowing them to be reached anywhere and anytime. The problem, however, is that these devices can solicit users at unpredictable (and even inconvenient) moments. Furthermore, users need to interact with these devices in attention-constrained environments where they could be involved in other activities (for example driving a car or having conversation with other people). Since user attention is a limited resource and should be spared as much as possible, context aware solutions must be devised in order to assist him in managing his communications. Such solutions may provide context sensitive notifications for the user through customization and filtering, and may even perform some communication tasks on his behalf such as automatic answering or redirection. These solutions spare the attention of the user by reducing his explicit interaction with his device, allowing him to focus on more important activities.

Convergence is another important driver in the telecommunications field. It is occurring at three levels simultaneously: service, terminal and network levels [4]. At the service level, providers are proposing rich-content and bundled services that extend well beyond voice to include text, image, audio and video. Examples of such services are PoC (Push-to-talk over Cellular), IM (Instant Messaging), MMS (Multimedia Messaging Service), video streaming, online gaming and access to various multimedia content such as news, sports and music. On the other hand, recent advances in electronics and miniaturization allowed terminals (which were historically optimized for one or few specific tasks) to become increasingly multifunctional as they encompass multiple devices: mobile phone, PDA, game console, music player, video camera, radio, GPS and TV receiver. This, in turn, is coupled to multiple access technologies such as GSM, GPRS, UMTS, WLAN, xDSL and cable. While all these technologies and services allowed the realization of previously unimaginable scenarios (such as accessing emails over mobile phones via WLAN or GPRS), they also created confusion for the average user who cannot always decide upon the best way to handle a particular communication. Again, context-aware solutions solve this problem by gathering context information such as user location, device capabilities and network status and using this information to optimize handling of user communications.

Consequently, context awareness seems to be the best answer for the challenges created by both mobility and convergence. However, although the context-aware computing paradigm has been introduced fifteen years ago [17], very few context-aware solutions made their way through from research labs to the consumers market. This is especially true for the personal communications domain, in spite of the readily available network infrastructures. This due to the fact that most research has been focused on the first steps in the design of context-aware solutions, namely modeling and provisioning context information, and few efforts has been put on the next steps (using context information to adapt user communications). We believe that focusing on the adaptation issue is very important for building successful context-aware solutions and that tackling this issue properly requires not only a good knowledge of the context adaptation requirements, but also a better understanding of the personal communications paradigm itself.

We envision a network-based agent capable of communicating with information servers (e.g. presence server [3], context server for providing context information of the user and his surrounding environment) and application servers (e.g. voicemail server, PoC server [5]). This agent intercepts all incoming and outgoing signaling flows for the devices of a given user and uses his context information to assist him by adapting, filtering, redirecting, and even initiating communication sessions on his behalf.

In this paper, we introduce INCA, an Intelligent Network-based Communication Assistant based on SIP (Session Initiation Protocol) [13]. We begin by defining a reference scenario in section 2 to introduce some of INCA capabilities. Then, we discuss basic requirements for communication adaptation in section 3 and we define the six main dimensions of a communication session in section 4. The discussions in these two sections will be used to define INCA's core architecture and functions. Then, we discuss INCA's contribution with respect to related work in section 6 and we end this paper with a conclusion and an overview of future work.

II. Reference Scenario

In this section, we present a scenario for a communication between two users, Alice and Bob, through Bob's own INCA (which will be simply referred to as "assistant"). This scenario is represented in Fig. 1.

Bob is a salesman who is currently traveling for a meeting. He therefore configured his assistant to forward all incoming calls on his office phone to his personal mobile phone. Alice calls Bob during his meeting. His assistant detects Bob's current situation (by querying the presence server) and infers that it must alert him to the call as silently as possible. It also detects that he is currently taking notes on his laptop PC (by querying the context server). So, it decides not to directly forward Alice's call to his mobile phone. Instead, it displays a silent visual message on Bob's PC alerting him to her call and prompting him to choose between accepting and rejecting it. Bob chooses to reject the call and his assistant prompts Alice to leave a message for him on his voicemail server before

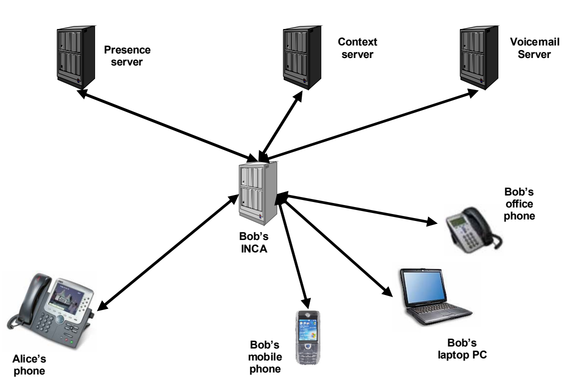

Figure 1. A reference scenario

disconnecting her. When Bob finishes his meeting, his assistant detects his situation change (through the presence server) and infers that he is ready to listen to the message left by Alice. So, it decides to alert him by ringing his mobile phone. When Bob answers, his assistant connects him directly to his voicemail server so that he can listen to Alice's message.

III. Basic Requirements for Communication Adaptation

The above scenario allows us to infer some basic requirements regarding INCA from an end-user perspective. We also identified other requirements which, along with the end-user requirements, will have significant impact on INCA's both architectural and functional aspects:

- *Context awareness:* the assistant should be responsible for fetching information related to the user and his surrounding environment and for detecting relevant changes in this information.
- *Autonomy:* the assistant should be able to initiate actions without explicit user solicitation, by referring only to the context information available to him, such as automatically contacting the user at the end of a meeting to alert him about his messages.
- *Proactivity:* the assistant should be able to intervene before a particular action is executed, such as intercepting a call before it reaches the user mobile phone and alerting him through his PC.
- *Reactivity:* the previous two requirements might involve sophisticated information processing (such as reasoning) that introduce delays in the assistant reactions. While these delays may be tolerated for complex tasks, they are completely unacceptable when dealing with simple tasks (e.g. unconditional forwarding) which require fast execution in order to avoid excessive waits for the average user. Therefore, the assistant must ensure that such tasks must be dealt with separately and in a "reactive" manner, i.e. they must be executed as soon as they are requested and with minimal delays.
- *Adaptability:* the assistant should be able to adapt the services that it renders to the user according to resource constraints. For example, if it needs to provide a complex service which requires more resources than it is currently available, it would attempt to provide a "scaled-down" version of the service instead of not providing it at all. This requirement is also true regarding time constraints.

IV. The Six Dimensions of a Communication Session

As we noted in the introduction, designing successful communication adaptation solutions requires a good understanding of the personal communication paradigm itself. This can be achieved by analyzing the evolution of this paradigm from its early days till now. We suggest explaining this evolution by using the concept of dimensions. In fact, the early communication paradigm relied on a single dimension (namely the address or *where* dimension) and later evolutions of this paradigm resulted effectively from the addition of more dimensions. We identified six dimensions which characterize a communication: *where, what, who, which device, when and why* (also known as the six W's). The first three dimensions have already been introduced in personal communications; the last three are yet to be. These dimensions are discussed below, using the chronological order in which they were (or we believe they will be) incorporated in the personal communication paradigm (Fig. 2):

- *Where:* This dimension represents the address of a communication endpoint. Historically, it was the first parameter to characterize a communication as it can be seen in legacy fixed networks (placing a call in these networks requires only dialing a phone number).
- *What:* This dimension represents the type and properties of a session. It was effectively introduced in 2nd generation mobile networks where the user must determine not only *where* to call (i.e. the destination phone number), but also *what* type of communication to use (a normal voice call or a SMS). Later, this dimension gained further attention through the development of dedicated protocols such as SDP (Session Description Protocol) [15].
- *Who:* This dimension represents the identity of the agent involved in the communication, whether it is a person or an automaton (e.g. a voicemail server). Contrary to the popular belief, the *who* dimension did not exist in legacy networks: a user who dials a fixed phone number does not determine *who* is to be contacted, but rather *where* the communication will be received, because the destination phone could be answered by the intended person as well as by any other person nearest to the phone when it rings. We believe that this dimension was effectively introduced with the Universal Resource Identifier (URI) concept in the Session Initiation Protocol (SIP) [13], since the URI could be used to identify and reach a particular person regardless of his network address, device type or even his physical location.
- *Which device:* This dimension represents the terminals that connect the user to the network. It helps determine the terminal through which the user will be contacted based on the capabilities of the available terminals and their modes of interaction with the user (as we have seen in the reference scenario). This dimension is becoming more and more important with the proliferation of widely heterogeneous communication devices, and we believe that it will soon be the next dimension to be introduced in personal communications. That is why we proposed in a previous work [1] a framework for indicating terminal capabilities in the IP Multimedia Subsystem (IMS), an IP-based next generation network for providing mobile multimedia services.
- *When:* This dimension represents time-related aspects of a communication session, such as start time and duration. Many interesting services are based on this dimension, especially services that involve customization of session start time. For example, a terminal might wait for the availability of a good wireless connection before sending a large MMS, or a message might be delivered to a recipient when his presence status is changed to “available”. This contrasts with today's services where the user has only one option for determining the start time of his communications, namely the “immediate” option. Therefore, we believe that temporal customization will be one of the major steps in the evolution of personal communications.
- *Why:* This dimension represents the reason or the purpose of a communication session, and could be conveyed in fields such as the session subject. It helps the user decide whether to accept an incoming session or not, but could also be used to justify sessions that are automatically initiated by agents (for example the session initiated by Bob's INCA to connect him to his voicemail server in the reference scenario).

A similar classification has already been applied to context information, as it can be seen from the five W's of context in [14]. This implies that it is possible to establish a mapping between a communication dimension and the context information which is involved in its adaptation. It also implies that any context-based communication adaptation solution should reason in terms of these dimensions.

Fig.1 shows the six dimensions in chronological order and their description along with their mapping to headers in the SIP/SDP signaling messages. The first three dimensions are fully represented in SIP or SDP, while the last three are partially represented. Nevertheless, we believe that SIP/SDP is the most convenient protocol for implementing a dimension-based adaptation solution.

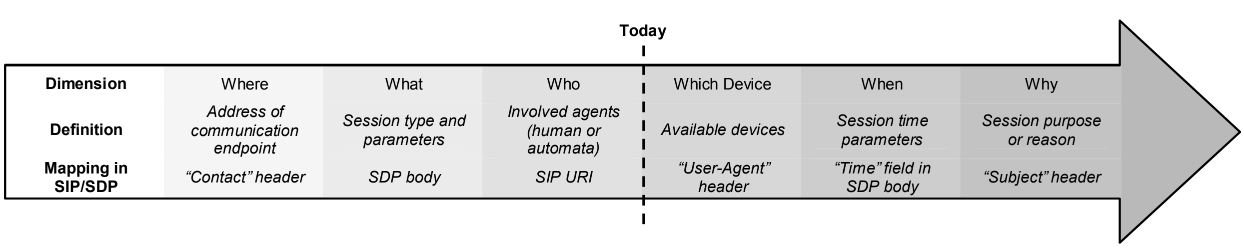

Figure 2. The six dimensions of a communication session

V. PROPOSED SOLUTION

In this section we propose a SIP-based context-aware agent for communication adaptation named INCA (Intelligent Network-based Communication Assistant). Its role is to assist the user in managing his personal communications by intercepting all SIP messages between the network and his devices and modifying them according to the user's context. We begin with an overview of INCA followed by a description of its layered architecture. Then, we show how the reference scenario presented in section 2 could be applied to INCA and we demonstrate its compatibility with the requirements defined in section 3.

A. Overview of INCA

INCA consists of a SIP back-to-back user agent [13] with a core based on a layered architecture. We define four layers, ordered in increasing level of complexity: message layer, operation layer, dimension layer and user layer.

INCA relies on the concept of "interaction" which consists of three parts: event recognition, reasoning and action execution. The execution of an interaction requires one or more layers depending on the interaction complexity. Simple interactions are entirely performed by the lowest layer only (Fig. 3(a)). More complex interactions require more layers, where the number of layers depends on the interaction complexity. If an interaction involves two or more layers, it is performed as follows (Fig. 3(b)). Each layer (starting from the message layer) performs event recognition and relays its results to the layer above until the topmost layer is reached. This latter performs the reasoning part and decides on actions which are relayed to layer below. Each layer then interprets these actions before relaying them to the level below until the message layer is reached.

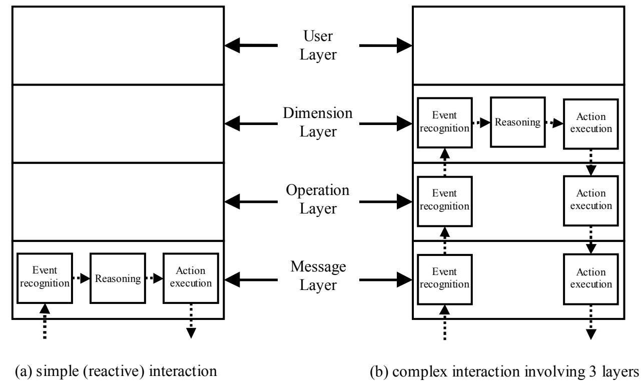

Figure 3. Execution of simple and complex interactions

B. Description of INCA layers

As it can be seen from Fig. 3, each of the four defined layers features the same modules: event recognition, reasoning and action execution. We describe each of these layers below along with the roles of the corresponding modules.

1) Message layer

This layer deals with message-level interactions, also called "reactive" interactions because they require simple yet fast processing. Typical interactions in this layer include replying to SIP requests and redirecting incoming messages (e.g. unconditional forwarding).

This layer communicates directly with the INCA SIP stack. Event recognition consists of parsing SIP messages and extracting their headers and values. Reasoning consists of reactive rules that analyze these values in order to determine header values for new SIP messages to send (if any). Action execution consists of structuring these values into SIP messages which are forwarded to the SIP stack.

2) Operation layer

An "operation" is defined as an exchange of a sequence of SIP messages in order to realize a meaningful action. It could be a simple transaction (a SIP request and response), or a series of interrelated transactions such as Third Party Call Control (3PCC) Session Establishment [9].

The operation layer is responsible for maintaining SIP relations with user devices and network servers (e.g. presence server, voicemail server). A SIP relation could be a SIP dialog [13] or a subscription with its related notifications [12]. Event recognition consists of acquiring parsed SIP messages from the event recognition module of the message layer and translating them into operations. These operations are fed into the reasoning module in order to update SIP relation states (i.e. dialog states and subscription states) and determine potential operations to be executed. Action execution consists of transforming these potential operations into SIP messages sequences which are relayed to the action execution module of the message layer.

3) Dimension layer

This layer is responsible for context acquisition and communication adaptation. Its name comes from the fact that it reasons in terms of dimensions when dealing with both context information and communication adaptation. In fact, it classifies context changes according to the six dimensions discussed in section 4 and uses this classification to plan actions which consist of dimensional changes to be applied to communication sessions. Event recognition consists of transforming operations from the layer below into changes in context dimensions [14]. Reasoning consists of applying these changes to an internal world model representing the user, his surrounding environment, his available devices and the network services at his disposal. This results in session dimension changes which could be either applied to existing sessions or used in the creation of new sessions. Action execution consists of transforming these dimensional changes into operations which are forwarded to the action execution module of the operation layer. If the reasoning module needs additional context information in order to produce a decision, it issues a request for information which is translated by the action execution module into corresponding operations (such as subscription for notifications [12]) and forwarded down to the operation layer.

4) User layer

The goal of this layer is to provide user-centric adaptation according to user state, preferences, and available services. Event recognition at this layer consists of translating context

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2007 proceedings.

dimension changes from the layer below into changes in user state (e.g. activity, current object of attention). These changes are forwarded to the reasoning module where they are used in conjunction with user information (e.g. profile and preferences) and service information (e.g. service profiles and models) in order to decide upon adaptation tasks to be executed. These adaptation tasks are then translated by the action execution module into changes in communication dimensions which are forwarded to the dimension layer. If the reasoning module needs additional user-related information in order to produce a decision, it issues a request for information which is translated by the action execution module and forwarded to the layer below.

C. Revisiting the reference scenario

Fig. 4 shows how the reference scenario of section 2 is realized using the SIP/SDP protocol (the signaling messages between INCA and information servers are omitted because of space limitations.)

Each interaction in this scenario requires a different number of layers depending on its complexity. Session termination requests and responses (messages 14 to 17 and 25 to 28) are typically handled by the message layer. Session redirection to voicemail server (messages 6 to 13) involves 3PCC call handling and is therefore performed by the operation layer. Session forwarding to the laptop PC (messages 1 to 3) involves querying the context server about currently active devices (in order to determine the device which has the user attention); so this interaction is performed by the dimension layer. Automatic notification of messages left (messages 18 to 24) requires getting information about the user state in order to assess his readiness to listen to messages and is therefore performed by the user layer.

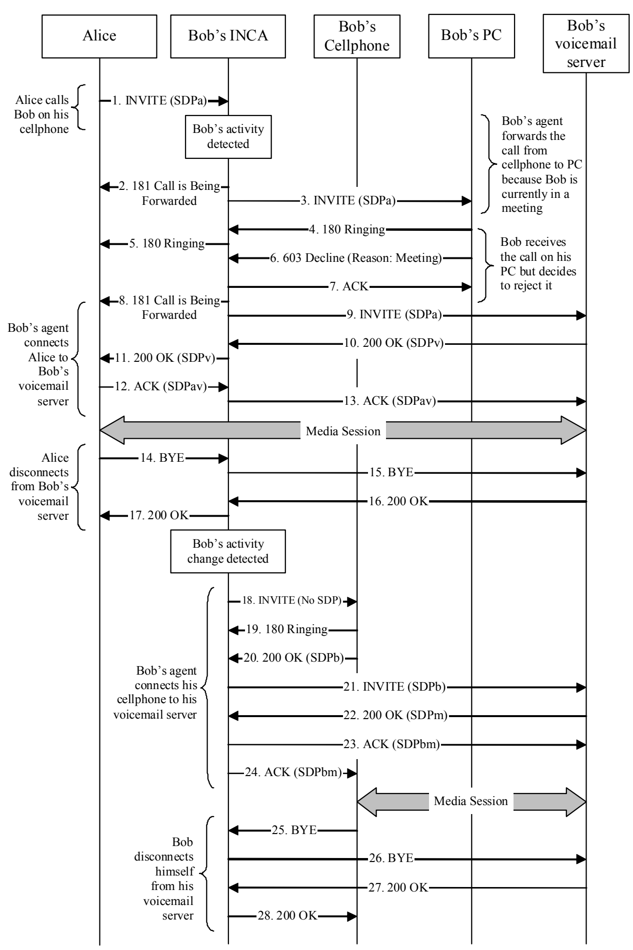

Figure 4. SIP signaling flow for the reference scenario

We note that, in each of the above interactions, the topmost layer involved always uses the services of the layers below. For example, the action execution part of the last interaction (automatic notification of messages left) starts with a detection, at the user layer, of the fact that Bob is ready to listen to his left messages. This is translated by the action execution module to a connection setup action with the voicemail server. This action is forwarded to the dimension layer where the network address of the voicemail server is fetched. Then, it is forwarded to the operation layer where it is converted to a sequence of SIP messages for session establishment. This sequence is finally transmitted to the message layer for execution.

D. Compliance with previously defined requirements

In this section, we show how INCA comply with the adaptation requirements defined in section 3. *Context awareness* is realized by the user and dimension layers since they can query information servers for context information whenever they need it. *Autonomy* is also fulfilled by these two layers since they can decide upon actions by referring to reasoning rules that rely on automatically acquired context information (rather than explicit user input). *Proactivity* is achieved by intercepting all signaling messages between the network and the various devices of a particular user and by triggering interactions which modify these messages before they actually reach their destinations.

Reactivity is made possible by the principle of layered interaction execution. In fact, the reasoning process (which maps events to actions) is not monolithic but distributed among layers. Therefore, interactions are entirely taken in charge by the lowest layer if they require simple and fast reasoning (e.g. unconditional forwarding) and are not delegated to higher layers unless they require more complex (but slower) reasoning.

Adaptability is also made possible thanks to the layered execution principle. For example, if a specific interaction needs the message layer and the operation layer in order to be executed but could not use both of them due to resource or time constraints, the event recognition module of the message layer could redirect its output to the reasoning module of its own layer (instead of forwarding it to the event recognition module of the operation layer). The reasoning module would then apply special rules which lead to default actions such as proxying SIP requests or sending error responses. This example of behavior of the message layer could also be performed by any other layer above when subjected to resource or time constraints.

VI. RELATED WORK

Many context-aware systems have been developed to demonstrate the benefits of context-aware computing. One of the earliest systems of this kind was the Active Badge Location System [17] which provides phone call redirection based on the location of the called person. Later research efforts, however, focused more on developing infrastructures and mechanisms for context provisioning and did not sufficiently consider context-based adaptation. For example, the Context Broker Architecture (CoBrA) [7] was primarily developed for supporting context-aware systems by addressing issues such as context modeling and reasoning, knowledge acquisition from different context sources and user privacy protection. The Service-Oriented Context-Aware Middleware (SOCAM) proposed in [6] aimed at supporting acquisition, discovery, and interpretation of context information to build context-aware services.

Even the few existing work that deals with the adaptation of personal communications did not propose any detailed architectures for the entities which are involved. For example, Ref. [2] defines a "Person-Centric Context Aware System" by proposing a generic block architecture only. This is also true for the "Communication Agent" that was proposed by Schmidt et al. in [16] for handling communications on behalf of the user.

The layered architecture we proposed in this article is inspired from the "logic-based subsumption architecture" [10]. This AI (Artificial Intelligence) architecture allows the decomposition of an agent structure into layers corresponding to different levels of behavior (or interactions). More complex layers in this architecture not only depend on lower, more reactive layers, but could also influence their behavior. This allows us to take into account performance issues that were not considered in previous work such as reactivity and adaptability.

VII. CONCLUSION AND FUTURE WORK

In this paper, we described INCA, an Intelligent Network-based Communication Assistant that uses context awareness to manage SIP-based personal communications. We first presented a reference scenario involving INCA and we defined the basic requirements for context-based communication adaptation. Then we defined the six main characteristics (dimensions) of a communication session and we proposed an agent architecture that takes into account what we have defined in sections 3 and 4. We finally discussed the contribution of INCA with respect to the related work.

Future work includes defining the three modules in each of the four layers. This includes, among others, a detailed study of the relationship between context dimensions and communication dimensions in order to specify the dimension layer modules. We are also investigating the knowledge modeling issue in these layers, especially the possibility of using an ontology language such as OWL (Ontology Web Language) [11] to model information such as dialog states (operation layer), context and communication dimensions (dimension layer) and user preferences and services (user layer). We are also studying implementation issues of INCA, especially using semantic web frameworks such as Jena [8] for implementing INCA's layered core.

REFERENCES

[1] B. El Saghir, N. Crespi, "A New Framework for Indicating Terminal Capabilities in the IP Multimedia Subsystem", IEEE GLOBECOM-ISET, November 2006.

[2] W. Li, F. Kilander, C. G. Jansson, "Toward a Person-Centric Context Aware System," Workshop on Requirements and Solutions for Pervasive Software Infrastructures, May 2006.

[3] 3GPP TS 22.141, "Presence service; Stage 1 (Release 7)," v7.0.0, December 2005.

[4] 3G Americas, "Convergence: An Outlook on Device, Service, Network and Technology Trends," July 2005.

[5] 3GPP TR 23.979, "3GPP enablers for Open Mobile Alliance (OMA); Push-to-talk over Cellular (PoC) services; Stage 2 (Release 6)," v6.2.0, June 2005.

[6] T.Gu, H. K. Pung, D. Q. Zhang, "A service-oriented middleware for building context-aware services," Journal of Network and Computer Applications, Vol. 28, Issue 1, pp. 1-18, January 2005.

[7] H. Chen, "An Intelligent Broker Architecture for Pervasive Context-Aware Systems," Ph.D. Dissertation, University of Maryland, December 2004.

[8] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, A. Wilkinson, "Jena: implementing the semantic web recommendations," Proceedings of the 13th international World Wide Web conference on Alternate track papers and posters, Session Industrial Practice I, May 2004.

[9] J. Rosenberg, J. Peterson, H.Schulzrinne, G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)", RFC 3725, April 2004.

[10] E. Amir, P. Maynard-Zhang, "Logic-based subsumption architecture," Artificial Intelligence, Vol.153, Issue 1-2, pp.167-237, March 2004.

[11] D. L. McGuinness, F. van Harmelen, "OWL web ontology language overview," W3C Recommendation, February 2004, http://www.w3.org/TR/owl-features/.

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

[13] J. Rosenberg et al., "SIP: Session Initiation Protocol," RFC 3261, June 2002.

[14] G.Abowd and E. Mynatt, "Charting past, present, and future research in ubiquitous computing", ACM Transactions on Computer-Human Interaction (TOCHI), Vol. 7 , Issue 1, pp. 29-58, March 2000.

[15] M. Handley, V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[16] A. Schmidt, A. Specker, G. Partsch, M. Weber and S. Hoeck, "An agent-based telecooperation framework," In Proceedings of CoBuild'98, Darmstadt, February 1998.

[17] R.Want, A.Hopper, V. Falcao, J Gibbons, "The active badge location system," ACM Transactions on Information Systems, Vol. 10, Issue 1, pp. 91-102, January 1992.