

SOA-BASED GUIDELINES FOR VALUE-ADDED SERVICE DEVELOPMENT ON JAIN SLEE ENVIRONMENTS

Julián Andrés Rojas Meléndez^{*}; Jesús David Ramírez^{**}; Juan Carlos Corrales^{***}

Recibido: 02/11/2011

Aceptado: 05/10/2012

ABSTRACT

The appropriation of the new technologies is a key aspect to maintain high competitiveness in a globalized economy for telecom carriers. The benefits provided by these new technologies allow them to easily adapt to the market conditions and to satisfy the changing needs of increasingly demanding users. One of these technologies is the JAIN SLEE specification, which presents a highly robust platform for rapid development and deployment of new value-added services. However the high degree of complexity of this technology limits the exploitation of its benefits and capabilities. In this paper, we propose a set of guidelines for value-added service development on JAIN SLEE environments based in the principles and guidelines for service creation defined in SOA, in order to facilitate this process.

Key words: Value-Added Service, JAIN SLEE, SOA, Service Building Block, Time-to-Market.

^{*} Estudiante de Maestría en Ingeniería Telemática, Investigador del Grupo de Ingeniería Telemática de la Universidad del Cauca, Colombia, Popayán - Cauca, Carrera 17 # 19N-166B, Tel: (+57-2) 8230443, email: jarojas@unicauca.edu.co

^{**} Estudiante de Maestría en Ingeniería Telemática, Investigador del Grupo de Ingeniería Telemática de la Universidad del Cauca, Colombia, Popayán - Cauca, Calle 35 N # 21-200, Tel: (+57-2) 8333009, email: jdramirez@unicauca.edu.co

^{***} Ph.D en Informática, Profesor Titular del Departamento de Telemática de la Universidad del Cauca, Colombia, Popayán - Cauca, Edificio de Ingenierías sector Tulcán, Oficina 403, Tel: (+57-2) 8209800, ext. 2129, email: jcorral@unicauca.edu.co

LINEAMIENTOS BASADOS EN SOA PARA EL DESARROLLO DE SERVICIOS DE VALOR AGREGADO EN ENTORNOS JAIN SLEE

RESUMEN

La apropiación de nuevas tecnologías por parte de los operadores de telecomunicaciones es un aspecto clave para mantener una alta competitividad en una economía globalizada. Los beneficios que proporcionan estas nuevas tecnologías les permiten a los operadores adaptarse fácilmente a las condiciones del mercado y satisfacer las necesidades cambiantes de usuarios, cada vez más exigentes. Una de estas tecnologías es la Especificación JAIN SLEE, la cual presenta una plataforma altamente robusta para el rápido desarrollo y despliegue de nuevos servicios de valor agregado. Sin embargo, el alto grado de complejidad de esta tecnología limita la explotación de sus beneficios y capacidades. En este artículo se propone una serie de lineamientos para el desarrollo de servicios de valor agregado en entornos JAIN SLEE basados en los principios y lineamientos para la creación de servicios definidos por SOA, de forma que se facilite este proceso.

Palabras clave: servicio de valor agregado, JAIN SLEE, SOA, bloque constructor de servicio, tiempo de despliegue.

INTRODUCTION

In recent years, the rapid development of information and communication technologies has given rise to a new set of business models in the telecommunications industry such as Telco 2.0 [1]. This new model aims to integrate the new information services with traditional telecommunication services, taking advantage of the convergence of the different networks underlying this type of services. It also has generated a great change in the service market conditions for telecom operators, where to stay competitive, it is imperative the development of new value-added services with a very low time-to-market. For this, is necessary that companies in the sector adopt new technologies that enable the rapid development of this type of services. Among these new technologies is the JAIN SLEE (JSLEE) specification [2, 3] which presents a platform for rapid development and deployment of new value-added services. However, technologies such as JSLEE also present a high degree of complexity, which limits the use of their capabilities and benefits in the context of reducing the time-to-market of new value-added services.

Otherwise, Service Oriented Architecture (SOA) has emerged as the architecture of choice in the telecommunications industry to meet the new market challenges and conditions. SOA provides vital principles, guidelines, framework and technique to help organizations to fully realize their business goals [4]. Within these principles and guidelines, SOA defines a service creation model, which can be used as a reference for the development of value-added services.

This paper propose a set of guidelines to develop robust value-added services on JSLEE environments, based on the service creation model defined in SOA, aiming to reduce the time-to-market of new services over this type of technology. The next section will present such guidelines by making a description of each of them. Section 2 presents a brief description of JSLEE architecture. Section

3 provides a description of a service developed following the guidelines in order to validate the effectiveness of these. Finally, section 4 will analyze the results and conclusions drawn from this work.

1 JAIN SLEE ARCHITECTURE

JAIN SLEE aims at defining a new kind of application server designed for hosting value-added services. In particular, a JSLEE container is designed for hosting communication applications while typical application servers have been designed for enterprise applications and they usually do not consider high-availability and performance concerns. Instead, the JSLEE specification has been designed for communications applications, and a JSLEE container relies on an event based model, with asynchronous interactions among components [5].

The atomic element defined by JSLEE is the SBB (Service Building Block). An SBB is a software component that sends and receives events and performs computations based on the receipt on events and its current state. SBBs are stateful components because they can remember the results of previous computations and those results can be applied in additional computations. SBBs perform logic that is based on the receipt of events. Events are used to represent occurrences of importance that may occur at arbitrary points of time. For example, the act of an external system delegating a call setup to the SLEE. An event may asynchronously originate from a number of sources, for example, an external resource such as a communications protocol stack, from the SLEE itself, or from application components within the SLEE.

Resources are external entities that interact with other systems outside the SLEE, such as network elements (i.e., messaging server or SIP server). JSLEE defines the resource adaptors which adapts the particular interfaces and requirements of an external resource into the interfaces and requirements of JSLEE [6].

2 GUIDELINES FOR JAIN SLEE SERVICE DEVELOPMENT

Below we outline a series of guidelines for carrying out the development of value-added services on JAIN SLEE environments; this is done in order to provide an organized and easy way to create new services. The guidelines are organized as follows: i) first a business model analysis is performed to obtain the initial requirements, ii) from these requirements are identified and specified the different service building blocks that compose the service, then iii) relations are established between the service building blocks. The last step iv) is the service implementation. This process is based on the service creation model under a SOA approach proposed in [7], it also uses the JAIN SLEE specification as a reference for defining the service components.

2.1 Business Modeling

To perform the business modeling, SOA proposes a set of steps and tasks based on the concepts defined by RUP [8], whose purpose is the identification and determination of the specific requirements necessary to create a solution that meets business objectives. The typical sequence of steps for creating the business model is the following:

- **Business Vision Definition:** This task captures the high level objectives within the business modeling process. Is performed from the point of view of the different actors involved in the business idea, seeking an understanding of the existent relationships and the context.
- **Business Use Cases Model Creation:** The business use cases model is a technique used to describe the business from an outside perspective. More formally, a business use case is "...a sequence of actions that a business performs that yields an observable result of value to a particular business actor, or that shows how business responds to a business event, to yield a business benefit." [7].
- **Business Architecture Definition:** The business architecture provides an overview of the relevant parts of the business in terms of their products, processes and services. It is used to identify the key components of the business and generates a pattern for building applications, services and other technical elements involved in the business.
- **Capture and Identification of Business Objectives and Requirements:** Through the different concepts and steps described above it is possible to identify the main and specific objectives of the business idea, as well as the requirements for its realization.

Following the steps above is possible to perform an analysis of a business idea that allows the construction of a satisfactory solution that meets the proposed objectives.

2.2 Service Building Blocks Identification

The identification of service building blocks (SBBs) has as main purpose defining the set of SBBs that will be used to provide the functionality established by the requirements. The identification also provides a description of the function of each SBB and the way they are related to each other [9]. The steps for performing this task are as follows:

- **Existing Resources Analysis:** This analysis is performed in order to find components that have already been implemented and can be reused to meet the requirements of the new service. These components are SBBs, resource adapters and other JSLEE elements.
- **Sub-processes Definition:** This step consists in decompose the logic sequence of each business use case previously defined into smaller processes or sub-processes, in order to ease the identification of the SBBs needed to meet the requirements of each one of them.
- **SBBs Definition:** This step determines what SBBs are needed in the implementation of

the solution, based on the analysis of each sub-process composing every business use case previously defined.

- **Sequence Diagrams:** In this step are defined the sequence diagrams of each business use case. These diagrams use the SBBs defined above and specify the relations between them, following the service logic sequence. UML formal notation [10] is used for defining the sequence diagrams.
- **SBBs Specification:** Once SBBs necessary to meet the solution requirements and their relationships to each other have been defined, within these guidelines is proposed to perform a standard description of each SBB based on the model defined in [11] in order to facilitate their implementation. Table 1 shows the proposed template to specify the SBBs making a description of each field.

2.3 SBBs Classification

In order to simplify the SBBs identification process, within these guidelines is proposed a clas-

sification model based on the role of each SBB in the construction of the solution, specifically on the relations established between them. Two different types of SBBs are defined:

- **Atomic SBB:** This SBB provides a specific functionality (e.g. data base access, protocol handling, etc.) and characterized by not establishing relations with other SBBs within its function logic. On the contrary, its functions and capabilities are invoked by other SBBs according to the defined service logic.
- **Composed SBB:** This SBB has as main feature the definition of the relationships with other SBBs within its function logic. It implements the high level logic of the service so its proper operation depends on the invocation of specific functions and capabilities defined by other SBBs.

2.4 Relationships between SBBs

Through the establishment of relations between SBBs is possible to implement the logical sequence of a developing value-added service. Ac-

Table 1. SBB Specification Template.

<i>Feature</i>	<i>Description</i>
<i>SBB ID</i>	Identification name of the SBB.
<i>Developer</i>	Information regarding the SBB developer.
<i>Version</i>	Implementation version of the SBB.
<i>Function Description</i>	Brief description of the SBB functionality.
<i>Abstract Class Name</i>	Name of the main class which implements the SBB function logic.
<i>Local Interface Name</i>	Local interface name (if it exists) which publishes the methods to be used by other SBBs.
<i>CMP (Container Managed Persistent) Fields</i>	Name and type of the CMP fields used by the SBB.
<i>Specification Profile Name</i>	Name of the profile specification used by the SBB.
<i>Child SBBs</i>	Description and priority of each managed SBB.
<i>Events</i>	Description of the different events generated or handled by the SBB.
<i>Methods</i>	Brief description of each method implemented by the SBB.
<i>Resource Adaptors</i>	Description of the resource adaptors used by the SBB if any.
<i>Alarms</i>	If any, brief description of its porpoise.

Source: the authors.

According to JSLEE specification there are two different types or relations between SBBs, synchronous and asynchronous relations. Below a description of each one is presented:

- Synchronous Relation:** In a synchronous relation there is an explicit connection between the involved SBBs. The logical process carried out to accomplish a specific task is done step by step with a predetermined logical order. The establishment of this type of relation is based on the definition of a parent-child relationship, called Child Relation. In a Child Relation, the parent SBB is the one that defines the relationship and makes use of the functionality of another SBB which is called child SBB. Child Relations are defined by the parent SBB through XML files.

Once a *Child Relation* between two SBBs is defined, the parent SBB may use it in two different ways. The first one consists in the invocation of the functions and capabilities of the child SBB through an interface called *SbbLocalObject*. The second one consists in the transfer of events, i.e. when the parent SBB receives a certain event this transfers the event to the child SBB to be processed.

- Asynchronous Relation:** Unlike synchronous relations, in an asynchronous relation there is not a direct connection between the involved SBBs and there is not an explicit definition of the relationship in the XML descriptor files of the SBBs. Asynchronous relations do not distinguish between the involved SBBs as in synchronous ones, which defines parents and child SBBs.

Asynchronous relations between SBBs consist in receiving and firing events throughout the Service Logic Execution Environment (SLEE), i.e. logic operations are performed after the receipt of an event by a SBB, which processes it and may proceed to fire other events that are received by different SBBs. This process may be performed as often as necessary within a service to accomplish a task or function.

Figure 1 shows both synchronous and asynchronous relations established between SBBs.

2.5 Service Implementation

The last step defined within these guidelines is the service implementation which is the final product obtained from the application of every step described above. This section describes the

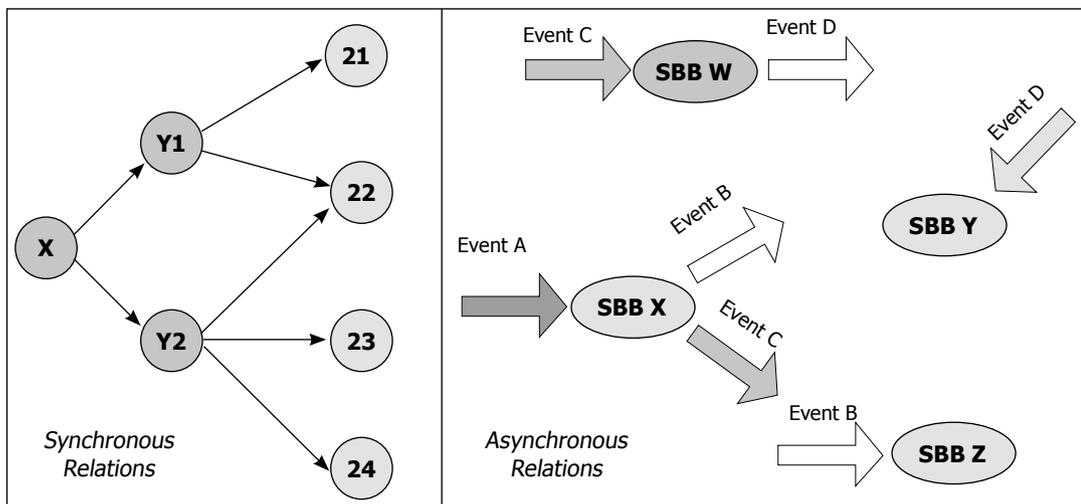


Figure 1. SBBs Relation Types.

Source: the authors

necessary steps to implement a JSLEE service through the Service Creation Environments (SCEs) EclipSLEE [12] and Alcatel-Lucent SCE-SE [13], proposed as the best alternative to develop JSLEE services in [14]. The proposed steps are as follows:

- **Service Project:** This is the first step and consists in the creation of the service project which defines the service body and contains all the service components.
- **Events and Profile Specifications:** In this step the developer creates and defines the profile specifications and costume events (i.e. events created by the developer that do not belong to any protocol stack) needed in the logic operation of the service.
- **Service Building Blocks:** For the SBBs creation is necessary to consider the type of relationships defined between them, because in case of having defined synchronous relationships, it is essential to create the atomic SBBs first and then the composed SBBs. If the type of relations defined is asynchronous, the order of

creation of the SBBs is irrelevant.

- **Service Descriptor and Deployable Unit:** The service descriptor is a XML file that identifies the service within the SLEE and the deployable unit is the file to be executed in the SLEE.

3 VALIDATION PROTOTYPE

This section makes a brief description of the prototype used to validate the guidelines proposed in this paper. Such validation consists in the development of a value-added service using the guidelines described above. The service developed is named *Reminder Call*; this one provides the capacity to define through a Web interface a list of phone numbers to be automatically called to deliver a predetermined message. This service represents a very helpful tool for companies that handle loan recovery and require reminding their costumers their respective payments or may be used to promote new products throughout their customers.

Figure 2 shows the SBBs and JSLEE elements that compose the *Reminder Call* service distinguis-

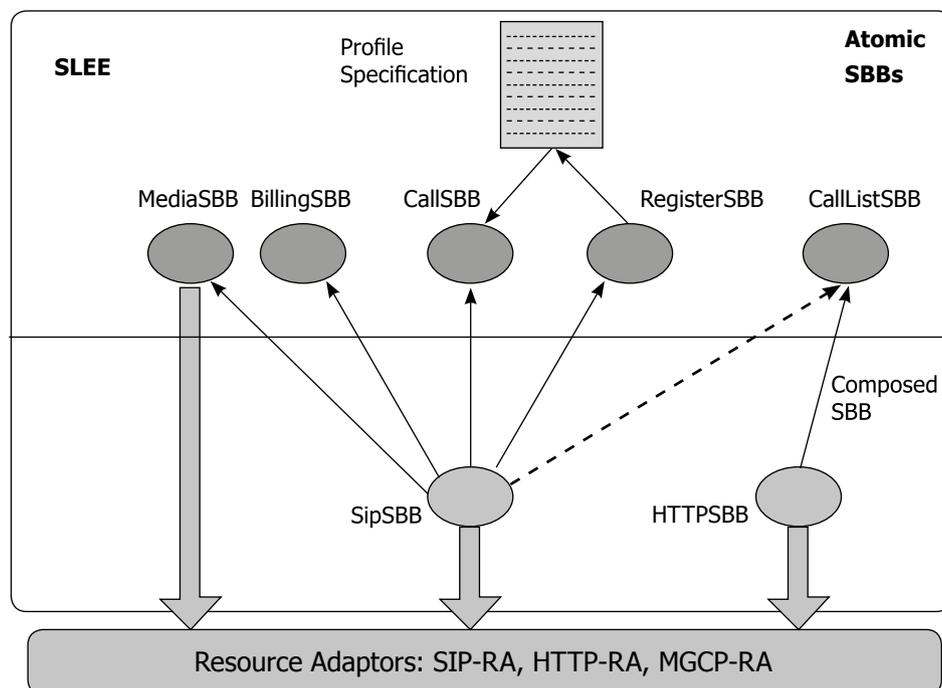


Figure 2. Reminder Call Service SBBs Diagram.

Source: the authors

hing between the atomic and composed SBBs. The composed SBBs are the *SipSBB* and the *HttpSBB* which are responsible for receiving and processing all the SIP and HTTP events that reach the SLEE. The atomic SBBs defined within the service are the following:

- **MediaSBB:** Is responsible for creating a connection with the Media Server and reproducing the predetermined message through the MGCP protocol.
- **BillingSBB:** This SBB stores in a data base all the information regarding the calls made by the service.
- **CallSBB:** This SBB obtains the contact information of each user to be called by the service from the profile specification.
- **CallListSBB:** This SBB is used by the composed SBB *HttpSBB* to generate all the calls using the list the user defined through the Web interface.
- **RegisterSBB:** This SBB is responsible for updating the contact information of every user in the profile specification.

Finally, it should be noted in figure 2 that the dashed line between *HttpSBB* and *CallListSBB* represents an asynchronous relationship, while the solid lines between the other SBBs represent synchronous relationships.

4 RESULTS AND CONCLUSIONS

The previous section describes the value-added service developed to validate the guidelines for value-added service development on JSLEE environments proposed in this paper. This service was developed and deployed over the telecommunications infrastructure of EMCALI EICE-ESP [15] in the city of Santiago de Cali, Colombia. Figure 3 shows how we adapted our proposal in the telecommunications infrastructure of EMCALI, for deploying the value-added service.

The application logic of the value-added service described in section 4 is allocated in the Mobicents

JSLEE server. This server communicates via HTTP with the Web server Glassfish containing the Web application that allows users to interact with the service. It also communicates with the Mobicents Media server via MGCP which provides the media content used in the functional process of the service. The Softswitch is the control entity of the network that handles the establishment of the calls made by the service and also allows the interaction of the users with the service through the access and transport networks using mobile or fixed devices.

Following we describe the results and conclusions obtained from implementing the guidelines to develop and deploy the value-added service *Reminder Call* using a JSLEE environment over the telecommunications infrastructure of EMCALI. The results obtained were analyzed in terms of the time used to develop the service through the guidelines proposed in this paper. This time is compared with the average *Time-to-Market* of new services for traditional service providers (9 to 18 months [17]) and with the average *Time-to-Market* of the Nokia-Siemens Networks Service Delivery Framework (18 days [18]). Table 2 shows a comparison between those service development times.

Table 2. Service Development Times.

Developer	Time-to-Market
Traditional Service Providers [16]	9 to 18 months
Nokia-Siemens [17]	18 days
SOA-Based Guidelines for JSLEE	10 days

Source: the authors.

Table 2 shows a very significant reduction in the development time of new value-added services compared to traditional service providers. This is achieved largely due to the use of JSLEE as a platform for the development and execution of services within the proposed guidelines. Traditional service providers make use of several technologies (information systems, communication networks, protocols, etc.) to develop new services which become a very complex and time consuming process.

Such complexity is simplified by JSLEE through its resource adaptor model which makes the developer to focus more on the service logic than in the communications infrastructure.

On the other hand the Nokia-Siemens Networks Service Delivery Framework (SDF) [19] is a highly robust platform for development and execution of value-added services which provides tools that facilitates the development of new services. In table 2 can be appreciated a difference of 8 days between time of service development of Nokia-Siemens SDF and the SOA-based guidelines for JSLEE services. Such difference can be attributed to the use of SOA as a framework for the proposed guidelines which provides an organized way and

promotes the component reuse for new value-added service development.

The guidelines proposed in this paper provide a simple and organized way to develop value-added services on JSLEE environments. Using SOA service creation model as a framework within these guidelines reduces development time of new services which represents great benefits to telecom operators that use JSLEE as a service execution environment.

5 ACKNOWLEDGEMENTS

The authors would like to thank University of Cauca, Colciencias Young Researcher program (Code: 3458) and TelComp2.0 project (Code:

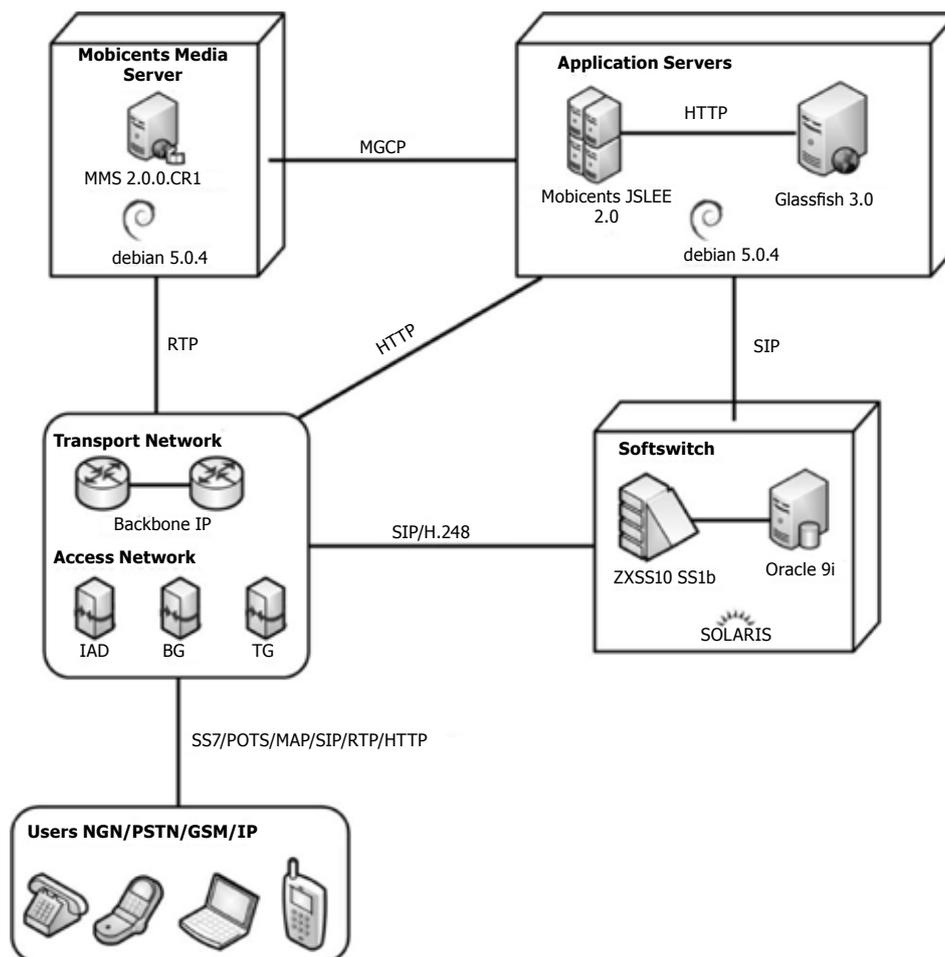


Figure 3. Proposal for Value-added service deployment.

Source: adapted from [16]

1103-521-28338 CT458-2011) for supporting the M. Sc. students Julian Andrés Rojas and Jesus David Ramirez, and EMCALI EICE-ESP for allowing the service deployment over its telecommunication infrastructure.

REFERENCES

- [1] STL Partners, "Telco 2.0 Business Model Innovation for the Digital Economy", [Online], Accessed Nov, 2011. Available: <http://www.stlpartners.com/telco2.php>, 2012.
- [2] Oracle Corporation, "JAIN SLEE API Specification, Java Specification Request (JSR) 22", [Online], Available: <http://jcp.org/en/jsr/detail?id=22>, 2004.
- [3] Oracle Corporation, "JAIN SLEE v1.1, Java Specification Request (JSR) 240", [Online], Available: <http://www.jcp.org/en/jsr/detail?id=240>, 2008.
- [4] S.A. Mohiuddin, "The Business-Technology Drivers and Benefits of SOA", [Online], BPTrends: Business Process Trends, Available: <http://www.bptrends.com/publicationfiles/EIGHT%20BPTrends-BusTechDrivers%26BenfOfSOA-TelcoExecOverview-Final.pdf>, 2007.
- [5] P. Falcarin and C. Venezia, "Communication Web Services and JAIN-SLEE Integration Challenges", *International Journal of Web Services Research*, vol. 5, no. 4, pp. 59-78, 2008.
- [6] P. Falcarin and L. W. Goix, "An Aspect-Oriented Approach for Dynamic Monitoring of a Service Logic Execution Environment", *IEC Annual Review of Communications*, vol. 59, pp. 237-242, 2006.
- [7] U. Wahil et al. *Building SOA Solutions Using the Rational SDP*, 1st ed. Riverton, NJ, US. IBM Redbooks, 2007, 636 p.
- [8] Rational Software Corporation, "Rational Unified Process: Best Practices for Software Development Teams", [Online], Available: http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf, 2005.
- [9] N. Fareghzadeh, "Service Identification Approach to SOA Development", *World Academy of Science Engineering and Technology*, vol. 35, pp. 258-266, 2008.
- [10] Object Management Group OMG, "OMG Unified Modeling Language (OMG UML) Superstructure v 2.4.1", [Online], Available: <http://www.omg.org/spec/UML/2.4.1/>, 2011.
- [11] S. Inaganti and G. K. Behara, "Service Identification: BPM and SOA Handshake", [Online], BPTrends: Business Process Trends, Available: <http://www.bptrends.com/publicationfiles/THREE%2003-07-ART-BPMandSOAHandshake-InagantiBehara-Final.pdf>, 2007.
- [12] Java.Net: The Source for Java Technology Collaboration, "EclipSLEE Project", [Online], Available: <http://java.net/projects/eclipslee>, 2011.
- [13] Alcatel-Lucent, "Alcatel-Lucent SCE-SE Eclipse Plug-In v 1.4.0", [Online], Available: <http://sourceforge.net/projects/sce-se/>, 2007.
- [14] J. Rojas et al, "Entornos de Desarrollo de Libre Distribución para Composición de Servicios JAIN SLEE," presented at 5to Seminario Nacional de Tecnologías Emergentes en Telecomunicaciones y Telemática, Popayán, 2010.
- [15] EMCALI EICE-ESP, [Online]. Accessed Nov, 2011. Available: <http://www.emcali.com.co/web/telecommunications>. 2012.
- [16] G Rojas et al, "Technical Criteria for Value-Added Service Creation, Execution and Deployment, on Next Generation Networks", presented at The Seventh Advanced International Conference on Telecommunications. St. Maarten, The Netherlands Antilles. 2011.
- [17] R. Page et al, "Next Generation Managed Services: Vodafone Adopts a New Operating Model for Service Providers", CISCO IGSG, [Online], Available: <http://www.cisco.com/web/about/ac79/docs/pov/Vodafone.pdf>, 2010.
- [18] Questex Media Group Company, "Philippine telco cuts time to market for new services by 80%", [Online], Accessed Nov, 2011. Available: <http://www.enterpriseinnovation.net/content/philippine-telco-cuts-time-market-new-services-80>, 2009.
- [19] Nokia Siemens Networks Corporation, "Nokia Siemens Networks Service Delivery Framework", [Online], Available: http://www.nokiasiemensnetworks.com/NR/rdonlyres/E452E18D-D288-4C85-A9CB-A80CAD0978BF/0/SDF_A4_2804.pdf, 2008.