Dynamic Service Orchestration in the IP Multimedia Subsystem

Richard Spiers and Neco Ventura
University of Cape Town, Rondebosch, South Africa
021 650 5296
Email: {rspiers,neco}@crg.ee.uct.ac.za

***Abstract*— Telecommunication operators have begun moving to an all IP-based Next Generation Network (NGN) to save costs as well as enabling the rapid development of new services. A NGN is defined as a packet-based network where the service functionality is independent of the underlying transport technologies. The IP Multimedia Subsystem (IMS) is considered to be an intrinsic part of a NGN, and it provides mechanisms such as user authentication, Quality of Service (QoS) and charging. Traditionally, these mechanisms had to be created for each new service that the network operator wished to offer. The ability to reuse these features greatly speeds up the development of new services. This, together with the open protocols being used for the control signalling, is predicted to create a vast increase in the number of services being offered by the network operator as specialised telephony knowledge is no longer needed to develop new services. This increase in the number of potential services on offer has exacerbated the traditional problem of service conflict detection and resolution.**

This paper describes the current state of research in this area, discussing the advantages and disadvantages of current systems, as well as proposing a novel design that aims to address this issue.

***Index Terms*—SCIM,IMS,SIP**

I. Introduction

The Third Generation Partnership Project (3GPP) was formed in the last month of 1998 to specify the evolution of GSM into a third generation cellular system. It is a partnership between several different telecommunication associations whose aim is to provide technical specifications and reports on 3rd Generation and beyond mobile systems. Originally, there was a heavy emphasis on new higher bandwidth radio access technologies being developed, standardised and deployed. The new radio access methods have increased the available bandwidth to such an extent that new services can be made available to the subscriber.

However, it has been realised that the network architecture needs to evolve and move beyond its circuit and voice based background to take advantage of the higher bandwidth being made available to the end user. This realisation has led to the conceptualisation and subsequent standardisation of the IP Multimedia Subsystem (IMS). The IMS is not intended to act as a standard for services, but rather to aid the creation of new multimedia applications while reducing both initial capital expenditures (CAPEX) and operating expenditures (OPEX). It does this by migrating the telecommunication services away from a circuit switched based network towards an all-IP based network. The IMS was originally designed to merge the world of cellular networks with the Internet. As time has progressed, this scope has been extended to include fixed line access networks. In order to develop this new architecture, the leading cellular standards body (3GPP) looked towards the standards body responsible for the Internet. The Internet Engineering Task Force (IETF) is an open collection of individuals who develop and promote the standards that are used in the Internet. They are responsible for the Session Initiation Protocol (SIP), which 3GPP has chosen to use as the signalling protocol for the IMS. The 3GPP is currently working on release 8 of the IMS specifications, expected to be published in the first quarter of 2009. They have already started planning release 9, which is expected to be frozen in December 2009 as well as planning release 10 [1]. The global sales of IMS equipment, including the core network elements such as the Home Subscriber Server (HSS) and Call Session Control Functions (CSCF) , increased by 94 percent in 2008 compared to 2007. The IMS equipment market is one of the few markets predicted to grow in 2009 and 2010. Although over 100 network operators have begun investigating IMS and have already chosen their vendors, less than 50 % of the networks have live traffic [2].

With the introduction of the IMS, the development of new telecommunication services has been opened to a much wider audience. The standards are freely available, and the signalling protocol is widely known in the Internet world. Developers can begin working on IMS services without needing specialised telephony knowledge. It is expected that this will lead to many different service providers, with many different services. This situation exacerbates the traditional issue of service conflict detection and resolution from previous network architectures, and is one that has been the subject of numerous research projects [3]. As each of these services (call forwarding, video conferencing, Video on Demand etc) could be developed by a 3rd party and not necessarily one network operator, it is vital to introduce some form of service conflict detection and resolution. While parts of the IMS framework have been standardised, this area has been identified as needing further work [4].

This work was supported in part by Telkom SA, Nokia Siemens, Vodacom, the National Research Foundation and the Department of Trade and Industry

A. Service Triggering In The IMS

In the IMS, non user originating services are provided through various Application Servers (AS). How and when these servers get involved in the flow of the control signalling, and thus provide a service, is controlled through the concept of Initial Filter Criteria (IFC). The "Initial" part of the name is kept for historical reasons, as originally the concept of Subsequent Filter Criteria was specified as well. However, this second category of filter criteria would break SIP routing rules, and hence have fallen away. IFC are some of the most important pieces of information stored in the network, as they determine what services will be provided to which users. Each filter criteria contains a set of information related to a user or service that enables the Serving Call Session Control Function (SCSCF) to make a decision on which AS a SIP request should be sent toward in order to provide a service. These filter criteria can only be evaluated with initial SIP requests that create a SIP dialogue, or are stand alone requests. As an example, a IFC would be examined when the SCSCF first receives a SIP INVITE request. The SCSCF would not evaluate its IFC when receiving a SIP BYE message, as it would never create a new dialogue.

These IFC are stored in a user's profile, which is stored in the Home Subscriber Server (HSS). When a user registers on the IMS network, the SCSCF will download their profile from the HSS. These IFC contain a PRIORITY field, which determines the order in which these IFC are evaluated, and hence determine the order that services can be triggered. Once the highest priority IFC has been evaluated and the request has been forwarded to the appropriate AS and back, the next IFC is evaluated as long as the first AS did not answer the request or modify the SIP REQUEST-URI. These priorities may be modified through interaction with an AS, which would result in the HSS pushing the modified IFC to the SCSCF. Two important distinctions can be made here. Firstly, the order in which services are triggered is important. For example, if a voice mail service that acts as a terminating SIP user agent has too high a priority, the user's end equipment would never receive a call. Secondly, the number of services on offer, and hence the number of IFC that need to be evaluated for any given user influences the session setup delay. As more application servers are added in to the signalling path for the session this delay may grow too large, resulting in a bad user experience for the session originator.

B. Improvements Necessary

Currently, the IMS framework does not allow for dynamic service interleaving. It only uses information stored in a SIP message, and can not take advantage of any other information, such as the status of an end user or the current time of day. The current mechanism is based on information drawn from a single user's profile, which limits the flexibility when dealing with multiple user interactions. The current IFC are only evaluated for the initial requests, whereas triggering services on following responses is also desirable [5]. This can currently be done through "record-routing", where a particular network element is inserted into the signalling path and monitors the message flow for a particular response. However, this is generally a solution for individual AS, and needs to be extended to handle multiple service invocations, across multiple different AS. Another limitation of the current system is the lack of service interaction conflict detection and resolution. Currently, there is no mechanism in place to handle the situations where services function properly when used separately, but fail to work correctly when mixed together. 3GPP have identified the need for research in this area, and have concluded a "Study on Architecture Impacts of Service Brokering" in September 2008 [4]. In this document, they state that "the study has identified areas in which service interaction management could enhance the operation of networks ... also identified a number of architecture alternatives and further enhancements that provide a starting point for further development of solutions. The study concludes that continuing the work .... related to service interaction management is recommended". This study also identifies area's that have been left for further study. The most notable of these areas is one of "[allowing] users to personalise and control their services – the architecture should allow end users to personalise and control how applications work together when there are multiple choices of integration available."

The remainder of this paper is organised as follows: Section II contains a review of the current literature on this subject. This is followed by Section III which discusses benefits or drawbacks of each discussed system. Section IV proposes novel enhancements in this area and the paper is concluded by Section V.

II. CURRENT STATE OF RESEARCH

The concept of a "service capability coordination" function was first raised by the 3GPP in 2001 [6]. The first attempt at this functionality defined it very loosely as a stand alone entity that was located in the application layer. It would interface with the SCSCF on behalf of the other application servers, and would use non standardised data to drive this interaction. It arose from the well known issue of service interaction management that had proved difficult in previous network architectures such as Intelligent Networks [3]. However, it was not standardised and was moved inside the architecture of a generic SIP application server, and largely forgotten. Responsibility for it moved through various work groups before finally settling in 3GPP Work Group SA2 (Architecture). This "service capability coordination" function is now known as a Service Capability Interaction Manager (SCIM). A study was started in June 2006 by the 3GPP "to study if there is enhancement needed to the current service interaction management architecture ... in order to satisfy requirements in TS 22.228" [7]. Since then there have been several papers written on this subject. The important features of these papers (in terms of the system's architecture and functionality) are discussed in the following two sections.

A. Architecture

This section will evaluate the current literature with a focus on the architecture of the SCIM. As there is overlap between

the papers cited in terms of how they define their architectures, only new concepts or differing approaches will be discussed from each paper.

Gouya et. al discuss the procedural aspects of a SCIM, which results in them moving the SCIM out of a SIP AS and placing it as a stand alone function [8]. They propose a distributed architecture with a SCIM totally separated from the other logical elements as well as a centralised architecture which has the SCIM integrated with the SCSCF.

Nakajima et. al discuss the issues relating to having different AS provided by 3rd parties [9]. This means that each SIP AS is technically a "black-box" with which one can only interact through SIP messages. They illustrate a problem that can occur in the IMS network if a 3rd party AS behaves incorrectly, namely one of creating massive message loops, and thus come to the conclusion that there needs to be some enhancements made to the SCIM interfaces to protect against these misbehaving AS. They do not discuss their proposed enhancements, but indicate some protection mechanisms that will be discussed further on. Either placing the SCIM inside the SCSCF or outside of it confirms to 3GPP standards from their point of view, with the *Sh* interface (a Diameter interface between the AS and the HSS) carrying the necessary information about the user. Placing the SCIM outside of the SCSCF has an advantage in terms of scalability, as they state that "an SCSCF is considered to be busy handling basic calls. For scaling, the SCSCF has to concentrate on handling basic calls by delegating the proposed functions to the SCIM" [9]. For discussing the best architecture in terms of the impact of a component failing, they place all sessions within three groups, namely basic, service or emergency calls. If a combined SCSCF/SCIM fails, all of these three groups will fail. In the separate architecture, if a SCIM fails only the service calls will fail. The basic and emergency services will still be able to function. If the SCSCF fails, all three of the call groups will fail as there is no longer a component forwarding messages to the SCIM or other destinations. Thus, from a failure impact viewpoint, it makes sense to separate these functions. As the SCIM or SCIM/SCSCF will be communicating with 3rd party application servers, this separation also makes sense from a threat management point of view. Malicious 3rd party AS should not be able to influence emergency calls by causing their point of contact in the network to fail. If they do cause the SCIM to fail, all calls, including service calls, could carry on functioning by being diverted to another SCIM. Another factor affecting the architecture is the flexibility of the resulting functionality. The SCSCF is a SIP proxy, whereas a separate SCIM is allowed to operate as a Back to Back User Agent (B2BUA). A B2BUA can perform call control actions and can maitain the complete call state. This means that the SCIM is allowed to create and terminate SIP dialogues. Thus B2BUAs are able to define their own timers outside of the timers specified in the SIP protocol.

Qi et. al propose introducing a SCIM aware module to each application server [10]. This application module would communicate with the SCIM and retrieve the address of the next AS in the service chain. The SIP message would be forwarded to this next AS straight from the existing AS instead of back to the SCIM and onto the next AS.

Xia et. al discuss Lucent's proprietary approach to a SCIM [5]. Here the SCIM is placed in the control layer separated from the SCSCF, like many of the previous authors. However this approach proposes to allow various other services to interact with the SCIM through different protocols, e.g. HTTP. Their SCIM would translate these HTTP messages into SIP messages, and route them accordingly in the core network. They do not define the interfaces that these different protocols would use.

Goveas et. al place more emphasis on accessing data from the SCIM [11]. They introduce a "Data Federator" module, which handles the retrieval of subscriber and service related data. It fetches the necessary information from the HSS, as well as from any other external data source such as a XDMS server, and thus introduces new interfaces between these remote data storage servers.

B. Functionality

This section will evaluate the current literature with a focus on the functionality of the SCIM. As once again there is overlap between the papers cited, only new concepts or differing approaches will be discussed from each paper.

Gouya et. al identify the need for the SCIM to communicate with several different AS to control the service interaction, as well as to download user information from the HSS through a Diameter interface [8]. They propose creating a SCIM with a "formal model of all the services it may trigger. This formal model would [make] explicit the decomposition of a service into service capabilities and the interactions needed between service capabilities and the end-users."

Nakajima et. al class all possible AS SIP interactions into two distinct groups. The first group is called "Connection Control Applications". This form of interaction controls the call, and includes applications such as "Click-to-Dial", where the user can click a web page link to start a voice call, or Third Party Call Control (3PCC), which can setup and control a call between multiple parties. The second group is called "Co-ordination Applications". According to them, this group includes services such as call redirection, call blocking etc. The difference between these two groups of sessions or calls is important, as it gives rise to the distinction that only one call / session from the Connection Control group can be triggered during some service interaction, whereas multiple services from the second group can be triggered during one session. This is due to the fact that services from the first group terminate the call or session. During service interaction at the proposed SCIM, one can refer to external information, such as presence of the end-user, or information contained in the actual SIP messages. Some services modify this information, which might cause a problem when another service needs to manipulate the same information. Thus, some form of protection needs to be put in place to address this. They also propose two additional protection mechanisms. Firstly, since the AS will be communicating with the SCIM through SIP messages, the SCIM needs to make sure that the SIP messages processed do not violate the SIP specification. Secondly, a

timer needs to be deployed to ensure that the session either carries on or is terminated (depending on the circumstances) if an AS does not respond within a certain time frame.

Kolberg et. al propose wrapping each service in a "cocoon" [12]. A SIP request is sent to this cocoon instead of the service. The cocoon saves a copy of the message and passes it on to the service unmodified. However, when the service is finished with the message, it is passed back to the cocoon and compared to the original message. It keeps track of the services executed and any modifications made by injecting a SIP extension header, known as the CONTYPE header. This header contains information about the service that has been activated (e.g. a unique identifier for the service) and any data that has been changed as a result. It would include the original value of a specific field of the SIP message as well as the changed value. If a cocoon receives a message without a CONTYPE header, it will execute the service as normal. However, if a cocoon receives a message with a CONTYPE header, it would need to evaluate if the service the cocoon is responsible for would conflict with the already activated service (detailed in the CONTYPE header). If it does conflict, either service would need to be disabled. However, this leads to a problem if the second service has priority over the first service, as the first service has already been activated. The original session setup will need to be repeated, but without the first service. This can be done by returning the SIP 380 ALTERNATIVE SERVICE message to the originating user containing the CONTYPE header with a STATUS field set to disable. The user would attempt to establish a new session by sending a INVITE message containing the CONTYPE header. When this header reaches the cocoon of the disabled service, it would see that it is to be disabled, and would not activate the service for this session.

Chiang et. al [13] extend the work done by *Kolberg et. al* [12]. They extend this work to better handle multi-party services, as well as proposing a potential application of these ideas to the IMS architecture. They place the cocoon functionality in the SCIM, which caches every request that passes through it, and add a SERVICE-INDICATION header, which replaces the CONTYPE header.

Qi et. al introduce the concept of utilising XML documents to store information about the application servers as well as network operator policies, which they use to control service interactions [10].

Xia et. al propose having several smaller execution tasks or modules working on each SIP message consequentially, to achieve different services [5].

Goveas et. al use the concept of acyclic graphs to store various different service flows. They define rules for these service flows, and select one depending on the service and user context-specific data. This is done by the introduction of a "Policy engine", which selects the most appropriate static service flow from the available user data. These service flows are defined by the service developer statically before the service provisioning takes place [11].

III. DISCUSSION OF RESEARCH

Each of the systems discussed in the previous section have similarities. The following two sections will discuss the advantages and disadvantages of any differences, in order to come up with a final proposed system, as well as identifying areas needing further work.

A. Architecture

Gouya et. al make several incorrect assumptions when reaching their conclusions. In their centralised network architecture, they have multiple SCIMs, yet claim that this is not a scalable approach as it creates a bottle neck at each SCIM. In their distributed approach, they place the SCIM within the SCSCF, and claim that this does not lead to scalability issues as one can have multiple SCIMs, each one within a separate SCSCF. This point falls away as soon as one realises that they can have the same number of SCIMs in both architectures. The only difference between their architectures is the placement of the SCIM within the SCSCF or outside it. As these are logical functional blocks, and not tied to an actual piece of hardware they could be co-located on one machine or on several machines, thus making this difference in their architectures trivial. They claim that having the SCIM inside the distributed SCSCFs results in a more robust and fault tolerant system as each SCIM/SCSCF failure will have a smaller impact on the overall network. However, if the SCIM fails, the SCSCF will fail as well, and vice versa as they are now one function from an architectural point of view. If the SCIM is placed outside the SCSCF (but still in a one to one distribution, i.e. one SCIM to one SCSCF) and it fails, the network will still carry on operating. The SCSCF will carry on forwarding basic calls, and only enriched services that require a SCIM will be disabled for the subset of users that are currently set to utilise that SCIM. Thus their conclusion that having the SCIM as part of the SCSCF is a more robust approach is incorrect. However, they do make a correct point regarding the reduction of complexity in the SCIM. The SCSCF already has to have the correct interfaces to download user data from the HSS. The SCIM can take advantage of this data, and does not need to implement these interfaces, thus reducing the complexity of the SCIM. This approach has the drawback of needing to modify the actual SCSCF - it is no longer possible to deploy SCIMs in a "plug and play" fashion. Another point to be discussed is that the SCSCF is considered the workhorse in the IMS architecture, as it is included on the signalling path of every dialogue - each message passes through the SCSCF so it can inspect them and take any necessary action. Taking the SCIM out of the SCSCF will reduce the load on the SCSCF instead of increasing it.

In *Nakajima et. al's* motivation for a SCIM they neglect to mention the SIP "hop-counter", which was designed to resolve the issue of message loops. Thus, it is no longer necessary to change the interface between the SCIM and AS as the communication protocol (SIP) already has built-in protection against these flaws.

Qi et. al propose an architecture where the next service to be triggered (or next "hop") is determined in parallel with

the service execution. For this model to have any benefit, the model assumes that the service logic execution of each service takes a long time. Thus, the overall progress is sped up as the messages can be sent straight to the next destination when the processing is finished instead of back to the SCIM. However, the number of signalling messages is increased, which may place additional load on the core network. A major drawback of this architecture is that it does not handle faults well. If an AS fails to respond, the whole service chain is broken, and fault management can only be as sophisticated as the individual module that has been added to each AS allows.

While *Xia et. al's* basic architecture is similar to the other approaches discussed here, they extend it to act as a gateway to various different protocols and systems. This approach conflicts with the pre-existing "best-practice" concept of defining gateway functions to handle interoperability, e.g. OSA Parlay X[14].

B. Functionality

Gouya et. al have identified that the requirement of a formal model, which has a static break down of the service interactions possible, is a flaw that requires future work. The "increased diversity in the domain of next generation services" demands that the "extension of the service capability interaction rules must be performed in a dynamic way to facilitate the introduction of new services" [8]. Manually reconfiguring a core network element every time there is a change in an existing service or a new service becomes available becomes too heavy a burden for the network operator to handle, especially as the number of services on offer increase. This means that there should be some form of automatic service description and detection.

Classing the services into different categories, as *Nakajima et. al* suggest, has several advantages. It makes it easier to detect conflicts between different services in the same category. Another benefit of this approach that has not been explored up until now is that it enables certain services or groups of services to be executed in parallel.

Kolberg et. al's approach leads to many SIP messages flowing back and forth as the session establishment has to be repeated for every possible combination of services that are not compatible. Another drawback is that every service has to know how to resolve these interactions, i.e. which services take precedence over the others. This approach requires heavy modification of each service, as well as continuous modification of every service whenever another service is introduced. Thus, this approach is not suitable for real world deployment.

Chiang et. al extend *Kolberg et. al's* work, placing the cocoon-handling within the SCIM. This means that it is no longer necessary to modify each AS when a service changes, but only the SCIM, reducing the network operator's required maintenance. Their work also leads to a reduction in the signalling over the access network. Most of the message flows now take place over the core network, which is an advantage, especially for the wireless access networks. However, there is still a large amount of redundant SIP signalling as the services which will eventually be disabled are still placed in the signalling path.

With *Xia et. al's* design, the order of execution or service triggering may be changed by each individual module. However, their design is still limited by the fact that only consecutive processing of the services may be done, and a failure in one module would lead to a cascaded failure.

IV. Proposed System

This literature review has lead to several conclusions about the design and functionality of the proposed SCIM:

- It should be a separate functional entity apart from the SCSCF.
- It will need interfaces to communicate with the various AS and with the HSS to gain information about the users.
- Information about each service will be needed by the SCIM to make intelligent decisions on combining the services and detecting conflicts.
- A "history" mechanism is needed to keep track of services invoked by each session and the current state of the session.
- The end user should be able to influence his service priorities / preferences. For example, Alice should be able to determine that only Bob can call her if her status is set to BUSY and the time is during office hours, or that all calls are redirected to voice mail and she receives an email notification whenever Eve attempts to call her.

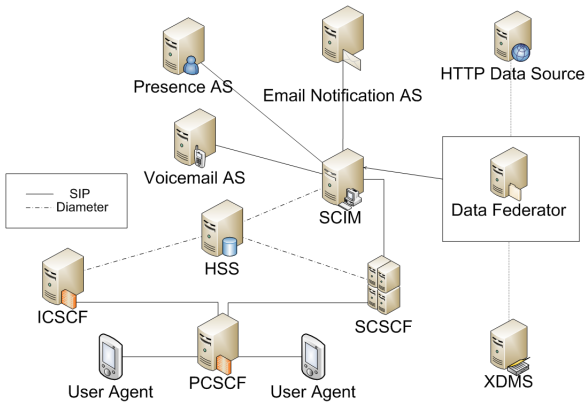

Figure 1. The Proposed Architecture

As the SCIM needs to dynamically route SIP messages based on certain conditions, a decision engine with its own rule set needs to be developed. The necessary execution environment which will allow results from one interaction with one AS to be stored and used as inputs to the next interaction with the same or different AS needs to be created.

To allow dynamic service execution, the SCIM will need to have certain information about each offered service. At the time of writing, there is no published work on automatically registering a new AS (and thus a new service) without manual intervention. A key aspect of the proposed design is that the system can, without human interaction, declare its capabilities and begin serving new requests from users. This information could be entered manually, but this leads to a high degree of management overhead. Thus it is proposed that each AS can

dynamically tell the SCIM about its capabilities and current status. The method used will be based on existing standards for end-user presence, which will be extended and used in a new and innovative way. When a AS comes on-line, it will send a SIP PUBLISH request containing its capabilities which the SCIM would receive based on a certain value in the SIP EVENT header. This also has the added benefit of creating an automatic monitoring system, as each AS will now have the added intelligence necessary to report its status, thus allowing one to get a snapshot of the health of the available services.

All the previous systems discussed still suffer from having to process service requests sequentially. The information sent by the AS to the SCIM in the SIP PUBLISH request will contain details on whether or not the service needs a response or if it can be triggered and forgotten. For example, a service that sends an email to the end user whenever some body tries to call him/her (if the user has set his/her status to busy or do not disturb) does not need to know if the call was successfully handled or modified in some way, just that the attempt was made. All the services that fall into this category could be processed in parallel, greatly speeding up the session set up time.

Another necessary aspect that has not been covered in literature is exposing the SCIM functionality to the end-user. This would allow the user to specifically tailor their service usage, creating a better user experience. It would be controlled by a default policy which would initially be created by the network operator with sane default values, before being manipulated by the end user. These policies would be stored in the network, and would be downloaded to the SCIM to be utilised.

V. Conclusions And Future Work

This paper has reviewed the current state of art regarding service orchestration in the IMS framework. It has concluded that while there is significant research in this area, more research is needed to find the optimal improvements to the current architecture. It has been determined that there is a need for some form of automatic registration of services, and has proposed a mechanism to handle this. Future work involves performance testing of the proposed system in order to determine the suitability of the discussed architecture for real world deployment. Considerations that need to be taken into account are the influence of the new design on session setup delays, the maximum number of total sessions controlled and the maximum number of sessions established per second. These measurements will be compared to another IMS testbed deployed in the standard manner as well as with the analytical models proposed by *Qi et. al* [10].

References

[1] 3G Americas, *The Mobile Broadband Evolution: 3GPP Release 8 and Beyond*, Feb. 2009.

[2] Infonetics Research, "IMS equipment and subscribers report," Mar. 2009. [Online]. Available: http://www.infonetics.com/pr/2009/4q08-ims-market-highlights.asp

[3] M. Calder, M. Kolberg, E. H. Magill, and S. Reiff-Marganiec, "Feature interaction: a critical review and considered forecast," *Comput. Netw.*, vol. 41, no. 1, pp. 115–141, 2003.

[4] 3GPP, "23.810 v8.0.0, On Architecture Impacts of Service Brokering," *(Release 8)(SA2)*, Sep. 2008.

[5] N. Xia and W. J. Zhai, "Study on IMS Service Broker," in *Proc. Third International Conference on Convergence and Hybrid Information Technology ICCIT '08*, vol. 2, Nov. 2008, p. 340–343.

[6] "The IMS lantern: Standardization: SCIM & service broker." [Online]. Available: http://theimslantern.blogspot.com/2007/05/standardization-scim-service-broker.html

[7] "3GPP specification: 23.810 versions." [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/23810.htm

[8] A. Gouya, N. Crespi, and E. Bertin, "SCIM (Service Capability Interaction Manager) Implementation Issues in IMS Service Architecture," in *Proc. IEEE International Conference on Communications*, vol. 4, Jun. 2006, p. 1748–1753.

[9] M. Nakajima, M. Kaneko, M. Hirano, and H. Sunaga, "SIP servlet dialogue handling for NGN Service Capability Interaction Manager," in *Proc. APSITT Information and Telecommunication Technologies 7th Asia-Pacific Symposium on*, Apr. 2008, p. 41–46.

[10] Q. Qi, J. Liao, X. Zhu, and Y. Cao, "DSCIM: A Novel Service Invocation Mechanism in IMS," in *Proc. IEEE Global Telecommunications Conference IEEE GLOBECOM 2008*, Dec. 2008, p. 1–5.

[11] R. Goveas, R. Sunku, and D. Das, "Centralized Service Capability Interaction Manager (SCIM) architecutre to support dynamic-blended services in IMS network," in *Proc. 2nd International Conference on Internet Multimedia Services Architecture and Applications IMSAA 2008*, Dec. 2008, p. 1–5.

[12] M. Kolberg and E. H. Magill, "Managing feature interactions between distributed sip call control services," *Computer Networks*, vol. 51, no. 2, pp. 536 – 557, 2007, feature Interaction. [Online]. Available: http://www.sciencedirect.com/science/article/B6VRG-4KXVB79-1/2/d2ba17665d6f030e94ce37a2444efc63

[13] W. Chiang and C. Tseng, "Handling feature interactions for multi-party services in 3GPP IP multimedia subsystem," in *Proc. International Conference on IP Multimedia Subsystem Architecture and Applications*, Dec. 2007, p. 1–5.

[14] Open Service Access (OSA), "Parlay X Web Services," *(Release 7)*, Jun. 2007.

Richard Spiers is studying for his PhD at the University of Cape Town, having graduated with a B.Sc. in Electrical and Computer Engineering. His previous work in this area involves research on IMS-based IPTV systems as well as video conferencing systems using the IMS platform.