Towards Service Capability Interaction Management in IMS Networks

Mosiuoa Tsietsi, Alfredo Terzoli and George Wells
Department of Computer Science
Rhodes University, P.O. Box 94, Grahamstown 6140
Tel: +27 (046) 603 8291, Fax: +27 (046) 636 1915
m.tsietsi@rucus.ru.ac.za
{a.terzoli, g.wells}@ru.ac.za

***Abstract*—For a long time, IP Multimedia Subsystem (IMS) services have been developed and deployed in a manner that does not carefully consider how these services can interact with each other in a beneficial way. Services can often overlap in terms of the logic that they execute, causing pockets of redundant code across several application servers. Recently, a more effective strategy is being employed of breaking down services into the basic building blocks needed to construct them known as service capabilities. This decomposition of services has made it necessary for service brokering and interaction management roles to be performed. A service broker must broker for third party application services wishing to access service capabilities while an interaction manager must resolve the interactions that need to occur between service capabilities in order to deliver a complete service. The challenge facing many IMS operators is that the standards are either incomplete or non-existent for performing these roles at this moment in time. In this paper, we focus solely on interaction management, what it entails, and propose a candidate architecture that extends the current IMS services framework. The design requires no modification of the current IMS protocols and procedures, and where extensions are proposed, existing IMS protocols and standard interfaces are reused.** [1]

***Index Terms*—3GPP, IMS, Service Broker, SCIM**

bibliographyreferences

I. Introduction

THE IP Multimedia Subsystem (IMS) is an all-IP architecture that is an integral part of the standards towards Next Generation Networks (NGNs). As a middleware component of an NGN, IMS sits between the access network layer below and the services layer above. The services plane has generated a considerable amount of interest as it is where the multimedia services that customers purchase or pay regular subscriptions for are hosted. Proponents for IMS technologies have gone to great lengths to develop and market certain services targeted for IMS as *killer applications* that will help to expand the customer base of operators' networks and curb plummeting revenue figures. Many of these attempts have thus far failed to solicit the excitement that was anticipated from the market.

Admittedly, it is difficult to imagine that there is a unique service that can be delivered only over an IMS network. Consider for example the Open Mobile Alliance (OMA). The OMA has an agreement with IMS Standards Development Organisations (SDOs) that one of its objectives will be to generate requirements for IMS [1]. The OMA produces specifications for interoperable data services for mobile devices. These deliverables are known as service enablers. Its specifications on service enablers for presence, messaging, push-to-talk over cellular (PoC) and conferencing have become de-facto standards in the mobile services industry and in all the OMA has published over 100 service enabler specifications. There is an effort to define ways in which OMA service enablers can co-exist with IMS, such as has been specified in [2], however none of their specifications explicitly require IMS to be present for correct or reliable communications between the customer's mobile device and the service enabler environment.

Given that there are no individual services that are likely to influence the market substantially, operators have begun to think differently about how they conceptualise services altogether. One way in which they have started doing this is in the shift towards enabler-based platforms. Service enablers as specified by the OMA are referred to as service capabilities by IMS and NGN standards bodies such as the 3rd Generation Partnership Project (3GPP) and ETSI (European Telecommunications Standards Institute) TISPAN (Telecoms and Internet converged Services and Protocols for Advanced Network). Service capabilities benefit the telecommunications domain in the same way that computer science benefits from object orientation in which functionality that is provisioned in the form of a service capability can be reused for different services, both current and future.

OMA realised that there was a need to develop a unified and consistent architecture for enablers to exist and be managed. This resulted in the development of the OMA Service Environment (OSE) which addresses the problem of enabler reuse as well as other issues such as the protection of enablers from unauthorised use and providing standard interfaces for 3rd party access [3]. The 3GPP has also identified a similar need for the management of services and published the first version of a technical report in 2008 that focused on the potential impact of introducing a management agent called a service broker into the IMS [4]. The goal was to describe how this agent would help manage and coordinate service interactions. According to the findings, the service broker would be located

[1]The work reported in this paper was undertaken in the Telkom Centre of Excellence in Distributed Multimedia at Rhodes University, with financial support from Telkom, Comverse, Tellabs, Stortech, Amatole Telecom Services, Bright Ideas 39 and THRIP.

between the IMS call session control functions and the application servers, and would be responsible for managing the execution of services by using a service interaction model. The study concluded with a number of suggestions regarding possible architectural placements of the service broker as well as providing the basic outline of an interaction model. However, at the time of writing, there is no standard within the 3GPP for a service broker.

In the absence of a clear directive, there has been some confusion in the research community between the terms service broker and service capability interaction manager (SCIM). In [5], [6] for example, these terms are used interchangeably, whereas in [7], [8], the SCIM is regarded as a sub-component of a service broker. In the latter approach, other functions such as policy evaluation, policy enforcement and workflow management are added to the function of a service broker in addition to the SCIM. It is the opinion of the authors that this is the most prudent path towards managing enabler-based environments and we choose to distinguish the service broker from the SCIM.

In this paper, we are interested in exploring service capability interaction management and in particular, its supporting infrastructure in the IMS. A black-box approach is applied with regards to interaction management and we focus rather on the mechanisms and protocols that need to be present between the SCIM and the service capabilities in order for effective interaction management to be realised. Though we are basing our approach largely on concepts derived from the 3GPP as the standards body that introduced the idea of a SCIM to the IMS world, the design still applies to non-3GPP IMS standards, since they employ identical schemes for the IMS services layer.

II. Background

A. IMS Services Architecture

Figure 1 shows the basic design of the IMS core network services layer. A customer in possession of user equipment (UE) requests a service from the IMS and the request is routed to the S-CSCF. If it has not already done so, the S-CSCF downloads the user profile from the Home Subscriber Server (HSS) and examines the initial filter criteria (iFC) contained therein against the request. The iFC indexes static rules that dictate which application server (AS) to route the request to depending on the properties of the SIP request and assign priorities to the targeted AS. The iFC are stepped through sequentially until the last rule is executed or an AS sends a response to terminate the session [9].

In [10], the SCIM is specified as an optional node which, when present, is co-located with a SIP AS and performs the function of interaction management. By interaction management we mean the ability to resolve the interactions between multiple service units that are required to fully deliver on an initiated request [11]. For example, let us consider a multimedia telephony service that uses call barring, call forwarding and voicemail components. The SCIM will need to be able to prioritise the execution of these components in an appropriate manner. For example, prioritising voicemail above others will

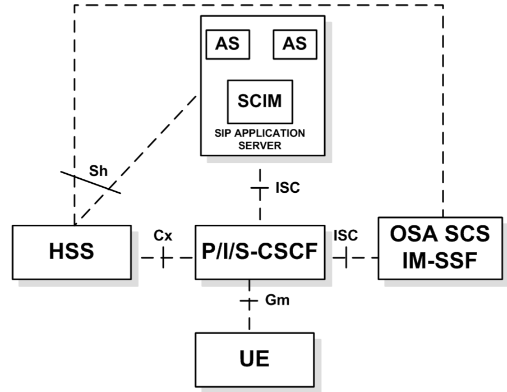

Figure 1. The IMS services architecture.

just cause all calls to be sent to voicemail and the customer would never receive any of their important calls. It would also be desirable for it to be able to handle the chaining of components in a dynamic way. For example, say that the barring component authorises a call from a non-barred caller address. If the forwarding component is activated, as might occur if the customer is away on holiday, but is unable to contact a configured forwarding address, this could call for a barring action to be taken. The logic would be that if the customer is away on holiday and at some point no longer wants to receive calls, all attempts to reach them should be barred. An additional condition could be that if this type of barring occurs at a time of the day at which the customer would normally be in their office, the call should be passed to voicemail. A SCIM would be expected to coordinate the invocation of each of these components and to factor in non-SIP information such as results obtained at runtime, calendar events or time of day.

Though the SCIM has been defined since 2002 (3GPP Release 5), the 3GPP left the functional architecture of the SCIM unspecified for a number of years. The issue of the SCIM was only revisited in 2008 when a study was initiated whose purpose was to investigate the potential impact of a SCIM on the current IMS services architecture [4]. The study found that the current IMS services infrastructure was insufficient for handling the types of interactions such as has been described in the multimedia telephony service example just given. There were several reasons that were found for this:

1) The current static invocation mechanism of the IMS which is based on iFC does not allow non-SIP features such as time of day to affect the triggering of services
2) The interleaving of services using dynamic rules such as conditions based on runtime results cannot be used to affect the triggering of services
3) The personalisation of service chaining by the customer, which can potentially be a very attractive service offering, is not possible

B. Standards and Service Building Blocks

IMS standards development organisations have taken similar stances on the standardisation of services in IMS. Instead of standardising whole services, they have opted instead to standardise the framework needed to develop those services. ETSI TISPAN take the view that a service comprises one or more service applications. A service application implements the functionality of a specific communications proposition while service capabilities are recognised as the reusable constituents of a service application [12]. The benefit here is that by standardising service capabilities rather than service applications, it is easier to provide service-level interworking since the capabilities used to express them are standard. The relationship between services and service capabilities are represented in Figure 2.

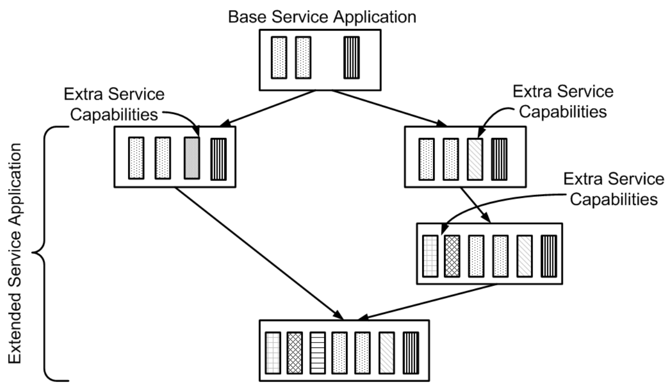

Figure 2. Services and service capabilities

The 3GPP adds more sophistication to this framework by distinguishing between service capabilities and service capability features. What are known in ETSI TISPAN as service capabilities are called service capability features in 3GPP. They represent the building blocks of a service application and are accessible over a standard interface. Two types of service capability features exist, framework service capability features and non-framework service capability features. The latter enable applications to make use of the functionality of the underlying network capabilities. Examples include location, data download, user status and user profile management. The former provides commonly used utilities that are necessary for the non-framework service capability features to be accessible, secure, resilient and manageable [13]. Examples include user-network authentication, application-network authorisation, registration and discovery. Service capabilities on the other hand are viewed as the bearer services that provide the capability for the transfer of information between access points and involve only low layer functions. Framework service capability features provide commonly used utilities that non-framework features need and are accessible in a way that is independent of any type of service. Figure 3 summarises this architecture.

C. SCIM Designs

Since the introduction of the SCIM, IMS researchers have been taking advantage of the lack of SCIM specifications to propose their own interaction models. In [14] the authors show how a multimedia gaming platform can be designed using messaging, conferencing and presence service capabilities. In [15] an interaction model is proposed that represents services as a combination of core and auxiliary service capabilities, with the core capabilities providing the major functionality and auxiliary performing other non-essential functions. In [16] describes an invocation mechanism that allows services to be invoked consecutively without signaling being transferred forward and backward to lessen the load on the S-CSCF. These designs rely on only slight modifications to the IMS service architecture and do not cater for user preferences or personalisation, which should be one of the main aims of the SCIM.

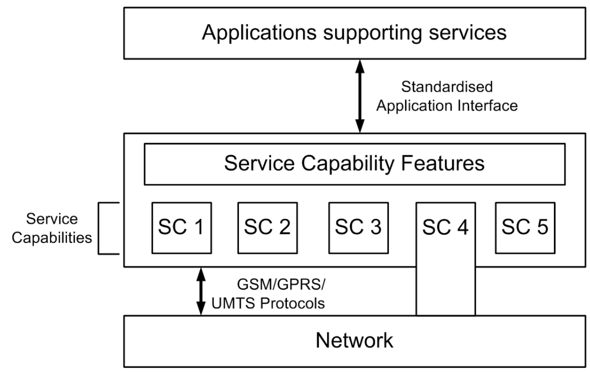

Figure 3. Service capabilities, service capability features and applications

III. Discussion

It is evident from the material related thus far that IMS standards are not yet up to date with the requirements of advanced service interaction management. The study conducted by the 3GPP uncovered many unresolved problems. In developing a conducive environment for a SCIM, a major priority would be to minimise the effect that its introduction would have on the services architecture. This would mean that though there are limitations in the static chaining of services through iFC, the SCIM would ideally be able to work with iFC and use them to achieve dynamic service chaining.

If a SCIM is to compose services based on service capabilities, it must be aware of those service capabilities. Registering service capabilities could provide details about their location, properties, protocol extensions and application interfaces so that the SCIM could maintain a repository for them. With access to this information, it would be possible for a network administrator to provision sets of relationships between capabilities, including default prioritisation values that would help specify how service applications are composed. Conditions could also be provisioned that would identify classes of runtime values that can be generated by the capabilities and how the SCIM should respond in terms of dynamic rules for handling those cases.

If this repository is well maintained, it would also be possible to allow customers themselves to discover network services from their UE and to customise the way in which capabilities are invoked to deliver a service to them. In so doing, the customer would not only be able to learn about service offerings but would also be able to apply their own preferences. In the next section we introduce the design of the network architecture that we are proposing for this purpose.

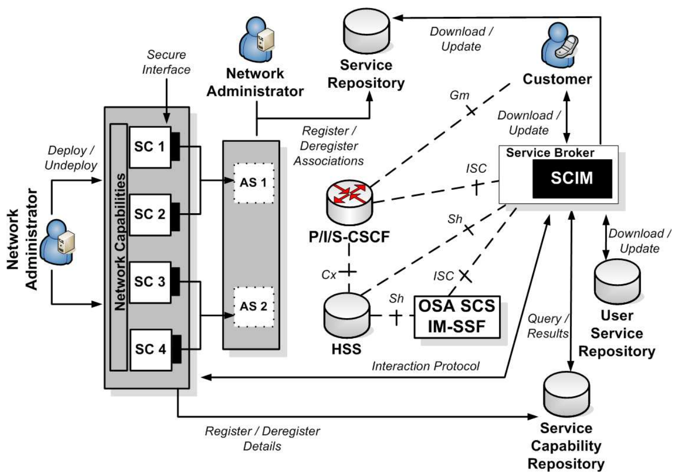

Figure 4. Proposed Design

IV. Design

Figure 4 illustrates the envisioned service environment for a SCIM. The broken lines represent standard IMS interfaces whilst the full line patterns represent proposed protocols. When comparing it with Figure 1 several extensions are evident. Firstly, there are no explicit application servers (AS), and AS1 and AS2 in the diagram are virtual services that are composed of underlying service capabilities (SC). It is anticipated that network administrators would be responsible for the deployment of these service capabilities as well as the creation of mappings that map services to actual service capabilities.

A. Management of Service Capabilities

Network administrators are responsible for the management of service capabilities in the network and to ensure that they are properly deployed, and are successfully undeployed. When a service capability is deployed, an entry must be inserted for it in the service capability repository (see section IV-C). The insertion of records into the repository can be manual, in which case they are entered by the administrator using out of band mechanisms such as a webpage or command line terminal. Alternatively, records can be published by each service capability instance such that when it is activated, it uploads its record to the service capability repository and by so doing, advertises its presence. This process of automatic upload would be aided by a communications protocol that would carry the appropriate details in the request. The nature of that protocol will be dependent on the implementation of the service capability repository, but two candidate protocols are discussed in section IV-C.

B. Service Capability Features

In [4] the 3GPP anticipated that the IMS Service Control (ISC) interface which is based on SIP would be used between the SCIM and the AS it interacts with. In our architecture, we recognise that service capabilities may not necessarily be implemented using SIP. As a result, the ISC is not used but other protocols that are decided by the specific implementation of the service capability. In section II-B we showed how service capability features acting as application interfaces are normally exposed to the rest of the network and through these access is allowed to the service capabilities. The details regarding the nature of these interfaces must be carried along with other information when the service capability is registered in the service capability repository.

C. Service Capability Repository

An important design goal for the service capability repository is to ensure that the implementation of the repository is not tied to any specific protocols or mechanisms but that a generic specification is provided for how the repository maintains data. This goal is qualified by stating that as much as possible, currently existing IMS protocols and mechanisms should be re-used. It is also necessary to be able to specify a consistent data model that will be used to store the details concerning service capabilities.

A similar challenge of implementing repositories in the IMS has already been solved in the form of the HSS. As a master database, the HSS is responsible for storing user related information such as user identification, user location and user profile information. In addition, as a result of the user data it stores, it is also responsible for supporting other functions such as mobility management, call/session support, access authorisation, service authorisation support and service provisioning support. These features make the HSS a very heavy protocol stack as the requirements for implementing a service capability repository are only a subset of the required features of an HSS. An alternative to implementing a new HSS would be to incorporate service capability information within the existing HSS as a new subset of the service provisioning support of the HSS, but this modification would violate our design goals.

Another candidate is the eXtensible Markup Language (XML) Configuration Access Protocol (XCAP). XCAP is an HTTP-based client-server protocol that allows clients to read, write and modify portions of XML documents stored on a server called an XML Document Management Server (XDMS) [17]. The protocol uses a mechanism known as XPATH which was developed by the World Wide Web Consortium (W3C) to construct expressions called document selectors that address portions of an XML file with varying degrees of granularity [18]. Resources that are stored on the XDMS must be associated with an application usage (appusage) which is simply a use case for XCAP. Each appusage is given a specific schema that the XML file for that use case conforms to. The schema defines conventions and constraints on how applications are permitted to interact with the resources stored in the file. Each appusage has an application unique ID (AUID) which uniquely identifies that application usage from others. Examples of defined appusages include pres-rules (describes policies governing what presence information can be given to which watchers), resource-lists (defines how buddy lists are managed) and rls-services (defines service addresses for linking groups of users). It is anticipated that a record in the service capability repository will consist of details such as the attributes of the service capability, a list of rules for normal behaviour, a list of rules for behaviour during error conditions, dependencies on other service capabilities and other items of information. These details can easily be captured in XML format.

D. Service Repository

The service repository is similar to the service capability repository since both address issues of storage and access to data. As a result, much of the discussion related in section IV-C also applies here. Information about application services currently exists in the HSS as a component of the initial filter criteria [9]. This information however is extremely limited since only the application server URL and default handling procedures are recorded. In the current scenario, the requirement is that a set of service capabilities that are associated with a service must be recorded. Again, the option exists to propose an extension of the IMS subscriber data however this comes at the price of modifying established specifications. Thus we propose that the service repository must be kept separate from the standard HSS, and may be implemented using any of the means described in section IV-C.

E. User Service Repository

The third type of repository is the user service repository. The need for it arises from the fact that it is not possible with the two types of repositories identified in sections IV-C and IV-D to amend the orchestration of services based on user preferences. The service repository is a global repository and contains information regarding all services offered in the network, while the service capability repository contains information about service capabilities in the network. None of these associate services to specific users (unlike an HSS) and hence a user service repository is needed.

F. SCIM

The SCIM is at the heart of the service architecture and is designed to interface with both existing IMS elements and those that are being proposed here. The SCIM behaves in part as a SIP AS since it must maintain the standard ISC interface between itself and the S-CSCF. As previously mentioned, the ISC interface is not necessarily used between the SCIM and the service capabilities due to the genericity and multi-protocol nature of the interface between them. However, the SCIM must implement a Diameter client that is able to use the standard IMS Sh interface in order to download user profile information including static interaction management information in the form of iFC from the HSS. The role of the iFC changes in our design in that the prioritisation of services is not determined by the iFC but by user preferences or by the default service composition rules prior to service personalisation. The iFC can be used to instantiate the default rules, but afterward, the personalised rules take effect.

The protocol that is used to query the repositories will be determined by the implementation. As previously mentioned in section IV-C, candidate protocols are Diameter and XCAP. If Diameter is used, an implementer may be tempted to use either the Sh interface (which usually exists between a SIP application server and the HSS) or the Cx/Dx interface (which usually exists between the I-CSCF/S-CSCF and the HSS). The problem with this implementation is that the command codes that are defined under these interfaces were created with a specific user profile schema in mind. The structure of the records in the proposed repositories is different, thus the Sh and Cx/Dx interfaces cannot be used in their currently specified state. The implementer would have to create a new application protocol based on Diameter and implement a totally new IMS interface separate from Sh and Cx/Dx. This route, while possible, is not a good choice in our opinion.

Another option is to use XCAP. In this case, the SCIM could use the currently defined XCAP client methods as they are to insert, query and update resources in an XDMS-based repository. Unlike the IMS Diameter interfaces, the currently available XCAP appusages are sufficient for managing XML documents on an XDMS that contain service and service capability records. For example, the resource-list appusage can be used when a list of homogeneous elements with certain properties need to be maintained. In common use, records subscribing to the resource-list schema contain buddy lists for users. The same schema can be used to store information about service capabilities. Again, in common practice the rls-services schema is used to reference lists of users' buddy lists and organise them into common groups (such as accounting, personnel, marketing etc). This appusage can be exploited to organise lists of service capabilities deployed by different administrators into groups depending on the service type.

G. Customer

The UE communicates in the standard way with the IMS core which is via the Gm interface which is implemented in SIP. The only difference in our extended design is that there is an additional option for the customer to download and update service associations stored in the service repository to their UE. IMS UEs are required to have an XCAP stack and communicate with an XDMS via the IMS Ut interface [19]. This requirement can be exploited in the case where resources are stored in an XDMS. An XCAP GET operation can be used to download the list of user-subscribed services via the SCIM. This list cannot be retrieved by the UE directly from the services repository since it maintains no associations between deployed services and user subscriptions to them. The SCIM then performs the appropriate Diameter request via the Sh interface and retrieves the needed content in a Diameter User Data Answer (UDA). The SCIM can then do the XCAP lookup on the services repository and download the records by using a selector that is based on the identity of the requesting UE and pass them downstream to the UE via the Ut interface.

Figure 5 shows a call flow that illustrates how the customer's requests are handled. Firstly, the SCIM subscribes for changes on the HSS user profile using Diameter. In the scenario depicted in the figure, the user initially has no service mappings, and a set is created for them dynamically by using

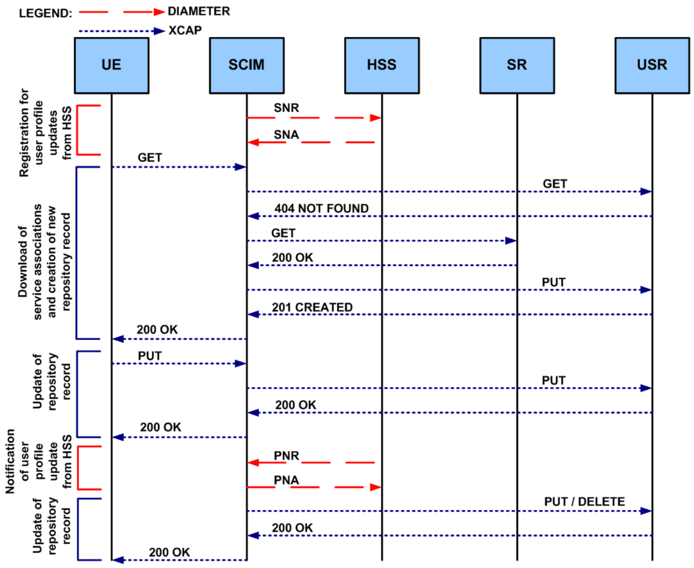

Figure 5. Call flow diagram depicting the interactions between UE, SCIM, HSS and repositories

default mappings in the service repository. After which, the mappings are downloaded by the UE. When the customer wishes to upload changes, these are uploaded to the user service repository. When the IMS Core makes a change to the user profile, the subscribed changes are sent to the SCIM from the HSS. Any changes such as the addition of a newly subscribed service or the removal of an existing subscription must be reflected in the user service repository, thus the SCIM proceeds to make the necessary modifications.

V. Conclusion

The term "service" has become an immensely overused term in the telecommunications research community, and has been used to refer to different things at different times. This ambiguity is removed when we differentiate between services, service capabilities and service capability features. The move to decompose services into their basic building blocks has consensus among the major standards bodies, but introduces challenges for network operators with regards to service brokering and interaction management. In this paper, we have highlighted the complexity of the problem by reducing the scope to only matters of infrastructural requirements for performing service capability interaction management without addressing how that function is performed in a realtime scenario, which is a challenge in its own right. A basic input/output model is applied where the likely inputs to the process of dynamic interaction management are detailed as well as the likely outputs, leading to the introduction of repositories, interfaces and agents to aid in this process. In future, algorithms for realtime interaction management will be explored and our findings will be used to test the architecture that has been proposed in this paper, possibly leading to changes and optimisations.

References

[1] Gonzalo Camarillo and Miguel Garcia-Martin. *The 3G IP Multimedia Subsystem (IMS)*. John Wiley and Sons, Ltd, third edition, 2008.

[2] OMA. Utilization of IMS Capabilities Architecture Version 1.0, TS. Open Mobile Alliance, August 2005.

[3] OMA. OMA Service Environment Version 1.0.5, TS. Open Mobile Alliance, October 2009.

[4] 3GPP. TR 23.810: Study on architecture impacts of service brokering. Technical Report TR23.002v8.0.0, 3GPP, September 2008.

[5] Ning Xia and Wen Jun Zhai. Study on IMS Service Broker. In *ICCIT '08: Proceedings of the 2008 Third International Conference on Convergence and Hybrid Information Technology*, pages 340–343, Washington, DC, USA, 2008. IEEE Computer Society.

[6] Kristin Kocan, William Roome, and Vinod Anupam. Service Capability Interaction Management in IMS Using the Lucent Service Broker Product. *Bell Labs Technical Journal*, 10(4):1–16, 2006.

[7] N. Blum, I. Boldea, T. Magedanz, U. Staiger, and H. Stein. A Service Broker Providing Real-Time Telecommunications Services for 3rd Party Services. In *COMPSAC '09: Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference*, pages 85–91, Washington, DC, USA, 2009. IEEE Computer Society.

[8] Andrew Do-Sung Jun, Gabor Kiss, Hira Agrwal, Hyunseo Park, Sookyang Kang, KyungChul Sihn, and DaeSik Kim. An IMS-Based Service Platform for the Next-Generation Wirelss Networks. *Communications Magazine, IEEE*, 44(9):88–95, September 2006.

[9] 3GPP. TS 23.218: IP Multimedia (IM) session handling; IM call model. 3rd Generation Partnership Project (3GPP), December 2008.

[10] 3GPP. TS 23.002: Network Architecture. 3rd Generation Partnership Project (3GPP), December 2008.

[11] 3GPP. TS 22.228: Internet Protocol IP multimedia core network subsystem. 3rd Generation Partnership Project (3GPP), December 2008.

[12] ETSI. TR 181.004: NGN Generic Capabilities and their use to Develop Services. Technical Report TR181.004v1.1.1, TISPAN, March 2006.

[13] 3GPP. TS 22.105: Services and service capabilities. 3rd Generation Partnership Project (3GPP), December 2008.

[14] N. Bertin E. Gouya, A. Crespi. SCIM (Service Capability Interaction Manager) Implementation Issues in IMS Service Architecture. In *ICC '06: IEEE International Conference on Communications*, pages 1748–1753. IEEE, June 2006.

[15] Yu-mei Wang, Kai Yangi, Rong-juan Li, and Lin Zhang. An improved SCIM-based service invocation mechanism for integrated services in IMS. In *Mobility '08: Proceedings of the International Conference on Mobile Technology, Applications, and Systems*, pages 1–5, New York, NY, USA, 2008. ACM.

[16] Qi Qi, Jianxin Liao, Xiaomin Zhu, and Yufei Cao. DSCIM: A Novel Service Invocation Mechanism in IMS. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1 –5, nov. 2008.

[17] Jonathan Rosenberg. RFC 4825: The Extensible Markup Language (XML) Configuration Access Protocol (XCAP). Internet Engineering Task Force, May 2007.

[18] James Clark and Steve Derose. XML Path Language (XPath) version 1.0. Technical report, World Wide Web Consortium, 1999. http://www.w3.org/TR/1999/REC-xpath-19991116.

[19] 3GPP. TS 23.228: IP Multimedia Subsystem. 3rd Generation Partnership Project (3GPP), September 2009.

Mosiuoa Tsietsi obtained his MSc in Computer Science at Rhodes University where he investigated the application of peer-to-peer protocols for telephony using SIP. He is currently reading towards a PhD at the same institution. His research interests include open source software and service delivery platforms for the IP Multimedia Subsystem.

Alfredo Terzoli is Professor of Computer Science at Rhodes University, where he heads the Telkom Centre of Excellence in Distributed Multimedia. He is also Research Director of the Telkom Centre of Excellence in ICT for Development at the University of Fort Hare. His main areas of academic interest are converged telecommunication networks and ICT for development.

George Wells is Professor and Head of the Department of Computer Science at Rhodes University. He completed his PhD degree in coordination languages for parallel and distributed programming at the University of Bristol, and continues to do research in this area, with a particular interest in methods for the simple, efficient use of multicore processors.