Defining a Service Delivery Platform Architecture by Reusing Intelligent Network Concepts

Rolan Christian and Hu Hanrahan
Centre for Telecommunications Access and Services[1]
School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg
Telephone:(011-7177237), Fax:(011-4031929)
E-mail:{r.christian, h.hanrahan}@ee.wits.ac.za

Network Services

Abstract **–** Telecommunication network operators and service developers need to develop and deliver a variety of single and multimedia based services to customers. To support these new customer services the underlying telecommunications infrastructure must provide various resources and capabilities that are reused in customer service development. These capabilities ensure customer service implementations are independent of the underlying infrastructure technologies and distribution mechanisms. In addition, these capabilities are offered to a variety of external customer service developers. To satisfy these requirements, the Service Delivery Platform (SDP) concept is proposed. The SDP is not standardised and current architectural representations are technology-specific. This paper contributes to the standardisation of the SDP by defining a technology independent SDP architecture. In our approach to define the architecture, we reuse concepts from the standardised Intelligent Network (IN). In this paper we define the SDP and its current architecture limitations. We also review the IN, its concepts and architecture. Based on both SDP requirements and IN we define a standardisable SDP architecture.

Index Terms **–** Standardisation, Architecture, Intelligent Network, Service Delivery Platform.

I. Introduction

Currently, the telecommunications (telco) industry is driven to provide newer services to customers. These services are dictated by current customer requirements, such as the need for integrated voice, video and data. Appropriately, telco infrastructure is able to support these types of customer services due to technological advancement in areas such as application servers, softswicthes, media servers, interworking gateways and Operations and Business Support Systems (OSS/BSS).

In addition to customer services and infrastructure, telco deregulation is enabling *convergence* between telco, Internet and enterprise service developers. With this merging, partnerships and new business initiatives between these service providers are created [1]. Within the telco context, an important initiative is the opening of the telco network to offer its capabilities to external service providers.

[1] The Centre is supported by Telkom SA Limited, Siemens Telecommunications and the THRIP Programme of the Department of Trade and Industry

The *Service Delivery Platform (SDP)* [2] is proposed to manage telco openness, orderly and efficient use of telco infrastructure and customer service development and delivered. The SDP enables customer services to be developed externally by developers that use Information Technology (IT). Also, it provides these developers with managed access to telco capabilities, such that telco capabilities are reused in customer service development.

The SDP is not standardised and many SDP interpretations are based dominantly on technologies and common vendor products. To aid its standardization, the SDP *requires a technology independent architecture that satisfies its requirements*. Similar to other standards, a SDP architecture forms the foundation for additional SDP standards.

We aim to *define a SDP architecture based on reusable and extendable concepts from other standardised architectures*. In this paper we focus on a particular standard and architecture, that is the classical *Intelligent Network (IN)* [3]. The IN, its concepts and architecture, contribute to both legacy and newer telco standards. Also, the IN shares similar requirements with the SDP, such as customer service development and delivery independent of infrastructure technologies. In addition, the IN provides an architecture that forms the basis for its standards, that is the *IN Conceptual Model* [3]. In this paper we reuse concepts from this conceptual model to guide the definition of a SDP architecture.

In this paper we first discuss the define the SDP and its requirements. We also discuss the current SDP perception, its technology influences and limitations. Second, we discuss the approach used to define a SDP architecture. Third, we review the IN architecture and its contribution to the SDP architecture. Last, we define a technology independent SDP architecture and verify its adherence to SDP requirements.

II. Service Delivery Platform

A. Definition and Requirements

The SDP is a *distributed IT platform that uses telco infrastructure to aid customer service development and de-*

livery. Telco infrastructure is the collection of physical elements that offer access to resources and capabilities, such as legacy service platforms, media servers, switches and OSS/BSS. The SDP abstracts functions from the telco infrastructure into generic service components. The SDP service components are independent of the telco infrastructure technologies and distribution, but enable the full potential of the infrastructure capabilities to be used. These service components are used to support customer service development.

The SDP promotes service development by offering *external* service developers simplified access to service components. Within the context of the SDP external service developers belong to IT using enterprises. As a result, the SDP provides IT-based mechanisms to enterprises, such that developers have managed and simplified access to service component functionality. Developers benefit from using service component functionality since they offer telco capabilities that may be used to enhance their existing services. For instance, customer services may use functionality to deliver *content* to customers, determine customer location and charge the customer. Also, service components support customer service delivery over reliable, secure and quality of service transports.

Customer service development requires the integration of service component functionality. Service components support both single-media and multimedia customer service developed and delivered. Multimedia services integrate a combination of voice, video and data. Also, by reusing the service components, customer service implementations gain independence from telco infrastructure technologies and distribution.

A standardised SDP architecture does not exist to manage the above SDP requirements. However, various interpretations of SDP architectures exist, but are influenced by telco, IT and vendor specific technologies. By abstracting common concepts from the various SDP interpretations a general SDP architecture is defined.

B. General Architecture

A generalised SDP architecture, from [2], is illustrated in *Figure 1*. This architecture abstracts common vendor SDP product functionality as platforms. These platforms aim to abstract and simplify telco resources and capabilities and offer their functions to external developers. The platform functions are represented as services. These platforms and their services are:

- Network Abstraction Platform - houses abstract and technology independent *network services* that provide a common point of access to heterogenous transport network capabilities.
- Content Delivery Platform - houses *content delivery services* that provision and deliver content used by telco services for customers. Content is a telco infrastructure resource and either belongs to the telco or is supplied by content providers.
- Management Platforms - contains *management services* that abstract telco OSS/BSS functionality.

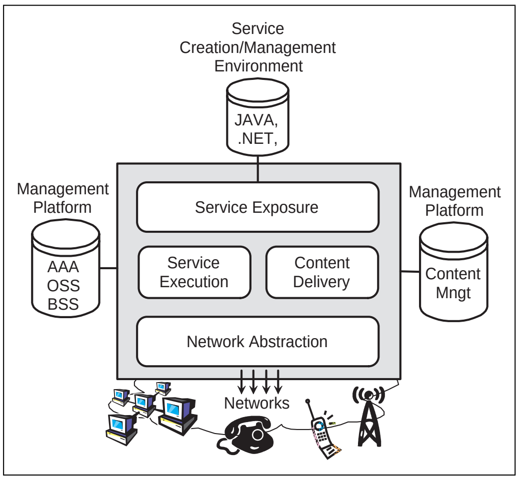

Fig. 1. General SDP Representation

- Service Execution Platform - deploys, executes and manages telco developed applications that provide *telco services* to customers. These services may include classical voice services, data services or multimedia based services. Telco services make use of network, content delivery and management services.
- Service Exposure Platform - defines *exposure services* that provide external developers with simplified access to all platforms services.

The generalised SDP satisfies SDP requirements. For instance, it defines service components by categorising them as either being network, telco, content delivery or management services. Also, it provides external developers with simplified access to service components by using exposure services. However, the generalised SDP architecture is the product of common vendor platform implementations. Vendor implementations are *technology-oriented* and use both standardised and proprietary technologies.

C. Technology Driven

The general SDP architecture is technology driven. For instance, some platform implementations are influenced by the technology-based *Service Oriented Architecture (SOA)* [4].

The SOA is a standardised distributed architecture containing web services. Web services provide simple functionality to access complex resources, systems or other service implementations. Web services provide technology and distribution independence by using XML-based technologies [5], such as WSDL [6] and SOAP [7].

In the context of the general SDP architecture, SOA compliant implementations are envisaged for the service exposure platform and management platforms. For instance, Parlay X [8] standard may be used to implement the service exposure platform. However, no SOA implementation is defined for the management platforms. Hence, vendor-based SOA implementations may be used for implementing the management platforms.

In addition to Parlay X, the Parlay [8] standard is nominated to implement the service execution platform. The

Parlay architecture defines various services and is implemented using CORBA [9]. However, many of the Parlay services may be distributed across the service exposure, service execution, content delivery and management platforms.

Both IT and telco standards influence the definition and structure of the generalised SDP architecture. However, standards have architectures that may be applied across multiple platforms of the generalised SDP architecture. Also, many standards demand specific technology choices to implement their architectures. Vendors that adhere to the generalised SDP architecture may also integrate proprietary technologies into platform implementations. As a result, the generalised SDP architecture remains technology-oriented and this hinders SDP standardisation.

III. Towards SDP Standardisation

Technologies influence the development of the SDP, its service components and the mechanisms used to enable simplified access to these components. Thus, to overcome technology dependence the SDP requires a standardised architecture that promotes technology, implementation and distribution independence. The SDP architecture must satisfy SDP requirements and express the generic concepts of the SDP. By defining this SDP architecture independent of technologies, SDP implementations may use any appropriate technologies. Thus, we aim to contribute to SDP standardisation by defining a technology, implementation and distribution independent SDP architecture that:

1) abstracts telco resource and capability functions into service components,
2) simplifies these service components and
3) provides access to these simplified service components for external developers in IT using enterprises.

To aid the definition of the SDP architecture we reuse concepts from a standardised architecture, that is the classical IN. Though considered legacy, the IN represents a telco milestone that started a trend to separate customer service implementations from the underlying telco infrastructure. Various principles, concepts and architectural structures are defined and standardised for the IN. These IN contributions are beneficial and are reusable in the SDP architecture.

IV. IN Contribution

The IN represents a distributed service platform that promotes customer service development independent of the underlying telco infrastructure implementations. This independence is gained by abstracting telco infrastructure capabilities into reusable software-based service logic. As a result, service logic is integrated to create customer service implementations. Also, the IN supports the delivery of these services to customers on telco transport networks.

Various details on the IN, its definition, concepts and architecture is encapsulated in a single source. This source is the IN conceptual model. The conceptual model represents a reference architecture for the IN. Also, it provides a foundation for the definition of IN standards. As a result, IN-compliant architectures are derived from the conceptual model and adhere to its principles.

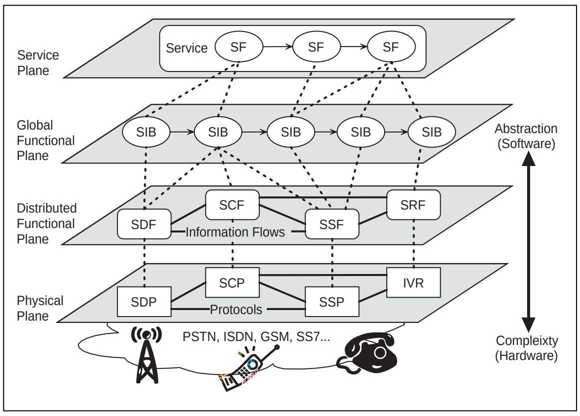

Fig. 2. IN Conceptual Model

The IN conceptual model is the main contribution of the IN that we reuse in the definition of the SDP architecture.

A. IN Conceptual Model

The IN conceptual model provides different *perspectives* on the IN. These perspectives are applied to the IN in a *top-down* approach. Each perspective is represented as a plane that contributes to the structure of the IN. The IN conceptual model is illustrated in *Figure 2*. The figure depicts the various planes.

The *service plane* defines the types of customer services that are provided by the IN. Customer services are described in terms of service features. Service features describe highly abstract capabilities offered by the IN. Hence, this plane does not define any service implementations, rather it defines the available IN *capabilities* for customer service development.

The *global functional plane* defines *Service Independent Building Blocks (SIB)*. A SIB represents a reusable piece of software service logic. A chain of SIBs, that execute in a specific order, represents the implementation of the service feature description. As a result, connections between multiple chains of SIBs implement a customer service. A dedicated SIB named the *Basic Call Process (BCP)* defines logic used to initiate and terminate the execution of SIB chains.

The *distributed functional plane* defines the underlying distributed telco functionality used to implement SIBs. A SIB is decomposed into a chain of function executions. Common functions are encapsulated into functional entities. The communication between functional entities to execute specific functions implements a SIB. The BCP is also implemented across communicating functional entities. These functional entities initiate and terminate communication with additional functional entities, so as to satisfy SIB executions. Functional entity communication is supported by information flows.

The *physical plane* defines the mapping of functional entities to physical network elements. This plane demands specific functional entities to be implemented on specific physical elements. This plane describes the distribution and technology implementations of the functional entities. For example, physical elements implement functional entity information flows using protocols, such as INAP [10].

B. Concepts

The IN increases the *intelligence*, rather than switching levels in the telco infrastructure [11]. The global functional and distributed functional planes are examples of this intelligence. The added intelligence abstracts the complexity of infrastructure technologies and distribution that are represented as physical elements in the physical plane.

The collection of planes represents the increase in *separation* between telco infrastructure, its capabilities and customer services. This separation enables customer service creation to be independent of the underlying infrastructure technologies. Also, this separation enables customer services to reuse infrastructure capabilities.

One of the IN philosophies is the *standardisation* of SIBs [12]. The standardisation promotes additional separation and independence between telco infrastructure and customer service definitions. For instance, changes to infrastructure do not effect SIB definitions and therefore do not effect service feature and customer service definitions. However, telco infrastructure must provide the needed functions to implement SIBs. Therefore, conformance to SIB standards is achieved by providing necessary functions to implement SIB definitions.

Each plane of the conceptual model provides a different *perspective* on the IN. The service and global functional planes provide a *service-oriented* view on the IN [13]. In this view technology and distribution of the telco infrastructure is hidden. The service plane enables developers to verify if the needed service capabilities are provided to define a customer service. These service capabilities are the service features. In the global functional plane developers view the integration of SIBs as the implementation of the service feature definitions and therefore the customer services.

The distributed functional and physical planes provide a *functional-oriented* view on the IN. In this view, only functional entities and their functions, as offered by physical elements, are perceived. The functional entities and their functions represent the capabilities offered by the physical elements. This view also provides the details on the technology and distribution of the functional entities and physical elements.

Though customers are connected to the telco via the transport network, customers perceive their *interactions* are with their services. The distributed functional and physical planes support this communication between customer and service.

Collectively an integrated *managed environment* for SIBs, functional entities and physical elements is proposed for the IN. This managed environment is based on the *Telecommunications Management Network (TMN)* [14] concepts. Though not fully standardised, the general TMN architecture provides layers containing network, service and business functions that manage the network elements, service elements and business processes respectively. These layers are incorporated into the IN planes.

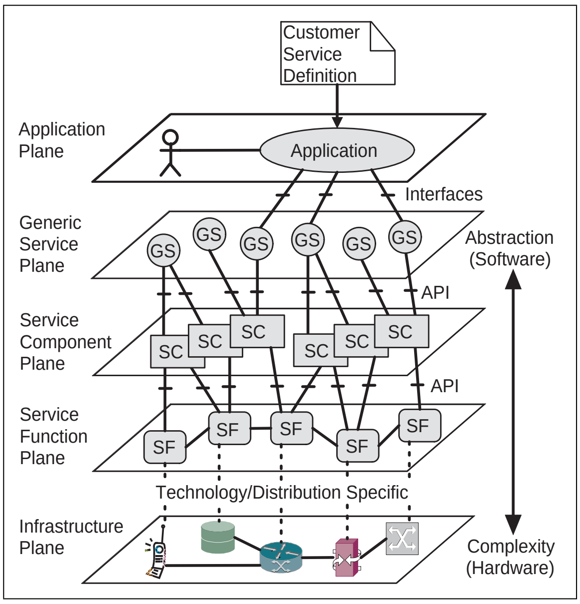

Fig. 3. SDP Conceptual Model

However, details on each TMN layer, its functions and implementation requires standardization.

By applying the conceptual model properties and principles to the SDP, within the confines of the SDP requirements, we define a SDP architecture. We name this architecture the SDP conceptual model.

V. SDP Conceptual Model

Similar to the IN conceptual model, the SDP conceptual model provides different perspectives on the SDP that are applied in a top-down approach. The SDP conceptual model is illustrated in *Figure 3*. The figure illustrates the SDP as a collection of layered planes.

First, customer services are defined by developers, using their specific software methodology and tools. The service definition also documents the interactions between customer and service, as well as the capabilities required from the telco to implement the customer service. Once completed, the customer service definition is implemented by developers as an application in the *application plane*.

Second, applications make use of telco capabilities by integrating reusable functions offered by *generic services*. Generic services provide simplified access to telco capabilities via their implementation independent interfaces. As a result, generic services may be grouped according to the capabilities they simplify. For example, services may provide telco, data, content or management-oriented functions. The collection of generic services are grouped into the *generic service plane*. This plane supports the generic services and their implementation independent interfaces that are exposed and used by applications.

Third, generic services simplify complex services that abstract telco infrastructure resources and capabilities. These complex services are named *service components*. Service component functionality is offered via a *Application Programming Interface (API)*. Thus, service components represent a common point in accessing telco capabilities. The collection of service components are managed in the *service component plane*.

Fourth, underlying telco infrastructure contains distributed, standardised and proprietary systems that are implemented and connected using various technologies. However, these systems provide a solid foundation of reusable resources and capabilities. To enable technology and distribution independent access to these resources and capabilities, their functions are abstracted into services. These services are named *service functions*. Service functions offer their functionality to service components via an API. Service functions communicate amongst themselves to satisfy service component requests. Hence, the result of service function communication is the execution of functions on telco infrastructure systems. The collection of service functions and their communication is managed in the *service function plane*.

Last, telco infrastructure is contained within physical elements and are connected on transport networks. These physical elements are managed in a *infrastructure plane* that enable their technology and distribution dependent functions to be accessed and used by service functions.

A. Concepts

The SDP conceptual model also increases the levels of *intelligence* in the telco. The service planes represent these intelligent levels. Higher service planes abstract the complex intelligence of lower planes, such that simpler access to telco capabilities is provided to external developers. Like the IN, the SDP conceptual model planes hide distribution details of applications and services.

The SDP conceptual model planes increases the *separation* and *independence* between telco infrastructure and application implementations. This independence is gained by the hierarchy of planes and services. As result, applications that use generic services are not integrated with the telco infrastructure. The various services support this independence by providing its functions to each other and applications via interfaces. These interfaces provide a means to access services independent of their implementation.

Similar to the IN, the SDP must implement its services and service planes using *standards*. By implementing standards further independence of applications and telco infrastructure is gained. As a result, the telco conforms to these standards and ensures infrastructure can support standardised services. Also, developers benefit from standardised services since they have a constant repository of reusable services for application development. Candidate standards for generic services are Parlay X, while Parlay may be used to implement the service components. Also, the IMS [15] may be used to implement a set of service functions, where SIP [16] provides an implementation for service function communication. Additional standards may be used in conjunction with the IMS and SIP to implement remaining service functions and their communications. As a result, service planes may be implemented as distributed platforms, that support the implementation of these standards.

Like the IN conceptual model, there are two perspectives created by the SDP conceptual model. The application, generic service and service component planes provide a *service-oriented* perspective on the SDP. For instance, customers view the telco as offering services (applications) and perceive they interact with these applications directly. Also, developers view the generic service plane as a resource of generic services to integrate into applications. The developer's view is limited to the generic service plane, while the telco is able to view the inner workings of the SDP. For example, the telco views the service component plane as a resource of service components to integrate into generic services. The telco also views the service function plane as a resource of service functions to integrate into services components. In these service-oriented views service implementations, technologies and distribution is hidden by the various planes and service interfaces.

The service function plane, in conjunction with the infrastructure plane, also provides a *functional-oriented* view on the SDP. In this view the telco perceives the infrastructure plane as a resource of technology and distribution specific capabilities to be abstracted and integrated into services functions.

Each plane in the SDP conceptual model is administered within a *management environment*. Like the IN conceptual model, a managed environment administers applications accessing generic services, service interactions, all services and telco infrastructure. Each plane contributes to this management environment. For instance, a service component plane implementation may provide management functionality to administer service components. Also, services in the service planes abstract telco OSS/BSS functionality that may be reused in this management environment. Like the IN and TMN, a separate management architecture may be defined, based on the SDP conceptual model.

VI. Satisfying Requirements

The IN and SDP conceptual models are summarized in *Table I*. In the table, the generic service and service component planes share concepts with the global functional plane. For instance, these planes define and implement reusable service logic used in customer service implementations. Also, the service function plane shares concepts with the distributed functional and physical planes. For example, these planes define and implement services that:

- contribute functionality to service implementations in higher planes;
- communicate with each other to fulfill their functions and
- abstract technology and distribution details of infrastructure.

Though the SDP conceptual model reuses concepts from the IN conceptual model, it adheres to SDP requirements. For instance, the service function plane manages the complexity of the telco infrastructure by abstracting and integrating telco capabilities into service functions. These service functions provide a generic API to access these capabilities. Service components make use of this API to offer

<table>
<tr><th colspan="2">IN Conceptual Model</th><th colspan="2">SDP Conceptual Model</th></tr>
<tr><th>Plane</th><th>Description</th><th>Plane</th><th>Description</th></tr>
<tr><td>Service</td><td>customer services described by integrating service feature descriptions. No implementation.</td><td>Application</td><td>customer services are described and implemented by enterprise. This is external to the telco and SDP.</td></tr>
<tr><td rowspan="2">Global Functional</td><td rowspan="2">building blocks (SIBs) define reusable service logic. Chains of SIBs implement service feature descriptions and therefore customer services.</td><td>Generic Service</td><td>application implementations reuse generic services via technology independent interfaces.</td></tr>
<tr><td>Service Component</td><td>service components offer a generic API to access service related telco capabilities.</td></tr>
<tr><td>Distributed Functional</td><td>implement SIBs using distributed functional entities that execute functions via information flows.</td><td rowspan="2">Service Function</td><td rowspan="2">service functions offer a generic API to access infrastructure specific capabilities. Also, service functions implement a communication mechanism to interact and satisfy service component request.</td></tr>
<tr><td rowspan="2">Physical</td><td rowspan="2">functional entities map to physical elements that implement their functions using specific technologies. Also, specific protocols implement functional entity information flows.</td></tr>
<tr><td>Infrastructure</td><td>connected and technology and distribution specific telco systems provide service functions with access to their specific functions.</td></tr>
</table>

TABLE I
SUMMARY OF CONCEPTUAL MODELS

less complex, more specific and reusable service functionality to developers. The generic services provide managed and simplified access to service components by offering technology independent interfaces. As a result, external developers from IT using enterprises are not bound to specific implementation technologies for customer service development. Hence, the SDP conceptual model is implementation, technology and distribution independent.

Additional decomposition on the planes and their services lead to more detailed structure and perspectives on the SDP. This structure and perspectives may be defined in future standards that use the SDP conceptual model as a foundation of concepts.

VII. CONCLUSION

We have defined a SDP architecture (SDP conceptual model) to contribute to the definition of SDP standards. The SDP architecture is based on reusable concepts from the IN conceptual model. The SDP architecture is independent of implementation, technologies and distribution. The architecture adheres to the generic SDP requirements by defining layered planes containing generic and reusable services that abstract infrastructure resources and capabilities. These services enable external developers, from IT using enterprises, to reuse their functionality in customer service development. Services provide technology independent interfaces that enable access to their functionality. Customer services are developed by integrating services' functionality into applications. Each plane and its services are implementable using a variety of standardised technologies, such as SOA and web services, Parlay X, Parlay, IMS and SIP.

REFERENCES

[1] A. Henten *et al*. *The Next Step for Telecom Regulation: ICT Convergence Regulation or Multisector Utilities Regulation*. *The South African Journal of Information and Communication*, (3), 2003.

[2] The Moriana Group. *Service Delivery Platforms and Telecom Web Services*. A Moriana Thought Leader Report, Section A: Executive Summary, Last accessed on 25/04/2006, http://www.morianagroup.com, June 2004.

[3] ITU-T. *Principles of Intelligent Networks*. Recommendation Q.1201, October 1992.

[4] D. Booth *et al*. *Web Services Architecture*. Working Group Note, W3C Web Services Architecture Working Group, February 2004.

[5] T. Bray *et al*. *Extensible Markup Language (XML) 1.0 (Third Edition)*. Recommendation, W3C, February 2004.

[6] E. Christensen *et al*. *Web Services Description Language (WSDL) 1.1*. Technical Report 1.1, W3C, March 2001.

[7] M. Gudgin *et al*. *SOAP Version 1.2 Part 1: Messaging Framework*. Recommendation, W3C, June 2003.

[8] The Parlay Group. *The Parlay Home Page*. Last accessed on 25/04/2006, http://www.parlay.org.

[9] O.M.G. *Common Object Request Broker Architecture: Core Specification*. Formal Version 3.0.3, March 2004.

[10] ITU-T. *General Aspects of the Intelligent Network Application Protocol*. Recommendation Q.1208, September 1997.

[11] J. Thörner. *Intelligent Networks*. Artech House Telecommunications Library. Artech House Publishers INC, 1994.

[12] J. Zuidweg. *Next Generation Intelligent Networks*. Telecommunications. Artech House Publishers INC, 2002.

[13] T. Magedanz and R. Popescu-Zeletin. *Intelligent Networks*. Networks and Communications/Telecommunications. International Thomson Computer Press, 1996.

[14] ITU-T. *Principles of Telecommunications Management Network*. Recommendation M.3010, May 1996.

[15] 3GPP. *Technical Specification Group Services and System Aspects; Service Requirements for the Internet Protocol (IP) Multimedia Core Network Subsystem; Stage 1 (Release 7)*. Technical Specification TS 22.228 V7.0.0, September 2004.

[16] J. Rosenberg *et al*. *SIP: Session Initiation Protocol*. RFC 3261, IETF Network Working Group, June 2001.

BIOGRAPHIES

Rolan Christian holds a BSc and BSc Honors degree in Information Technology from the University of Natal. He also holds a MSc degree in Electrical Engineering from the University of the Witwatersrand. He is currently pursuing his PhD degree in Electrical Engineering at the University of the Witwatersrand.

Hu Hanrahan is Professor of Communications Engineering at the University of the Witwatersrand. He leads the Centre for Telecommunications Access and Services (CeTAS), a research and advanced teaching centre devoted to improving knowledge and practice in the evolving telecommunicationms access networks and telecommunicationms services.