CATS: Context-Aware Triggering System for Next Generation Networks

José Simoes[1], João Goncalves[2], Telma Mota[2], and Thomas Magedanz[1]

[1] Fraunhofer Institute FOKUS, Kaiserin-Augusta-Allee 31,
10589 Berlin, Germany
[2] PT Inovacao, Rua Eng. José Ferreira Pinto Basto,
3810-106 Aveiro, Portugal
{jose.simoes,thomas.magedanz}@fokus.fraunhofer.de,
{joao-m-goncalves,telma}@ptinovacao.pt

Abstract. Considering context information, namely location, in order to create new services has become a commercial trend. The innovative approaches that context-aware mechanisms make possible are being targeted by service providers of diverse areas. On the other hand, Service Oriented Architectures play a central role in allowing component reuse and low cost service creation. Together with IP Multimedia Subsystem enable the convergence of telecommunications and web services, allowing the network transport technologies to be abstracted from the services above. By integrating these three technologies, a number of synergies can be explored. Existing services can be easily enriched with context information, made available on a variety of networks and new services can be composed using previously existing building blocks. This paper explains how this integration can be achieved, and demonstrates the potentialities of this architectural paradigm with a prototype service.

1 Introduction

With the rapid advance in technology, it is becoming increasingly feasible for people to take advantage of the devices and services in the surrounding environment to remain "connected" and continuously enjoy the activity they are engaged in, be it sports, entertainment, or work. Such a ubiquitous computing environment will allow everyone permanent access to the internet anytime, anywhere and anyhow. Nevertheless, mobility is just one of the innumerous aspects that will play a preponderant role in the services of the future. In fact, being aware and able to communicate context is a key part of human interaction. Context is a much richer and more powerful concept, particularly for mobile users and can make network services more personalized, adaptable, interactive and therefore useful. Harvesting of context to reason and learn about user behavior will enhance the "internet of services" or "cloud computing" vision, allowing services to be composed and customized according to user context [1]. Combining both concepts, will allow services to be tailored to customer needs, providing richer experiences, thus enabling a whole new set of sensations, improving the user overall Quality of Experience (QoE). By QoE the authors refer to the subjective measure from a user's perspective of the overall value of the service provided.

To support such vision, it is important that services are deployed under Next Generation Networks (NGNs), empowering cross-fertilization scenarios among different access networks. The IP Multimedia Subsystem (IMS) represents a natural response for this dilemma by combining traditional telecommunications concepts and internet service technologies. Furthermore, IMS can be considered to be an overlay control subsystem over heterogeneous networks [2], which makes it a natural choice for converging different access networks. On the service creation point of view, we need an architectural model that technically supports the previous described concepts. Service Oriented Architecture (SOA) is considered as the philosophy of encapsulating application logic in services with uniformly defined interfaces and making these publicly available via discovery mechanisms. Not only the notion of complexity-hiding and reuse but also the idea of loosely coupling services, are part of such ideology.

Based on these principles, we introduce the Context-aware Triggering System (CATS), a service created by the composition of different enablers, over an IMS environment, focusing not only in the technical structure of such application, but also on the business models that can arise from such architectural innovation. The ability to trigger a reaction to a specific occurrence or a set of events is a key functionality in context-aware systems. It can be seen as a standalone or distributed service build over a myriad of variables. In recent applications, it is usually implemented as a notification in a variety of ways: sound, light, messaging, video, etc. A concise survey of the current approaches to provide context-based services over mobile devices will be addressed in section 2. Then, we present in section 3 the proposed architecture and the concepts that allow ubiquitous context-aware service creation. A special focus over the architecture and concept validation is covered in section 4. Finally, section 5 concludes the paper by discussing future work.

2   Related Work

In the past few years, the interest for context-aware services (CAS) has raised increasingly, not only from the industry but also for the research perspective. Google mail was one of the first companies to provide such services by offering targeted advertising and Really Simple Syndication (RSS) feeds according to user demographic information and e-mail content type [3]. From the mobile marketing market, AcuityMobile developed and patented Spot Relevance [TM], a technology with the ability to deliver the right marketing content, to the right person, at the right time, in the right location.

The panorama for research institutes is not that different. VTT Information Technology (Finland) conducted a study to evaluate user needs for location-aware mobile devices [4] where they present user needs under five main themes: topical and comprehensive contents, smooth user interaction, personal and user-generated contents, seamless service entities and privacy issues. This study helped to identify key issues that should be addressed, but didn't propose any improvement. In order to manage various context data and make the best use of this data for application services in ubiquitous environments, Yoon-Ae Ahn [5] proposes a mobile object data management framework for location enhanced applications, which consists of a data collector, a context manager, a knowledge base, an inference engine and a mobile object database. Despite the validity of such an approach, it does not define interfaces

outside a closed environment and does not target other types of context, namely presence, weather, time, etc. The work done by Kukhun et al. [6] presents a location-aware geographical pervasive system that provides mobile users with a service that corresponds not only to their requests but also to their preferences, location and querying time. Although this vision is going towards the authors' ideas, the work only establishes initial ground steps and does not specify any kind of architecture.

By providing an analysis of the requirements of a middleware for context-aware systems, focusing on mobility and privacy issues, the authors of [7] present PACE middleware, whose intent was the design of context-aware applications and solutions for modeling and managing context information. Focusing on mobile environments, the Web, semantics and their convergence with a shared communications world, project SPICE [8] presents a platform, which is capable of context acquisition, context representation, context enabling and use in mobile services. Although the work developed is very interesting in what concerns contexts, it does not target any notification or distribution mechanisms.

In fact, notification mechanisms are getting more prominent, especially with the advent of social networking (communities), as many see it as the new trend in the landscape of communications. The first success example of this interaction model that transmits an "always connected" sensation to the users was RSS, even if the underlying technology is actually based on polling.

In order to meet the demand for popularized application, socialized service of geographical information and solve the sharing, interoperation and integrated application of geographical information in network environments, paper [9] introduces a Service Oriented Architecture (SOA) model, which can conduct distributed deployment, by combining loosely-coupled and coarse-grained application components on the internet according to the demand and effectively support the development of a geographical information service. On the other hand, there are approaches such as [10] which use IP Multimedia Subsystem (IMS) [11] as an architectural framework to provide multimedia services over a packet based next generation network (NGN), where they use IMS application servers and enablers for service delivery. In fact, the idea of extending IMS management to SOA based next generation networks has already been acknowledged at [12].

By combining concepts from the previously enunciated approaches, we present an architecture based on IMS as infrastructure technology and SOA as a software architectural model, which will enable the provision, creation and dissemination of context-aware services for next generation networks.

3   Architectural Approach

The Context-aware Triggering System may be described as a multi context notification service, enhanced with user preferences and social activities. It allows the originating party to create one or more multimedia content items that can be distributed in a myriad of forms according to the recipient(s) set of contexts available, over an

unicast or multicast overlay network. This means that the same trigger may be experienced in different ways depending on the user device, presence status, user preferences or other external factors/contexts. In other words, the main contribution focuses on a framework, which has the capability of casting content into context, allowing personalized, contextualized, adapted and interactive multimedia distribution across heterogeneous access networks.

3.1   General Overview

In order to achieve such service, the proposed architecture adopts a SOA model and it is implemented using an IMS infrastructure in order to provide a network agnostic control layer, facilitating at the end, seamless service delivery across a variety of heterogeneous networks. This approach will provide a way to have a flexible and modular architecture, making it possible to extend in the future by adding extra enablers and consequently add more functionalities. Figure 1 depicts a high level overview of the used components as well as indicates the protocols involved to communicate between themselves.
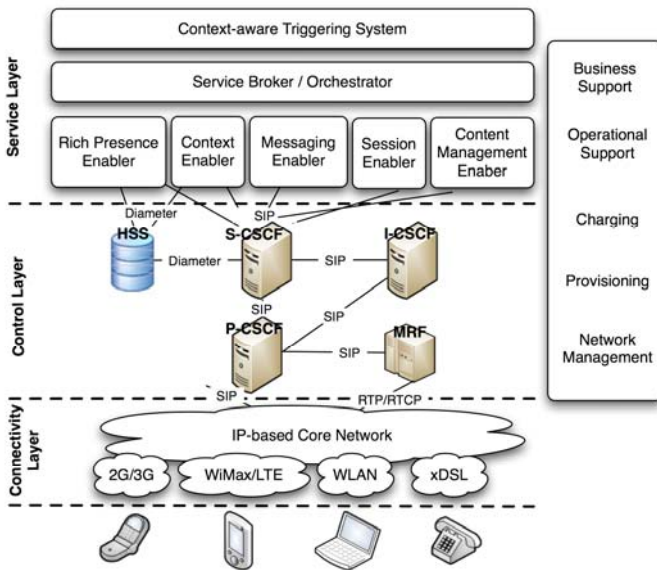

Fig. 1. General overview of the Context-aware Triggering System architecture

3.2   Functional Architecture

In the previous section we introduced the main components, which interact with the CATS; however, it is important to know how each one of them work and how they communicate with each other. Figure 2 details the functional architecture of CATS, focusing only the service layer, where our contributions are most notable.

Although all enablers interact directly with the application server, from figure 2 it is possible to identify that some of them are triggered through the service broker (the Rich Presence and Context enablers). The reason why not all components are directly induced by the service broker is related to the fact that at this stage the Messaging, Content Management and Session enablers specifications still aren't mature enough to allow reuse for any service. Consequently, at this point it makes no sense to provide means for these enablers to be correctly registered and exposed in order to be discoverable and orchestrable, as it would be usual in a SOA environment. Nevertheless, the further specification of these enablers is one of the goals of the authors' research, maximizing the simplicity of the application by exploiting the advantages of a service-oriented architecture. One of these advantages is the possibility that different service providers implement different parts of the system.
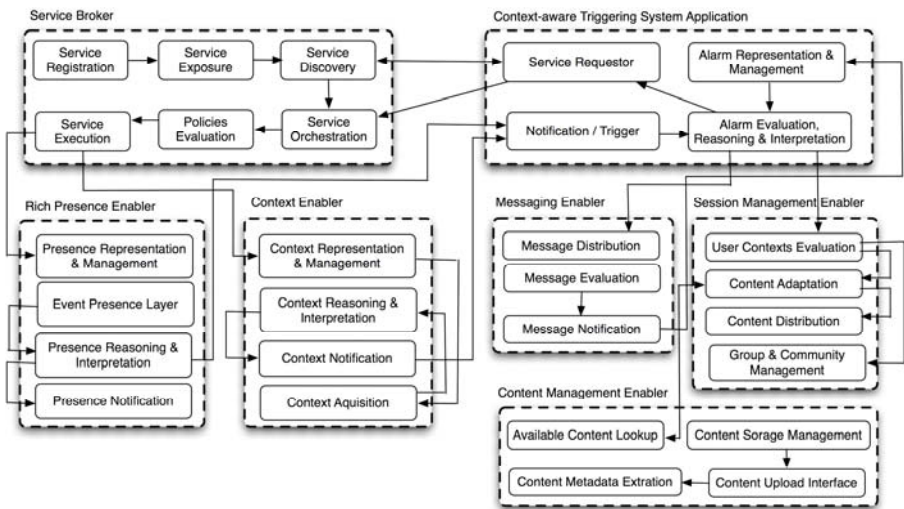

Fig. 2. Functional architecture of the Context-aware Triggering System

In the preceding scenario, we assume that the capabilities, ontologies, semantics and policies of the specified enablers where previously registered and exposed at the service broker. That means that the service broker is able to know how to execute every of the functions of the other components, as well as make them available whenever other components request so. To simplify, in this section we abstract provisioning and charging mechanisms, as well as business and operational support systems, focusing them later on, when convenient, on section 3.3. In order to understand the role of each enabler, table 1 provides a small description of the basic functionalities of each component. A full description of the functionalities and characteristics of these components can be found in [13].

Table 1. Description of the components inside the Context-aware Triggering System

Component	Description
CATS Application	Responsible for managing and representing the triggers defined by the end user.
Service Broker / Orchestrator	Coordinates a group of services that communicate with each other. In this context, services comprise intrinsically unassociated units of functionality that have no calls to each other embedded in them.
Rich Presence Enabler	Offers the basic presence functionalities as well as location-based information.
Context Enabler	Decouples context from application. In general it deals with context acquisition as a mechanism to obtain context from diverse sources, context fusion for merging correlated contextual information, context discovery as a mechanism to locate and access context sources, context diffusion/dissemination for efficiently propagating the context while ensuring availability and reliability [14].
Messaging Enabler	Two main functions: send messages across multi platforms to one or multiple users and allow two-way communication, authorizing users to use the same channels to create, modify or delete triggers
Session Management Enabler	Manages all the user-to-content and, vice versa, the content-to-user relationships (Content Distribution module). In fact, it provides the necessary signaling to deliver a specific content to its consumers.
Content Management Enabler	Responsible for the content provisioning, storage and management. When uploaded, the content may be tagged with metadata supplied by the user or content provider.

3.3   Design Decisions

Although the research in what concerns context brokers, enablers or frameworks is very active, it usually does not cover the entire life chain of context acquisition, reasoning and distribution. Our approach on the other hand focus on the whole cycle, giving particular emphasis to the efficient matching of content to context, as well as distribution mechanisms (context-aware multiparty delivery). In order to better understand our decisions, we separated the problem analysis into four different topics:

Context View

A generic context-aware infrastructure that effectively decouples context from the application and supports many different types of context-aware services requires a flexible model. Such model shall allow diverse context processing components, probably operated by different service providers, to be integrated, enabling the applications to handle only the highest level context. A generic model that satisfies some of these constraints is a producer-consumer, publish-subscribe, broker model [15]. This is the approach used in the CATS framework and its design principles are documented in [16]. All context-processing entities can either be CxC, CxP or a combination of both. In this sense, we can identify two main parts in our context-aware system:

- Context Management subsystem concerned with context acquisition and dissemination.
- Context modeling concerned with manipulation, representation, recognizing and reasoning about context and situations.

Service View
Considering a SOA and IMS approach allows for different service providers and network providers to share responsibilities in the service execution. IMS puts forward a well defined set of interfaces towards the upper layers. The network providers, depending on their business model, may allow third party enabler implementations to interact directly with their IMS infrastature, or for certain key enablers only allow a SOA bus to be set up on top of them. Either way, the synergic possibilities between different service providers and network providers, relying on different business models, are obvious.

One of the consequences of context adaptation, particularly when using SOA, is that adaptation can be partly satisfied by the ability to compose and orchestrate services on the fly, based on user and/or environmental context. As service composition is a dynamic and flexible process, which allows for reconfiguration as the context changes, this actuation will represent an indispensable part of what it means to be context aware. A step toward meeting this challenge was to design a clear separation between the context management supporting architecture and service architecture and to clearly understand the interaction points between context and services. Previous context-aware architectures have shown that incorporating context change decision triggering within the application, service or service enabler is a poor performing design.

Network View
From the network perspective, one key objective of CATS is to investigate the dynamic optimization of content delivery to a group of users based on the group's context; i.e. context-aware multiparty delivery, with QoS guaranteed. The rationale for this work and the associated network's view of the CATS architecture enabling it are presented in detail in [17]. As efficient network support of real-time group communications requires dedicated multiparty networking technologies, such as IP multicast associated with QoS dynamic control, to send the same content simultaneously to multiple receivers, with intermediate routers duplicating packets and allocating network resources (e.g., bandwidth) as needed, CATS takes this features into consideration.

Moreover, the heterogeneity of group members, e.g. in terms of link characteristics, access to network resources, devices used, physical mobility and environment, makes it almost impossible to deliver content to all group members in the same manner. Dynamic adaptation of the multiparty delivery is thus needed to optimize group communications and maximize satisfaction of each individual group member. Hence, although the same content is to be sent to all group members, its delivery needs to be optimized for each user based on his/her context. In that respect, optimizations need to be considered at all levels of the protocol stack, including session, transport and network layers. Furthermore, the context considered to drive the dynamic adaptation of the multiparty delivery needs to encompass not only the networking context of each group member (e.g. link quality and characteristics) but also its environmental

context (device capabilities, physical location, speed, etc.). Naturally, different adaptations can be applied simultaneously for different members of a group receiving the same content, with adaptations for each user being driven by its own networking and environmental contexts. Figure 3 illustrates the network architecture for context-aware multiparty delivery, including context detection and delivery.
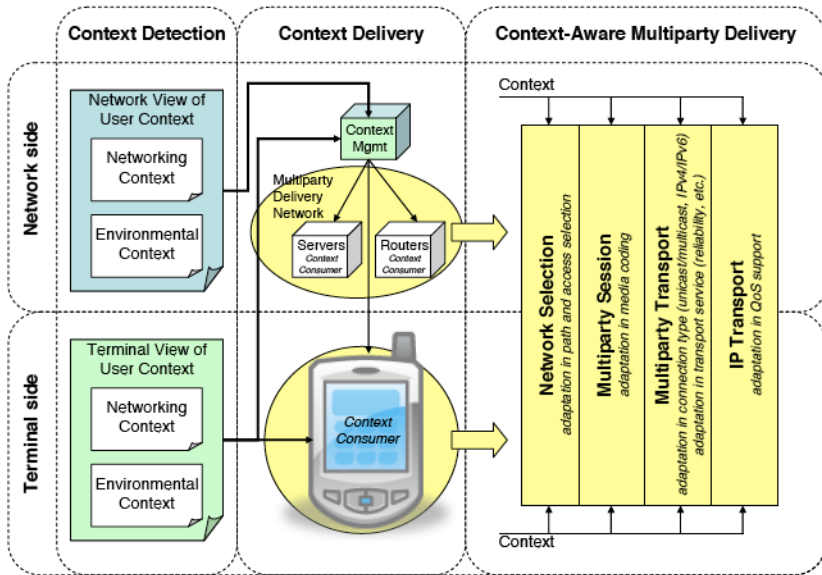

Fig. 3. Context-aware Multiparty delivery

Content View

Our view focuses on four different stages: production, processing, management and consumption. The first stage includes content identification, targeting, selection, and classification. Here, content is carefully prepared, monitored, and very importantly, related content metadata is generated. The subsequent phase includes adaptation of the content and its preparation for mobile use. In order to push the content into the platform, there is a preliminary step to pre-process the content in order to make it suitable for the platform's infrastructure. Then the item, and its metadata are stored in a database, which is part of the content management system. The last step is content consumption. After a group of people, an individual user or a context driven event trigger the consumption of the content, it will be delivered to the end-user terminals for final consumption, via multicast or unicast.

4   Validation

4.1   Use Case: Targeted Advertising

One of the most common use cases is advertising. Using CATS, advertising companies can send targeted campaigns towards their customers or even aiming at new

clients that are willing to receive commercials in exchange for some other benefits (discounts, free goods or even free SMS's or minutes). This brings opportunities not only for operators but also to developers which build applications under other platforms such as social networks. By correlating this new hype (social networks) with the CATS application, the possibilities and amount of collectable contexts are huge, not to mention the increasing target audience that can arise from concepts such as "Friend of a Friend" (FOAF) [1], widely available inside these environments.

A typical scenario would present users in proximity of certain sell-points with information about the goods that are available on saving prices. Moreover, we will assume that there is not a "one-to-one" relationship between the advertising company and the shop, but a "one-to-many". This means that, although the trigger will be triggered by the location, the information targeted for customers will not only take rich presence information and user preferences into consideration, but also other contexts, providing the user with a new type of experience. Although the matching between content and context is still in a premature phase, exploring this topic will enable us to increase the relevancy of a specific trigger towards a specific user (this is done during the inference process).

When the session enabler receives a notification to start a media session with the user, it verifies the user's device settings and network conditions to check whether the content needs to be adapted. If not, the advertised content and probably a discount coupon are sent to the user. Upon entering the shop, the user uses his coupon to get a 20% discount on a brand article. Later on, when the coupon number is inserted in the system, the user's history profile is updated accordingly, leveraging this information for future reference. Despite the fact this architecture can be used for advertising purposes, it would have to be improved to fulfill both advertisers and customers' needs. Moreover, in a world in which the line between users, providers and advertisers is becoming increasingly blurry, a trend best exemplified by the Web 2.0 phenomena, avoiding intrusion and advertisement spam has become very pertinent. Work towards this issue will be addressed by future work.

4.2   CATS Concept Validation

In order to test the concept proposed on this paper, a prototype of the Context-aware Triggering System was developed. The application was developed using JAVA and deployed on a BEA Weblogic SIP Server[2], using both SIP and Hypertext Transfer Protocol (HTTP) protocols. At this stage, it was not possible to interconnect this application with any Broker or Orchestration server, therefore, the discovery process was bypassed in our trials and the service orchestration hard coded on the CATS application to enable service execution between all the enablers.

The validation scenario was setup under FOKUS Open SOA Telco Playground[3] where the following components where used: OpenIMSCore[4], FOKUS Presence

---

[1] Friend of a Friend, FOAF Vocabulary Specification 0.91, http://xmlns.com/foaf/spec/

[2] BEA WebLogic® SIP Server, Converging Internet, SOA and next-generation telecom services, www.bea.com/sip

[3] FOKUS Open SOA Telco Playground, http://www.opensoaplayground.org

[4] Open IMS Playground, www.open-ims.org

Server[5] and Converged Open Messaging Server (COMS)[6]. Once again, no Context enabler was available and consequently, the CATS application was responsible for acquiring the contexts. Session Management Enabler and Content Management Enabler were also not tested at this stage. By using a proper configuration webpage, a user could select which capabilities the trigger should have.

On our tests, we decided that a user should be notified every time a specific friend is in a 1Km distance of his house. Furthermore, the trigger should only be triggered if the user was also in the same region and it is not raining. As a notification mode, the user defined to be alerted via message: via SIP message when online and via SMS otherwise. Moreover, we enabled trigger modification via SMS, so that the trigger would consider not only raining but also snowing into its evaluation. When the user's friend finally moves inside the targeted area, the trigger evaluates the user location, the current weather conditions and when all conditions are verified, its checks the current user presence status. As it was offline, the user was notified via SMS that his friend was in the surroundings. Using such a testing environment enabled us to see that the concept works and can be useful. Nevertheless, this was not performing in a dynamic way like presented in the architecture. Therefore, although the concept was tested, there is still a need to validate the architecture.

5   Conclusions and Future Work

Service creation paradigms are changing as users demand for adapted, contextualized and personalized services, allowing interactivity and easy integration with their social environments, enabling them to access and consume content/services anytime, anywhere, anyhow. Moreover, users want to be in control of their services which require an adaptation of today's business models and technologies in order to support this desire. In fact, to achieve this vision, we need an architectural model that technically supports the previous described concepts. SOA is considered as the philosophy of encapsulating application logic in services with uniformly defined interfaces and making these publicly available via discovery mechanisms. Based on such premises, CATS architecture enables users to design complex and personalized notification mechanisms using a simple and clear interface. Moreover, this architecture allows two-way communications, introducing interactivity into the scenario, enabling users to add, edit or delete a trigger across a set of different platforms (e-mail, SMS, SIP message, XMPP, HTTP, etc.). By using IMS as a control technology, the architecture becomes access network independent and therefore easy to integrate with a different set of devices using multiple technologies.

Furthermore, exploring such an architectural approach enabled the identification of interesting topics that could improve the functionalities, efficiency and scopes of the CATS application. Firstly, the context to content matching, which will improve the relevance of the content recommended or targeted to the end user, depending on the user's current context and profile. Both these dimensions will be relevant to choose content based on its semantic metadata and on its format. The effectiveness of such

---

[5] FOKUS Presence Server, www.open-ims.org/presence
[6] FOKUS Converged Open Messaging Server, www.open-ims.org/coms

improvement is believed to be directly impacted by the completeness of content metadata, the availability of context information, and the existing mechanisms to perceive user preferences. Secondly, the integration with social networks will be addressed under the context enabler; not only to improve context-gathering mechanisms but also to cover business related issues. Thirdly, it showed how interesting this architecture can be for advertising. Correlating this fact with industry reports [18], [19] which indicate advertising will be the business model of the future, makes all sense to explore this idea in the future.

As future work, this architecture will be extended under two parallel branches. The first will be under the European Project C-Cast where a framework to collect data, manage context groups, enable context driven content creation, reason contexts, distribute and efficiently manage context aware multiparty and multicast transport is defined. The second, inside the FOKUS Open SOA Telco Playground where an advertising enabler will be deployed to create a unified multi channel (TV, Mobile and WEB) advertising solution capable of targeting users according to their preferences and needs, concerning intrusion and advertisement spam avoidance by introducing the concept of buddy list, already existent and well developed in the context of social networks, generalizing it for advertising companies.

To sum up, this paper presented an application and architecture which pretend to act as a starting point for future context aware next generation network architectures (supporting both unicast and multicast distribution mechanisms) improving at the end the users quality of experience by addressing topics such as personalization, contextualization, adaptation, interactivity and mobility.

References

1. Baker, N., Zafar, M., Moltchanov, B., Knappmeyer, M.: Context-Aware Systems and Implications for Future Internet. In: Future Internet Conference and Technical Workshops, Prague, Czech Republic (May 2009)
2. Camarillo, G., Garcia-Martin, M.: The 3G IP Multimedia Subsystem: Merging the Internet and the Cellular Worlds. John Wiley & Sons, Chichester (2004)
3. Google Mail Privacy Policies: Targeted Ads in Google Mail (2009),
   `http://mail.google.com/mail/help/intl/en_GB/about_privacy.html`
4. Kaasinen, E.: User needs for location-aware mobile services. In: Personal and Ubiquitous Computing. Springer, Heidelberg (2003)
5. Ahn, Y.: Design of a Mobile Object Data Management Framework for Location enhanced Applications. In: International Conference on Convergence and Hybrid Information Technology, pp. 270–273 (2008)
6. Kukhun, D., Soukkarieh, B., Lopes-Ornelas, E., Sedes, F.: LA-GPS: A location-aware geographical pervasive system. In: 24th IEEE International Conference on Data Engineering Workshop, Cancun, Mexico, pp. 160–163 (2008)
7. Hightower, J., LaMarca, A., Smith, I.: Practical Lessons from Place Lab. IEEE Pervasive Computing Magazine (2006)
8. Zhdanova, A., Zoric, J., et al.: Context Acquisition, Representation and Employment in Mobile Service Platforms. In: 15th IST Mobile and Wireless Communications Summit, Mykonos, Greece, June 4-8, 2006, pp. 64–68 (2006)

9. Shujun, D., Liang, L., Chengqi, C.: Research on Geographical Information Service Based on SOA. In: Proceedings of the IEEE International Conference on Automation and Logistics, Qingdao, China (2008)

10. Brajdic, A., Lapcevic, O., Matijasevic, M.: Service Composition in IMS: A Location Based Service Example. In: 3rd International Symposium on Wireless Pervasive Computing, pp. 208–212. IEEE, Santorini (2008)

11. 3GPP TS 23.228: Service requirements for the Internet Protocol (IP) multimedia core network subsystem (IMS); Stage 1. Release 9, 2008-12-19

12. Blum, N., Magedanz, T., Schreiner, F., Wahle, S.: From IMS Management to SOA based NGN. Journal of Network and Systems Management 17 (2009) (accepted for publication), ISSN 1064-7570

13. full description of the functionalities and characteristics of these components

14. Requirements and concepts for context detection and context-aware multiparty transport. In: Deliverable 6-C-CAST: Content Casting, European Project ICT-2007-216462 (2008)

15. Chen, H.: An Intelligent Broker Architecture for Pervasive Context-Aware Systems, PhD Thesis, University of Maryland, Baltimore County (2004)

16. Goncalves, J., Moltchanov, B., et al.: Context Management Architecture for Future Internet Services. In: ICT Mobile Summit 2009, Spain (June 2009)

17. Antoniou, J., Simoes, J., et al.: Context-Aware Multiparty Networking. In: ICT Mobile Summit 2009, Spain (June 2009)

18. Blackshaw, P.: The Global Village, Virtually Realized: Social networking engages users. In: Nielsen Online Customer Insight (May 2008)

19. Global Mobile Forecasts to 2013 Worldwide Market Analysis, Strategic Outlook & ed-Forecasts to 2013. In: Informa telecoms & media Reports/Forecasts (December 2008)