Telecommun Syst (2011) 48:301–316
DOI 10.1007/s11235-010-9345-8

Contextualized User-Centric Multimedia Delivery System for Next Generation Networks

Jose Simoes · Thomas Magedanz

Published online: 27 May 2010

Abstract Telecommunication and Internet services are constantly subject to changes, seeking the customer's full satisfaction. Enriching these services with innovative approaches such as context-aware, social, mobile, adaptable and interactive mechanisms enable users to experience a variety of personalized services seamlessly across different platforms and technologies. In this sense, 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 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.

Keywords Contextualization · Personalization · Multimedia Delivery · Personalization · User-Centric

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. Location and presence are examples of context based services widely deployed today. 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 ap-

J. Simoes (✉) · T. Magedanz
Fraunhofer FOKUS, Kaiserin-Augusta-Allee, 31, 10859 Berlin, Germany
e-mail: jose.simoes@fokus.fraunhofer.de

T. Magedanz
e-mail: thomas.magedanz@fokus.fraunhofer.de

plication 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 an enhanced version of the Context-aware Triggering System (CATS) [3], a user-centric multimedia delivery 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 that take users needs, desires and intentions in concern. 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 Sect. 2. Then, we present in Sect. 3 the proposed architecture, the components that allow ubiquitous user-centric context-aware service creation and introduce some of the application capable behaviors. A special focus over some use-cases, business models and concept validation is covered in Sect. 4. Finally, Sect. 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 [4]. 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 [5].

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 [6] 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 [7] 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. [8] 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. An asset management system based on Radio Frequency Identification (RFID), Web Geographic Information System (WebGIS) and Short Message Service (SMS) is specified in [9]. The proposed method tracks and monitors the changes of locations of assets and shows their statuses on a geographical map. The movement of assets is automatically identified using RFID techniques and corresponding data are transferred by a Global System for Mobile communications (GSM) module in real time. Furthermore, this method facilitates automated notification of asset movement through SMS. This work presents some similarities to what is presented in this paper, however in a very limited and static context.

In fact, notification mechanisms are getting more prominent, especially with the advent of social networking (communities), has many see it as the new trend in the landscape of communications. Social Internet portals such as Facebook [10] provide all sorts of notifications aiming at improving the overall Quality of Experience by the end user towards his friends. However, at this point in time, it is still lacking a method to personalize the way a user status is updated in what concerns privacy, location and other environment factors. In this sense, in recent research activities, user perceived QoE became a very hot topic, as it can be applied to mostly any service, measuring the overall user satisfaction. Assuming QoE is a consequence of a user's internal state (e.g., predispositions, expectations, needs, motivation, mood), the characteristics of the designed system (e.g., complexity, purpose, usability, functionality, relevance) and the context (or the environment) within which the interaction occurs (e.g., organizational/social setting, meaningfulness of the activity, voluntariness of use) [11]; its measurement and evaluation are still in their primary stages and consequently very difficult to perform. On the other hand, the principles behind its achievement are well known and documented in [11, 12]. Among the top ranked we can find personalization, contextualization, adaptation, interactivity, mobility, privacy, price and rewarding.

Correlating these facts with research done around social networks, it is easy to identify parallelisms between what social communities offer and what users are looking for. These studies have been performed in [13] and [14], where different aspects of social networks were analyzed and matched with the user requirements. Moreover, social networks adoption rate in the last few years [15] confirms this trend. In this sense, a lot of research from both academia and industry is being done around this topic. To overcome the discrepancy between online and "offline" networks, reality mining techniques can be empowered to approximate both worlds, proving awareness about people actual behavior. It typically analyzes sensor data (from mobiles, video cameras, satellites, etc.) to extract subtle patterns that help to predict and understand future human behavior. These behavioral patterns, in turn, allow the prediction of individual-level outcomes such as job satisfaction or understand broader concepts such as how do social networks evolve over time [16].

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 [17] 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 [18] which use IP Multimedia Subsystem (IMS) [19] 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 [20]. 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 and creation of user-centric context-aware services for next generation networks.

3 Architectural approach

The Context-aware User-Centric Multimedia Delivery System may be described as a multi-context multimedia delivery 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. 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 (e.g., weather, traffic, location).

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 methodology 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.

Doing a top-down approach will enable an easy understanding of the presented architecture. The **Service Layer** is comprised of, the enablers and the service broker/orchestrator, which host, build, manage and execute the services. The enablers are generic components with defined capabilities, implemented as services in SIP and/or Web Application Servers (ASs), to be reused as building blocks across multiple applications and services. They may be used to provide a service directly to the end user, or in conjunction with others. The service broker is an element that allows dynamic and semantic service discovery, exposure, orchestration and execution through the usage of policies. It is therefore responsible for the execution of service request constraints through the orchestration of several enablers.

The **Control Layer** is composed by databases and network control servers for managing session setup, modification and termination (when needed). The Home Subscriber Server (HSS) is a database, which serves as a central repository for user related information, including user identity, user preferences, among others. In the future the authors pretend that some of this information is stored into a Common Profile Storage (CPS), defined by the Third Generation Partnership Project (3GPP) in [21], a framework for streamlining service independent user data and storing it under a single logical structure in order to avoid duplications and data inconsistency. Being a logically centralized data storage, it can be mapped to physically distributed configurations and should allow data to be accessed in a standard format. As seen in Fig. 1, the HSS communicates with other entities in the control and application layers by using Diameter [22].

The **Connectivity Layer** includes routers and switches for the IP Core and access networks including 3GPP's General Packet Radio Service (GPRS) or Universal Mobile Telecommunications System (UMTS) Radio Access Network (RAN), 3GPP2's Code Division Multiple Access (CDMA) 2000 RAN, Wireless Local Area Networks (LANs) and various fixed Digital Subscriber Line (DSL) options. The User Equipment (UE), like PCs and mobile phones, connect to the network at this layer.

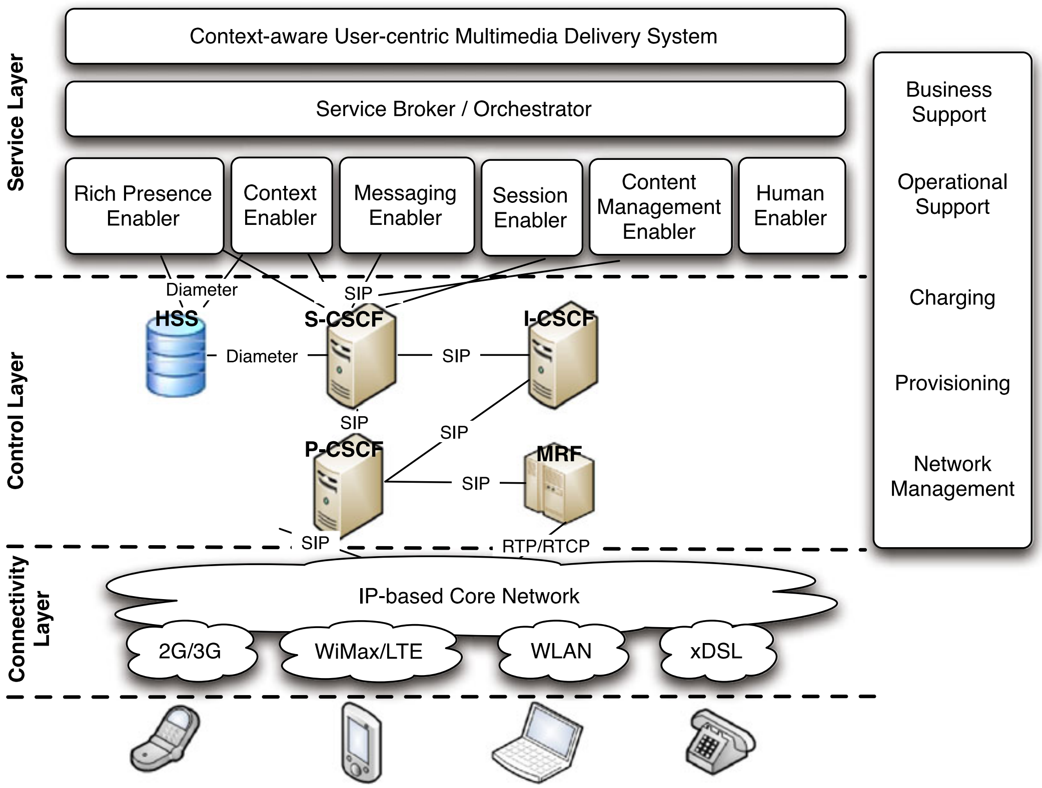

Fig. 1 General overview of the Context-aware User-Centric Multimedia Delivery System architecture

3.2 Functional architecture

In the previous section we introduced the main components, which interact with the multimedia delivery system; 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, focusing only the service layer, where our contributions are most notable. Once again, we will use a top-down approach to detail the specific functionalities of each entity.

Although all enablers interact directly with the application server, from Fig. 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, Human and Session enabler 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, namely service abstraction, discoverability, composability, autonomy, loose coupling of components, among others. One of these advantages is the possibility that different service providers implement different parts of the system. Despite both the Session Management, the Content Management and the Human enablers are not required in order for the system to work, the possibilities will be greater if they are available.

In the preceding scenario, we assume that the capabilities, ontologies, semantics and policies of the specified enablers were 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 Sect. 3.3. In order to understand the role of each enabler, this section 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 [23].

3.2.1 Context-aware User-Centric Multimedia Delivery System application

This component is mainly responsible for managing and representing the multimedia triggers defined by the end user.

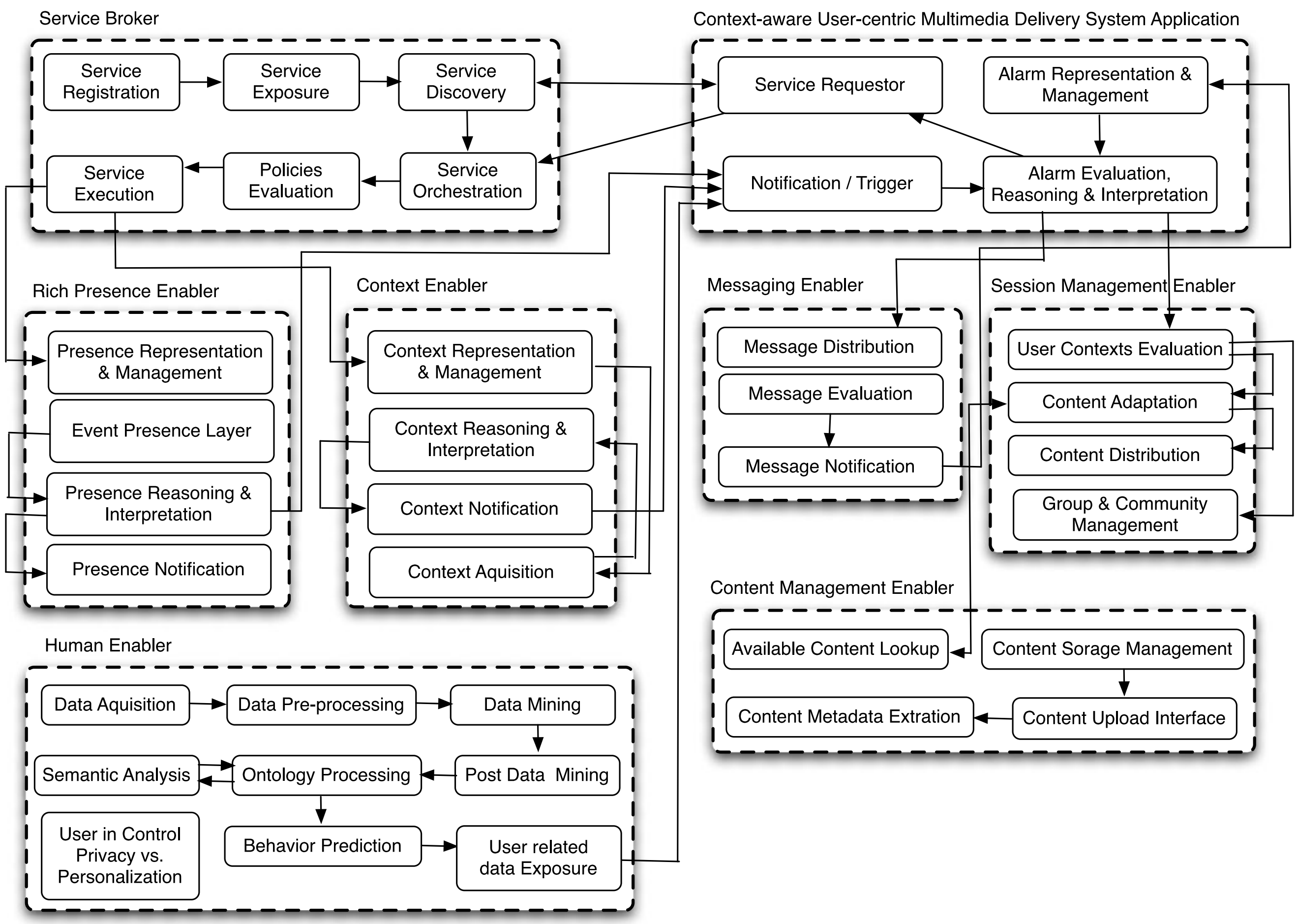

Fig. 2 Functional architecture of the Context-aware User-Centric Multimedia Delivery System

Upon a trigger creation, depending on its characteristics, it passes through an evaluation, reasoning and interpretation process. At this stage, the application realizes which capabilities are going to be used for that specific trigger. After, using the service requestor module, it interacts with the broker in order to realize which components can provide the required capabilities. Subsequently, it demands service execution based on the previously gathered information about an exposed service. Furthermore, when some presence or context information changes; it should be capable to deliver the notifications to the reasoning module in order to evaluate whether the trigger should be adapted to the new environment conditions. When the application logic decides to activate the trigger, according to a set of variables, namely user preferences, context information, user profile, presence information or other specified upon trigger creation, the targeted users are notified via the Messaging server or Session enabler (when required).

3.2.2 Service Broker/Orchestrator

The Service Broker provides a way for realizing rapid and low-cost system development and improving total system quality. It has the responsibility to coordinate 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. Service Discovery and Service Registration relate to required mechanisms for registering and discovering new services by using service registries. Service Exposure refers to the required mechanisms by which external users of this service-oriented and service-driven model discover and access network and other enablers functionalities. The next step, Service Orchestration is the mechanism by which several atomic services (service building-blocks) are orchestrated in order to realize a chain of processes (a composition of services), to either meet a predefined goal or to create new services. Optimally, however not mandatory, policy based service access controlling mechanisms govern the external invocation of services. After all requirements are fulfilled, the service is executed.

3.2.3 Human enabler

This enabler is based on a Generic Human Profile (GHP) and represents a set of properties build within a generic structure that allows the services of the future to use user related information, while respecting their privacy, needs and concerns. Opening the door for opportunistic communications, user context is disclosed according to contextual privacy policies and settings, enabling systems and devices to sense how, where and why information and content are being accessed and respond accordingly. In addition, by using semantic, ontology and keyword technologies that understand the meaning of information and facilitate the accessibility and interconnection of content, it is possible to generate/infer new types of knowledge that can relate to users' behaviors, needs or intentions. Together with the HSS and Context Enabler, by storing users external contexts, it is capable of comparing different sorts of data so far not correlated improving on the one hand the algorithms, but on the other hand the user overall satisfaction as services become more contextualized, adapted and consequently personalized. Moreover, it allows the integration of social data from different platforms, providing a unified way to access users' (Humans) friends' lists, among others, combining both online and offline social networks data.

3.2.4 Rich Presence enabler

The Presence Server offers the basic presence functionalities as well as location-based information. Due to its flexible and modular architecture, it can be easily extended and customized for other event packages and applications when desired and needed. The management module is responsible for managing all the user subscriptions and keeping them up to date. Every time a new notification comes, the Event Presence Layer passes this information to the Reasoning and Interpretation module which according to the defined settings can trigger a notification or not. This module is very important as it reasons whether the notification should be sent or not, preventing users or other application servers from being flooded with information they did not request.

3.2.5 Context enabler

Although in context-aware systems, domain knowledge is very much tied to the application, building context aware applications from scratch is not practical. Therefore, one of the aims of this component is to decouple 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 [23]. The Representation component is fundamental because the context information needs to be represented and modeled for being machine interpretable and exchangeable using well-defined interfaces. The goals are to support easy manipulation, easy extension, scalability, efficient search and query access of context information. The Reasoning and Interpretation module is responsible for deriving high-level contextual information from lower level context or raw data (e.g. databases or sensor output). This inference typically comprises several processing stages. Although context-aware content selection is not included in the application described in this paper, it will be added into this entity and addressed in future work. Lastly, the Notification module is responsible for delivering context information to the enablers/applications that subscribe to its changes.

3.2.6 Messaging enabler

This entity represents a next generation cross network and platform messaging system. This component has 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, by using the Message Distribution and Notification modules respectively. The evaluation module is where the application logic is comprehended. By using predefined codes, the messaging enabler can perform the notifications to the respective application server. Despite the flexibility of this component, the application is the entity responsible for choosing on which channel the message should be delivered: SMS, Email, Multimedia Message Service (MMS), SIP message or Extensible Messaging Presence Protocol (XMPP).

3.2.7 Session Management enabler

Session Management is the entity in the architecture that 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. Session management handles different types of events regarding session control, specifically: session establishment, session renegotiation (upon a given trigger or change), session termination and session mobility. Thus, it participates in dynamic changes such as switching between different content as it is closely inter-linked with media delivery. Moreover, it is responsible for User Context Evaluation, where according to the user device settings and network conditions it is capable of adapting the content, providing the user with a more compelling experience. The Group & Community Management interface is used to group users with the same context resources (device type and access networks) and enable the communication between them by using multicast delivery mechanisms when convenient. Other grouping approaches such as presence, preferences and behavior will be further explored in our future work under the Context enabler.

3.2.8 Content Management enabler

Content Management is the entity that is responsible for the content provisioning, storage and management. The content that is to be delivered to other users at some point in time is previously uploaded. Both user generated and professionally created content, namely advertisement content, can be managed by this component. When uploaded, the content may be tagged with metadata supplied by the user or content provider. Also, further content metadata extraction may be performed by the system. The uploaded content is ideally stored in dedicated storages and then made available to the other enablers through an interface that is able to perform lookups and searches.

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:

3.3.1 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 [24]. This is the approach used in our framework and its design principles are documented in [25]. All context-processing entities can either be Context Consumers (CxC), Context Providers (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.

3.3.2 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 infrastructure, 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 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 triggering decisions within the application, service or service enabler is a poor performing design.

3.3.3 Network view

From the network perspective, one key objective of our framework 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 CUMDA architecture enabling it are presented in detail in [26]. 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, we took these 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

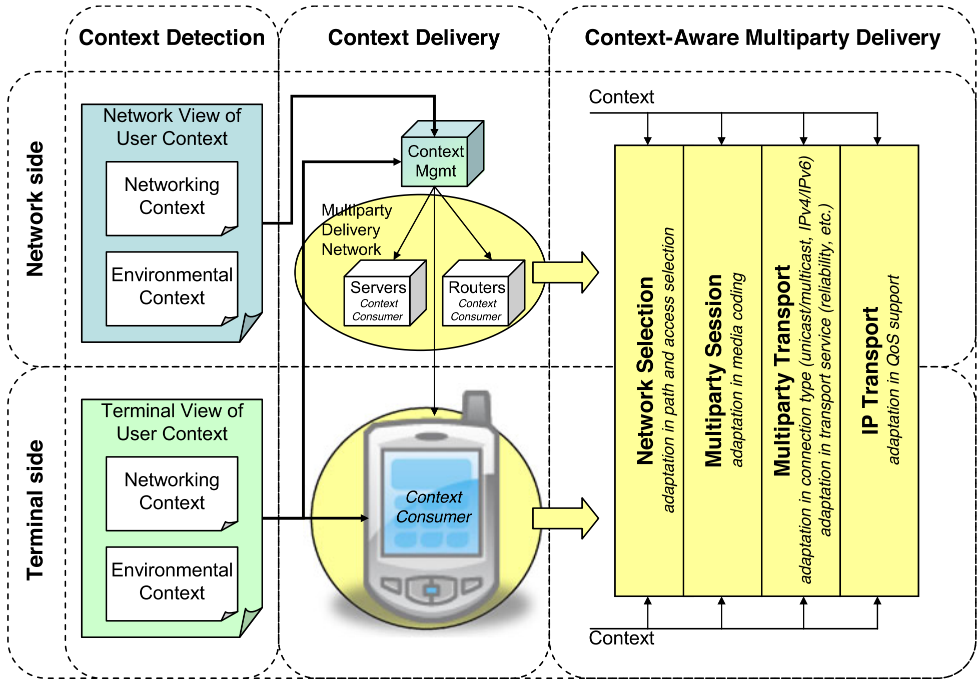

Fig. 3 Context-aware multiparty delivery

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.

3.3.4 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 triggers the consumption of the content, it will be delivered to the end-user terminals for final consumption, via multicast or unicast.

3.4 Application behavior

To better understand the sequence flow between different components, this section presents three basic scenarios: when the trigger is set, when the trigger is changed, and when the trigger is fired. The examples only describe a subset of the functionalities this architecture is capable of offering.

3.4.1 Setup a trigger

In the scenario depicted in Fig. 4, the user uses a webpage or a device proprietary application to setup a trigger. By starting the application (1) the user triggers a discovery mechanism on the Context-aware User-centric Multimedia Delivery Application (CUMDA) towards the broker (2). The broker then returns the current capabilities and descriptions on the existing services (3) and this information is later forwarded to the user in a human readable format (4). Based on the received information, the user selects the enablers and contexts he/she wants to use for his trigger (5). This information is processed at the application and resent in a computer readable format to the service broker (6). There, the user and service policies are cross checked to ensure the user is

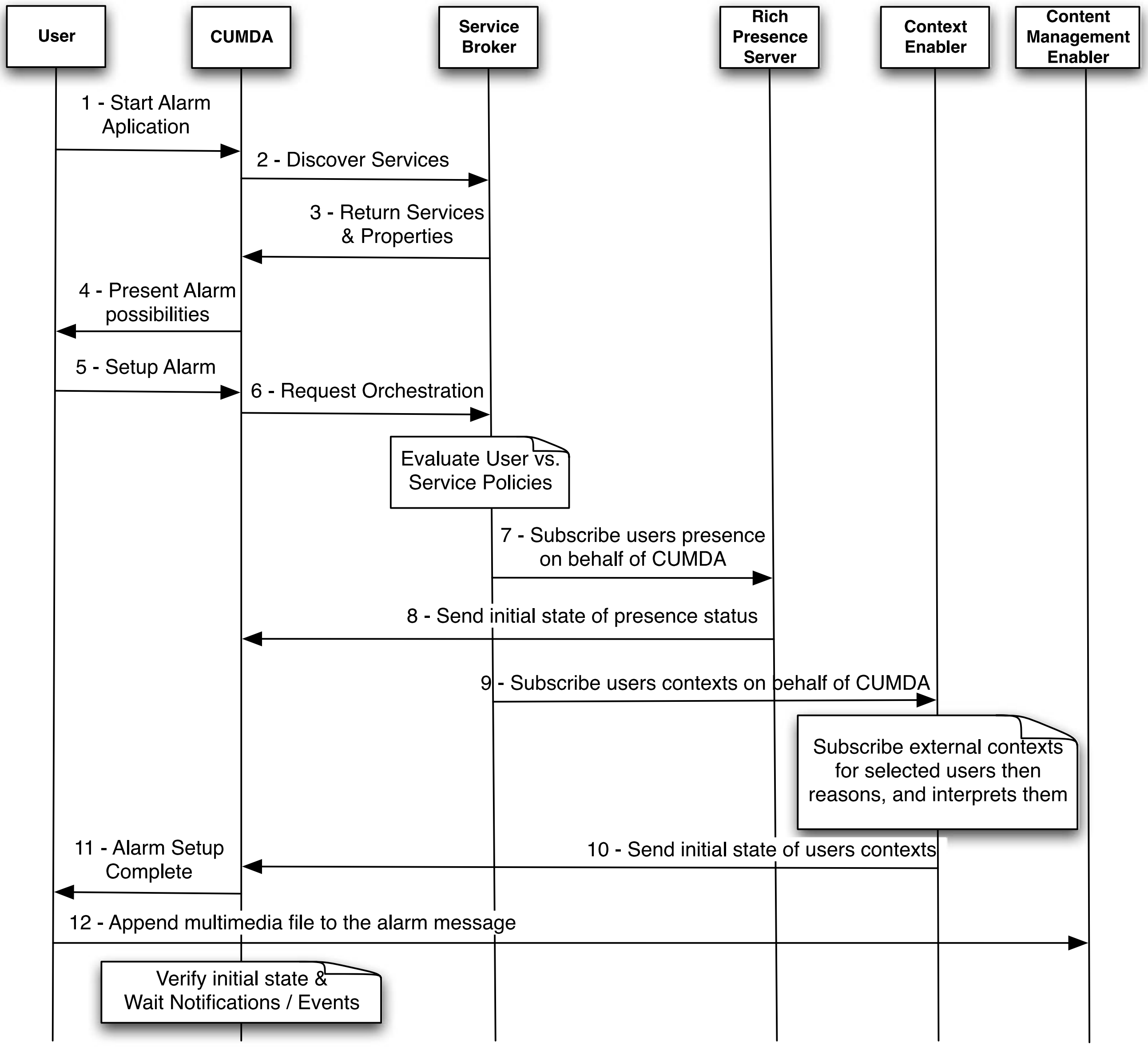

Fig. 4 Message flow between the components while setting up a trigger

allowed to use the orchestrated services. When no problems occur, the services are executed in the respective enablers. The first service to be executed is the subscription of the rich presence status of the selected users (7). Upon execution, the enabler sends the initial subscription status of the requested users to the application (8). Then the broker subscribes specific contexts of a particular user or a set of them on behalf of the CUMDA (9). Similarly to the other enabler, the context enabler sends the initial states of the requested contexts (10). After the trigger setup is completed (11), the user may use the Content Manager, if available, in order to upload a video or audio message to be delivered along with or alternatively sets up a plain trigger message (12).

3.4.2 Change a trigger state

This example assumes a trigger where the user had previously setup a reminder to all her sports friends living in a 20 km range not to forget the next beach volleyball practice. However, in order to avoid disappointment, she decides to enhance the trigger so that the event is canceled in case it's raining. As she is out of town with limited access to a computer, she sends the following SMS to the server "TRIGGER JSI569 ADD WEATHER RAINING, ACTION CHANGE TEXT 'E-vent Cancelled due to bad weather' ". Although the syntax is not yet clear, it exemplifies how this could work. Figure 5 illustrates the message flow between all intervenient. The scenario starts with the user sending an

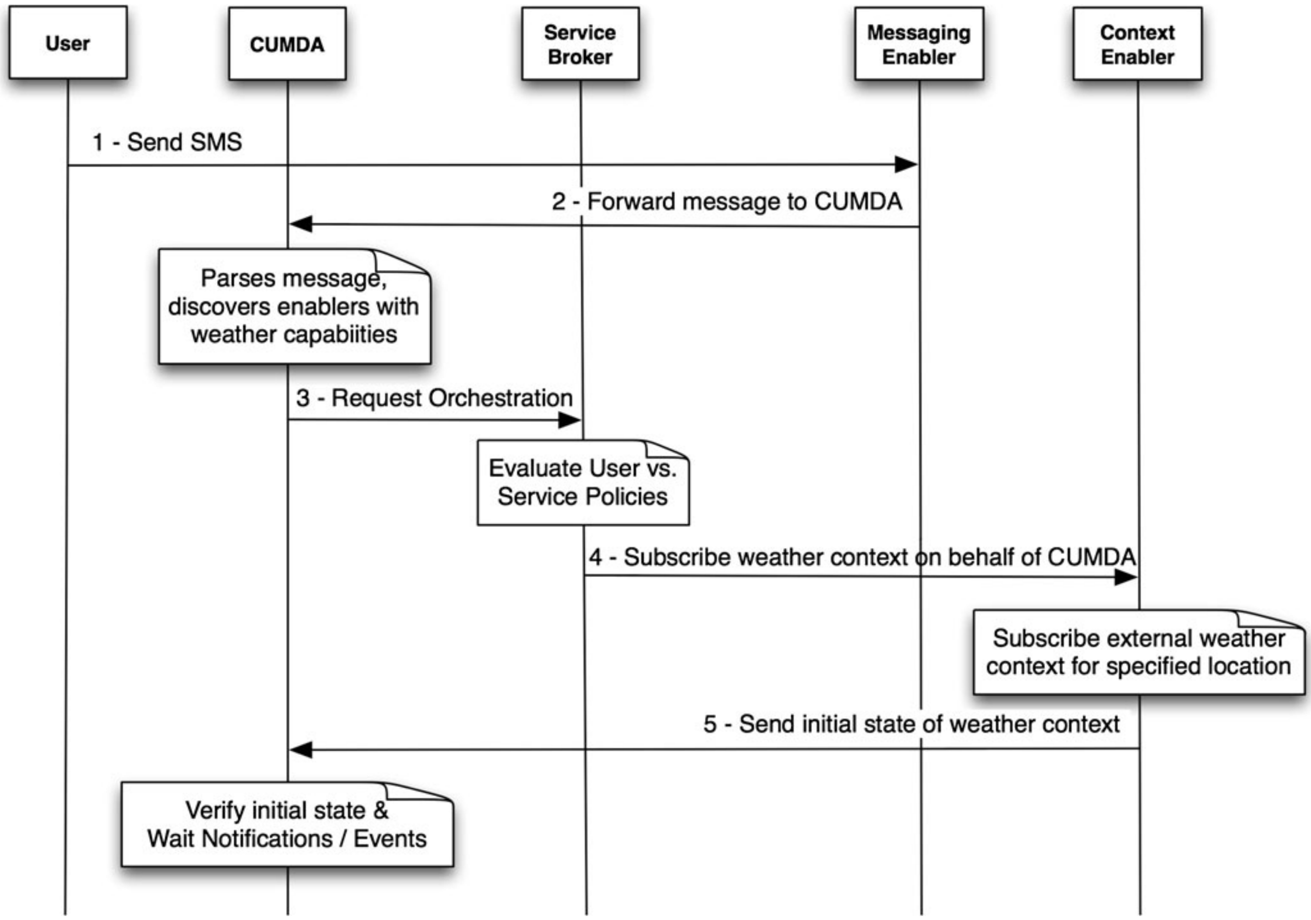

Fig. 5 Message flow between the components when the trigger is changes via SMS

SMS (1) to a specified number attached to the messaging enabler (ME). Upon reception, the ME forwards the message to the CUMDA (2) based on the first word of the message "TRIGGER". Based on its Evaluation, Reasoning and Interpretation module, similarly to what happen in the previous scenario (see Fig. 4), the CUMDA detects that it needs to acquire some weather conditions for a specific location and therefore tries to find an enabler that offers such capabilities. After receiving this information from the broker the CUMDA requests it to orchestrate and execute the service (3). After checking all the policies, the service is executed and the broker subscribes for weather conditions on behalf of the main application (4). As it is impossible for the context enabler to collect every type of context, it outsources/subscribes this information to third party entities. When it receives an update for the required location, this is then forwarded to the application in order to act accordingly (5). Subsequently, the system stays on hold, waiting for any notification or event.

3.4.3 The triggering

Following the previously defined scenarios redline, after a setup and change of a trigger, it is eventually activated. Figure 6 shows the message changes that occur when the defined trigger timer is fired.

As this trigger type was not supposed to be activated by any rich presence status (includes location) or weather update, it was waiting for the stipulated time to trigger itself. However, in the meantime, it has been receiving weather (1) and rich presence status (2) updates whenever changes occurred. When the trigger occurs, the internal procedures try to gather and evaluate all the context information. In this example, only two of the users were within the defined notification range. Based on their presence status, the CUMDA decided that "user A" should get an SMS (4a) because she was busy and "user B" should be notified via SIP message (5a) because she was online with her SIP client. These messages are preceded by the notification towards the messaging enabler (3a). Alternatively (3b, 4b, 5b), if at the time of the trigger definition the user defined a video to be sent, the Session Management Enabler would be used to deliver it,

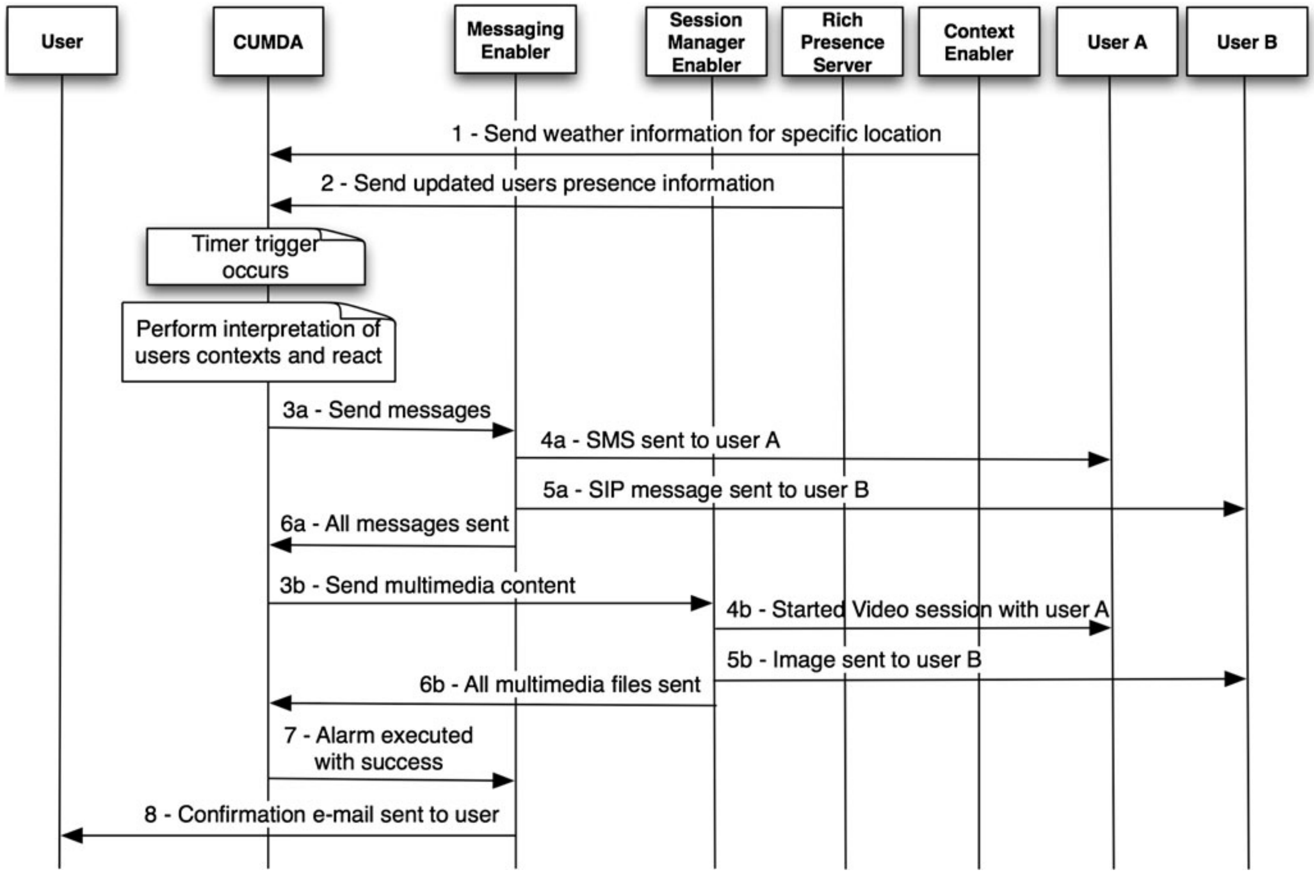

Fig. 6 Message flow between the components when the trigger is fired

instead of the Messaging Enabler. This is also dependent on the user context, as in some particular environments a less intrusive trigger, like a simple message, may be preferable even if a video is available. After all messages are successfully sent, the application is acknowledged (6). Lastly, according to the author presence status, the messaging server is informed (7) to notify the user (8) that the trigger was executed successfully. In case of problems, these should also be reported on the last message to the end user.

4 Validation

In order to corroborate the proposed architecture not only based on technical but also business aspects, this section presents two different use cases as well as introduces a prototype implementation of what can be achievable with such an architectural model.

4.1 Use case 1: notify me and warn my friends

For those who are familiar with long duration travels by train or bus certainly already experienced sleepiness. What if there was an application that could act as geographic trigger, wake you up near your destination and at the same time warn your friends or business colleagues that you are arriving. With such an architectural approach, this scenario can easily become a reality. Moreover, it would be possible to personalize the way each intervenient would be notified. Actually, this concept would allow users to build their own triggers based on intuitive graphical interfaces that represent logical interfaces towards a myriad of enablers. In Fig. 7, we assume that the user wants to use Location, Messaging, Presence, Time and User Preferences enablers. The trigger composition starts with the user selecting that only her girl friends between 18 and 40 will be targeted. Passing to Location, the user is able to specify a geographic point and a radius where the service will be provided (inside or outside the area), which corresponds to the desired destination of the transportation. After that, she selects which presence status should be considered for the service she wants to deploy. Then, the map is updated with all her contacts that are within the specified area with the selected presence status. Subsequently, the user defines the criteria to send messages. According to this decision, the system automatically detects what is the best way to contact each of her friends. At last,

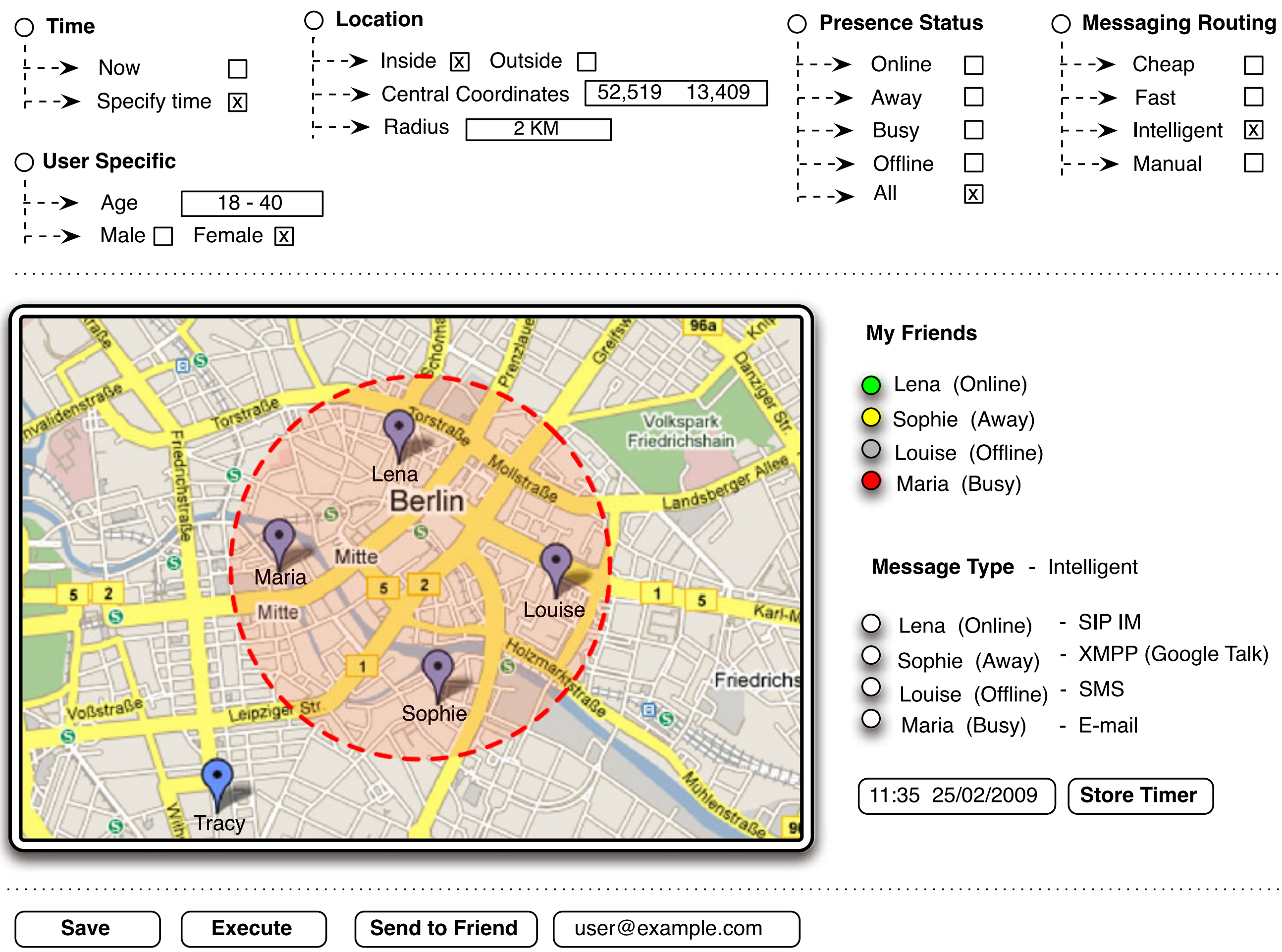

Fig. 7 Example of new trigger setup graphical interface

the service can be executed or programmed to run at another time. Before being executed, the trigger can be saved as a template either for future consumption or sharing with friends.

This approach brings new business models that can arise from proactive actions initiated or event triggered by the user itself, changing the established paradigm of a user, from service consumer to a service producer. Moreover, besides offering what most users seek in services nowadays: personalization, contextualization, mobility, interactivity, adaptation and at the same time usage control. Moreover, it could help to increase service operators Average Revenue per User (ARPU) as this application would take advantage of network infrastructures and other operational systems, e.g., SMS platforms, Global System for Mobile Communications (GSM) or Third Generation (3G) networks. Furthermore, it would allow service providers to better understand their customers and their buddies, increasing the value of their profiled data, which could be used for future services/environments such as social applications. Another possibility would be sponsoring the trigger service through advertising. Based on a cross reference between the multimedia content to be delivered (text, video, voice, etc.), the user preferences and profiled information, the content could be extended with relevant advertising information. In this case, the system could for example extend the content of the message to include a special discount, to all intervenient, in a coffee shop near the arriving point. This would however require that content other than text to be tagged semantically so that the reasoning and interpreter module inside the context enabler and CUMDA related information could be correlated. This will therefore be addressed in future work where we will improve content to context matching.

4.2 Use case 2: targeted advertising

The proposed architecture can also be used to address advertising scenarios, where 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

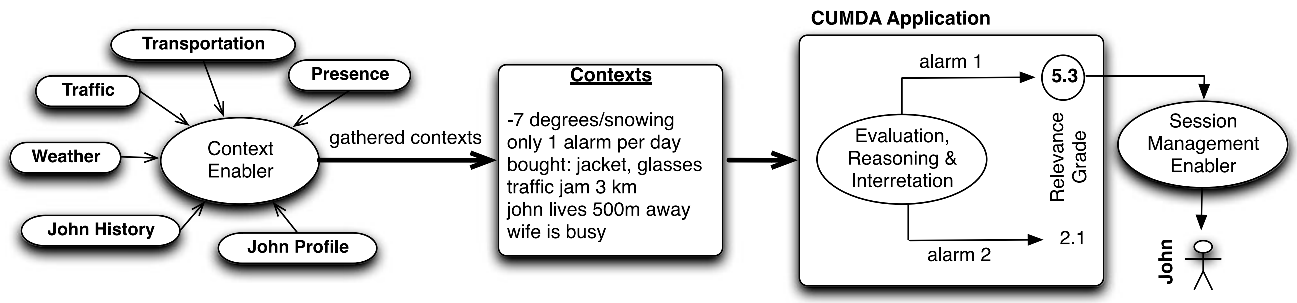

Fig. 8 Overview for two different triggers when using advertising

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 CUMDA, 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) [27], widely available inside these environments.

The following scenario presents users in proximity of certain sell-points with information about the goods that are available on special 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 activated 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. As the messaging flows between components were explained in Sect. 3.4, we will focus on the example. Supposing that different shops had their triggers and respective content previously setup under approximately the same location, the airport. At this point in time, several contexts associated with the defined triggers are proactively being collected by the context enabler. At some point in time, user John activates (due to location) simultaneously two different triggers. When this happens, both triggers get a relevancy grade and the higher one is the one that will be finally targeted to John. Figure 8 gives an overview illustration of this process. 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.

At this point in time, it is still not possible to correlate both contexts and content itself, but as explained before, this will be addressed in future work, improving the way the relevancy grade is determined, as at this point in time it only takes the store information into consideration, and not the content of the advertising product itself. In this particular case, "trigger 1" had been set by an accessories shop and "trigger 2" by a wine shop. During the inference process we admit that the reasoning algorithm had the words "umbrella" and "cap" under the shop associated metadata. Gathering all contexts, it seems to be more relevant when we consider that it was snowing and due to traffic conditions the best way to go home would be walking. As his wife had her presence status set to busy, he would have to walk home alone. Moreover, his profile indicates he has bought some clothes recently, which means he will eventually be interested in buying some accessories. This resulted into a 5.3 relevance grade. On the other hand, the correlation between wine products and the existing contexts was low. Consequently the relevance grade was only 2.1.

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, John uses his coupon to get a 20% discount on a brand new umbrella. Later on, when the coupon number is inserted in the system, John'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.

4.3 CUMDA concept validation

In order to test the concept proposed on this paper, a prototype of the Context-aware User-Centric Multimedia Delivery System was created. The application was developed using JAVA and deployed on a BEA Weblogic SIP Server [28], 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 sever, therefore, the discovery process was bypassed in our trials and the service orchestration hard coded on the CUMDA to enable service execution between all the enablers. The validation scenario was setup under FOKUS Open SOA Telco Playground [29] where the following components where used: OpenIMSCore [30], FOKUS Presence Server [31] and

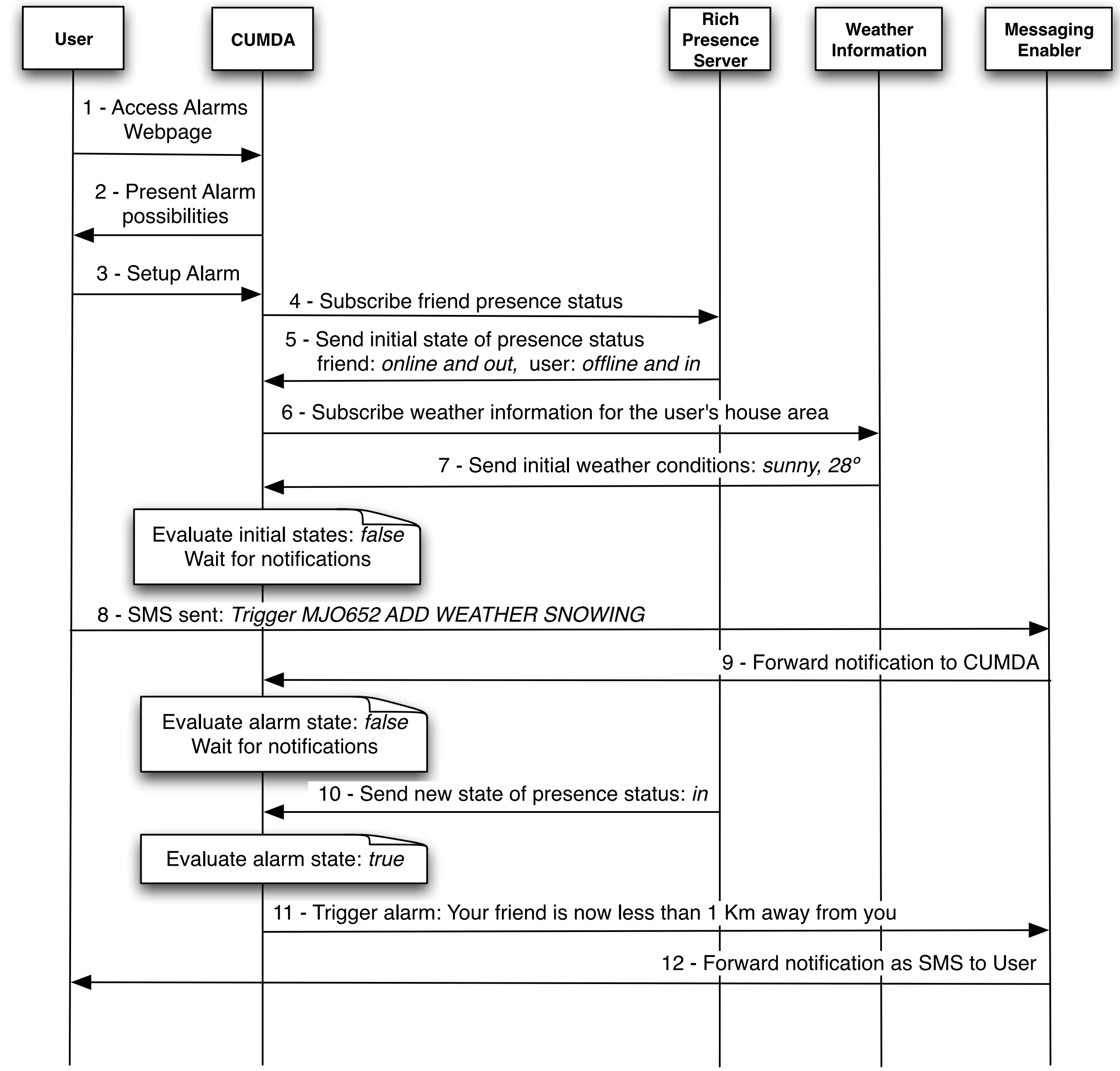

Fig. 9 Message flow between the components for the concept validation scenario

Converged Open Messaging Server (COMS) [32]. During the experiment we used the Telecom Italia Labs Context Broker [1] as context enabler and interconnected it with the CUMDA, which was responsible for requesting/subscribing the contexts. Session Management Enabler and Content Management Enabler were minimally tested at this stage (managing multimedia sessions with minimal multimedia adaptations—device capabilities). 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 1 km 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 with a message: via SIP message when online and via SMS otherwise. Moreover, we enabled trigger modification via SMS, like explained in Sect. 3.4, 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 is notified via SMS that his friend is in the surroundings. Figure 9 details the interactions among all the components involved in the concept validation scenario.

Using such a testing environment enabled us to acknowledge 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, the architecture needs to be further validated.

5 Conclusion 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, CUMDA architecture enables users to design complex and personalized context-aware user-centric multimedia 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 multimedia 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 CUMDA. 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 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 further explored under the human and context enablers; 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 [33, 34], 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 lists, policies and preferences, already existent and well developed in the context of social networks and telecom environments, generalizing it for the advertising world. 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, improving at the end the users perceived Quality of Experience by addressing topics such as personalization, contextualization, adaptation, interactivity and mobility.

Acknowledgements This work was supported by the European ICT Project—C-CAST—Content Casting (Contract-No. ICT-216462) Furthermore, the authors would like to thank all the project partners involved in C-CAST for their insightful comments and ideas.

References

1. Baker, N., Zafar, M., Moltchanov, B., & Knappmeyer, M. (2009). Context-Aware Systems and Implications for Future Internet. In *Future Internet conference and technical workshops*, Prague, Czech Republic, May.
2. Camarillo, G., & Garcia-Martin, M. (2004). *The 3G IP multimedia subsystem: merging the Internet and the cellular worlds*. New York: Wiley.
3. Simoes, J., Goncalves, J., Mota, T., & Magedanz, T. (2009). *CATS: Context-aware triggering system for Next Generation Networks, wireless and mobile networking* (Vol. 308/2009, pp. 251–262). Boston: Springer.
4. Google Mail Privacy Policies (2009). Targeted Ads in Google Mail. http://mail.google.com/mail/help/intl/en_GB/about_privacy.html.
5. Aquity Mobile (2009). http://www.acuitymobile.com/solutions/spot_relevance.php. (last accessed, Nov. 2009).
6. Kaasinen, E. (2003). User needs for location-aware mobile services. In *Personal and ubiquitous computing*. Berlin: Springer.
7. Ahn, Y. (2008). Design of a mobile object data management framework for location enhanced applications. In *International conference on convergence and hybrid information technology* (pp. 270–273).
8. Kukhun, D., Soukkarieh, B., Lopes-Ornelas, E., & Sedes, F. (2008). LA-GPS: A location-aware geographical pervasive system. In *24th IEEE international conference on data engineering workshop*, Cancun, Mexico (pp. 160–163).
9. Meng, S., Chen, W., Liu, G., Wang, S., & Wenyin, L. (2008). An asset management system based on RFID, WebGIS and SMS. In *Proceedings of the 2nd international conference on Ubiquitous*

information management and communication (pp. 82–86). New York: ACM Press.
10. Facebook (2010). www.facebook.com.
11. Hassenzahl, M., & Tractinsky, N. (2006). User experience—a research agenda. *Behaviour & Information Technology*, *25*(2), 91–97.
12. Kelkki, K. (2008). Quality of experience in communications ecosystem. *Journal of Universal Computer Science*, *14*(5), 615–624.
13. Jamali, M., & Abolhassani, H. (2006). Different aspects of social network analysis. In *Proceedings of the IEEE/WIC/ACM international conference on Web intelligence*, Dec. 2006 (pp. 66–72).
14. Sorensen, L., & Skouby, K. (2008). Next generation social networks—elicitation of user requirements. In *IEEE 19th international symposium on personal, indoor and mobile radio communications*, Sep. 2008 (pp. 1–5).
15. Owyang, J., Bernoff, J., Pflaum, C., & Bowen, E. (2009). *The future of the social Web* (Forrester Report). April.
16. Eagle, N., Pentland, P., & Lazer, D. (2009). Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences of the United States of America*, *106*(36), 15274–15278.
17. Shujun, D., Liang, L., & Chengqi, C. (2008). Research on Geographical Information Service Based on SOA. In *Proceedings of the IEEE international conference on automation and logistics*, Qingdao, China.
18. Brajdic, A., Lapcevic, O., & Matijasevic, M. (2008). Service composition in IMS: a location based service example. In *3rd international symposium on wireless pervasive computing*, IEEE, Santorini, Greece (pp. 208–212).
19. 3GPP TS 23.228 (2008). Service requirements for the Internet Protocol (IP) multimedia core network subsystem (IMS); stage 1. Release 9, 2008-12-19.
20. Blum, N., Magedanz, T., Schreiner, F., & Wahle, S. (2009). From IMS management to SOA based NGN. *Journal of Network and Systems Management*, *17*, 33–52.
21. 3GPP TR.32.808 (2008). 3rd generation partnership project: technical specification group services and system aspects; telecommunication management; study of common profile storage (CPS) framework for user data for network services and management (Release 8).
22. Calhoun, P., Loughney, J., Guttman, E., Zorn, G., & Arkko, J. (2003). Diameter base protocol. IETF RFC 3588.
23. Requirements and concepts for context detection and context-aware multiparty transport. (2008). In *Deliverable 6-C-CAST: content casting, European project* ICT-2007-216462.
24. Chen, H. (2004). *An intelligent broker architecture for pervasive context-aware systems*. PhD Thesis, University of Maryland, Baltimore County.
25. Goncalves, J., Moltchanov, B. et al. (2009). Context management architecture for future Internet services. In *ICT mobile summit* 2009, Spain.
26. Antoniou, J., Simoes, J. et al. (2009). Context-aware multiparty networking. In *ICT mobile summit* 2009, Spain.
27. Brickley, D., & Miller, L. (2007). Friend of a friend, FOAF vocabulary specification 0.91. http://xmlns.com/foaf/spec/.
28. BEA WebLogicŠIP Server (2010). Converging Internet, SOA and next-generation telecom services. www.bea.com/sip.
29. FOKUS Open SOA Telco Playground (2010). http://www.opensoaplayground.org.
30. Open IMS Playground (2010). www.open-ims.org.
31. FOKUS Presence Server (2010). www.open-ims.org/presence.
32. FOKUS Converged Open Messaging Server (2010). www.open-ims.org/coms.
33. Blackshaw, P. (2008). The Global Village, Virtually Realized: Social networking engages users. In *Nielsen online customer insight*, May.
34. Global Mobile Forecasts to 2013 Worldwide Market Analysis, Strategic Outlook & edForecasts to 2013. In *Informa telecoms & media reports/forecasts*, December 2008.

Jose Simoes has graduated with distinction in Telecommunications and Computer Engineering in 2006 at Institute Superior das Ciencias do Trabalho e da Empresa (ISCTE), Portugal. He also studied at Universidade Federal de Santa Catarina in Brazil. In the past he has worked for Netcall (1st VoIP Provider in Portugal) as Network Engineer and Business Consultant. Currently he is a Ph.D. student at Technical University of Berlin and a research fellow at Fraunhofer Institute FOKUS in the competence center for next generation network infrastructures—NGNI. His current interests are User Quality of Experience (QoE) applied to next generation services (personalization, contextualization, adaptation, mobility and interactivity) and advertising, focusing on architectures for next generation heterogeneous networks. His work research also includes social networks, reality mining and context-aware technologies.

Thomas Magedanz (Ph.D.) is professor in the electrical engineering and computer sciences faculty at the Technische Universitat Berlin, Germany. In addition, he is director of the “next generation network infrastructure” division of the Fraunhofer Institute FOKUS, which provides various testbeds and tools in the context of converging networks and open service delivery platforms. Since more than 20 years Prof. Magedanz is working in the convergence field of fixed and mobile telecommunications, the internet and information technologies. Under his leadership many service development platforms, toolkits and testbeds have been developed, such as the Grasshopper Mobile Agent platform, the OSA/Parlay Playground, the Open Source IMS Core System, and most recently the Open SOA Telco Playground. In the course of his research activities he published more than 200 technical papers/articles. In addition, Prof Magedanz is senior member of the IEEE, and editorial board member of several journals.