International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:1, No:11, 2007

Service Architecture for 3rd Party Operator's Participation

F. Sarabchi, A. H. Darvishan, H. Yeganeh, and H. Ahmadian

***Abstract*—**Next generation networks with the idea of convergence of service and control layer in existing networks (fixed, mobile and data) and with the intention of providing services in an integrated network, has opened new horizon for telecom operators. On the other hand, economic problems have caused operators to look for new source of income including consider new services, subscription of more users and their promotion in using more network resources and easy participation of service providers or 3rd party operators in utilizing networks. With this requirement, an architecture based on next generation objectives for service layer is necessary. In this paper, a new architecture based on IMS model explains participation of 3rd party operators in creation and implementation of services on an integrated telecom network.

***Keywords*—**Service model, IMS, API, Scripting language, JAIN, Parlay.

I. Introduction

A LEGACY telecom network consists of different access networks and each network has its own interface, standards, control and service layer. In some of these networks there is no separation between layers. These unorganized interworking and network discontinuity have caused many problems for operators and network users. From operators' point of view, scalability will be difficult and network expansion will be costly. From users' point of view, network services will only be available to those who are the users of the network and there is no possibility of utilizing other network services.

Also in these legacy networks there is very little possibility for 3rd party participation in providing services for existing users. In fact, the network creators are service providers too.

The idea that service and control layers have similar logical and interoperability behavior in all networks (even though in reality they use different protocols and interfaces), has brought about a new idea in telecom world. The idea based on service, is a concept which limits technology at user access point, and converges different networks in service and control layers. Future telecom world is moving towards network convergence and creation of an integrated infrastructure, independent of user access, with the intention of providing many services [1], [2].

Consequently, the most important effect of this convergence will be reduction in economical pressure created from different service infrastructure. It will also bring about employment opportunities for telecom operators, especially for operators with limited fund, who can operate as a 3rd party operator without incurring costly of infrastructure creation [3].

SDP concept is derived from service convergence idea in telecom. This concept presents architecture for achieving a multi-service media ranging from ordinary telephone services to video/audio conference as well as internet services for all users in different networks. SDP with the help of a standardized frame work creates an environment for implementation and management of different services [5].

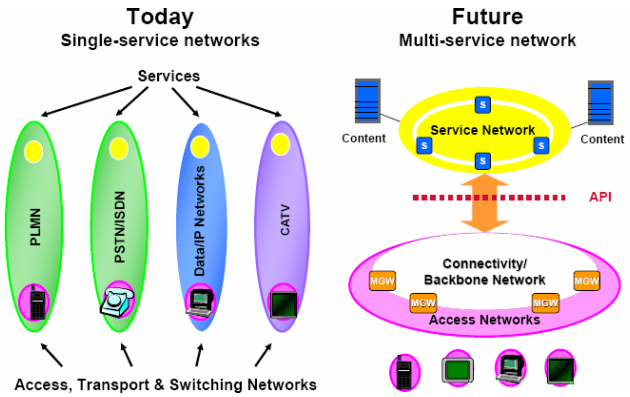

Fig. 1 Network convergence [22]

IMS model is pacing into maturity and many telecom vendors are now supporting standards, interfaces and nodes of this model. We will briefly consider this model in section 2. In section 3, tools required by 3rd party operators for service creation will be looked at. In section 4, the required infrastructure for connection to service layer by 3rd party operators will be considered, and in section 5, an example of SDP for participation of 3rd party operators in providing services in an integrated network based on IMS model will be presented.

II. IMS Architecture

IMS [4] model with presenting all services found in fixed and mobile network on a unified infrastructure, defines a standard architecture for next generation networks. The change of the existing networks to all IP network on which all services and media (Audio, image, music and …) are offered to users utilizing an integrated structure, is the foundation of IMS based networks. The advantages of using IMS model are:

- Independency of network core from access layer technologies
- Mobility at user and service layer
- Quick deployment of new services based on standard structures
- Network structure is based on scalability and redundancy

IMS layer, or better to say, control layer consist of logical nodes for multimedia session setup, call routing, user authentication control and charging. Transport layer consist of logical nodes for different access network connection with IMS layer and service layer (Gateway behavior). Service layer consist of logical nodes for creation and implementation of services using open standards. One valuable advantage of IMS architecture is that it allows reuse of many existing network functions for creating new services. This feature eases the way for 3rd party operator's participation in utilizing network resources for creating and providing new services. This capability is provided by a node in service layer called OSA SCS application server. In the architecture that is presented in this paper, this node will be used as a user interface between network and 3rd party operators. In general, in service layer IMS is used for creation and implementation of three kinds of application server (AS) as described below. Physically, these nodes can be located in one server or many servers to perform their functions.

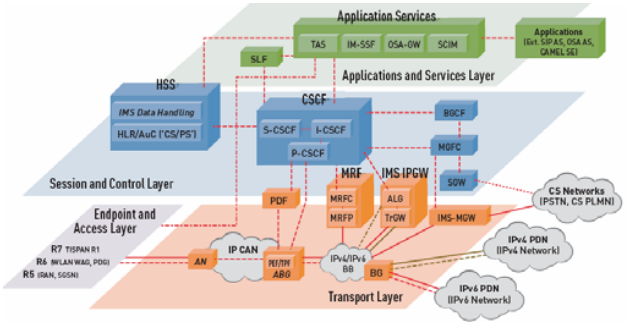

Fig. 2 IMS architecture model [4]

SIP Application Server provides services based on SIP, like messaging and presence. These nodes are usually based by network operators for providing basic services for users.

IM-SSF Application Server provides access to IN services utilizing interactive INAP protocol and access to IN nodes.

OSA SCS Application Server provides service creation by 3rd party operators using interfaces like OSA/Parlay.

In this paper we try to present the service-oriented idea and implementation capability by deployment of OSA SCS application server element, and presenting solutions for how 3rd party operators can make connection with this element, resulting in providing services to different network users who are connected to IMS core.

III. Service Creation Tools

New service Creation capability in service layer, independent of network infrastructure and lower layer technologies, is one important advantage and objective of next generation network. In order to able 3rd party operators to make use of these services, tools for creation and implementation of services have been devised. These tools can be presented in two main methods:

1) Standard Application Programming Interfaces (APIs) that span diverse NGNs, allowing 3rd party application developers to produce new services. API tools define methods and classes with the use of common programming languages and network operators call these methods that are located in 3rd party operator server with the help of distributed software indicated in section 4. In the past few years several industry efforts have emerged to develop such open APIs, including Parlay [6], JAIN [7], and the Open Services Architecture (OSA) [6].
2) Scripting Languages are lightweight, highly customizable, and typically interpreted languages, appropriate for rapid application development. These qualities and features make scripting languages applicable to the field of application programmability next to APIs. Scripting languages represent, in an XML-based file, a description of the service logic. Typically, scripts are created, edited, and validated using regular editors or as a result of applying transformation techniques. For example the Service Creation Mark-up Language (SCML) [14], VoiceXML [12] and Call-Control extensible Markup Language (CCXML) [13] are scripting languages that connect existing components with a particular API, depending on the script file content.

A. API tools

OSA/Parlay: The Open Service Access (OSA)/Parlay [6] defines an architecture that enables the interworking between the IT applications and the telecommunications features through an open standardized interface. It is defined by the Parlay Group, an international consortium of Information Technology and Telecommunications companies.

Parlay specification provides 3rd party application developers with language-independent APIs that allow access to functions and capabilities. Hence, one can create service by defining the process of a service operation with API and execute that service by placing APIs executing codes on the 3rd party operator sever and calling these APIs from network operator server.

The network functionality is described as Service Capability Features (SCFs) and applications could be deployed in a 3rd party domain. SCFs implement groups of Parlay/OSA APIs and provide access to the network capabilities. They are implemented by Service Capability Servers (SCSs) that are logical entities and interact with the network elements. These building blocks are placed on network operator server and a 3rd party operator with the help

of distributed infrastructure like CORBA or WEB Service and with calling these APIs can create its own service and provide that service in the network.

SCSs as network resources, frees 3rd party operators from interaction with network infrastructure. In using this standard, any common programming language can be used for service creation and provision.

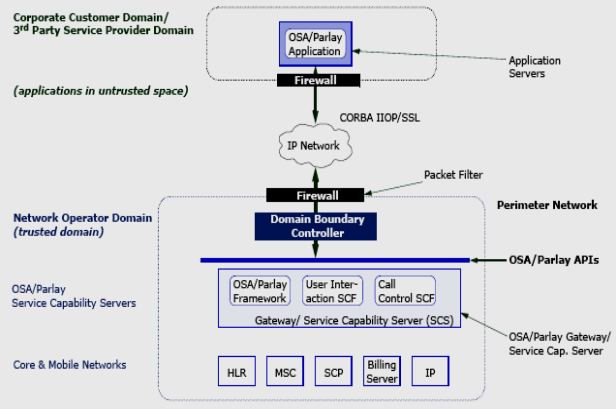

Fig. 3 Parlay Architecture [6]

JAIN [7] is a set of API based on Java, and is used for rapid product development and next generation network services. JAIN also provides a Java based infrastructure on which 3rd party operators and network operators can provide their services in an open environment. The JAIN is similar in many respects to Parlay nonetheless some of the differences are that:

1) JAIN provides APIs not only at the functions layer but also at the protocols layer.
2) In JAIN all the APIs are specified in Java.
3) JAIN clearly defines a Service Logic Execution Environment (SLEE).
4) JAIN defines a service creation environment (SCE).

JAIN provides service portability, independence from network, open and secure deployment of telephone, data and wireless network resources. JAIN supports a set of API at different network levels including JAIN SIP, JAIN IN… call control and JCC/CAT [8]. These API are utilized for creation, handling and terminating of sessions with 3rd party JAIN.

In JAIN there is also an environment for execution and management of services called JSLEE. This part acts as a software platform for network operator server. In this environment, service building blocks with the name of Service Building Block (SBB) are implemented and 3rd party operators with the help of JAIN-SPA can have access to these blocks. Note that SBB is the same as SCS is parlay.

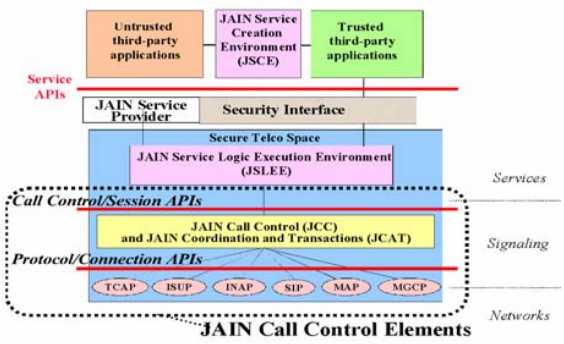

Fig. 4 JAIN Architecture [7]

B. Script Tools

CCXML/Voice XML: In general voiceXML is used as a technology allows a user to interact with a web server through voice-recognition technology, which exploits Media Server capabilities. One dynamic factor in providing voice XML is its responsive capability to IVR users. VoiceXML framework is based on HTTP protocol and is related with Internet and telephone network. In this system, the user is confronted to some resources like pre-recorded audio, text-to-speech synthesis (TTS), Automatic Speech Recognition (ASR) and DTMF. VoiceXML is a high-level abstraction language and this means that developers with little training can use it. VoiceXML makes it easy to rapidly create new applications and protect developers from low level programming issues.

VoiceXML provides very little in terms of call control, besides CCXML by providing advanced telephony operation like conference control, click-to-dial and … is a supplement for voice XML. CCXML has been designed to complement and integrate with a VoiceXML system. CCXML is used to overcome the weak point of voice XML in handling call control.

SCML is language was designed by JAIN engineering group with the intention of service creation in next generation networks. Even though current engineering group endeavor is concentrated on call control capabilities, the objective of SCML designers is based on providing services like mobility, multiparty, multimedia and etc.

SCML server is based on JCC programmable interface and is therefore independent from network protocol and signaling layer. This language hides network technology from service creator and provides basic events managements so that call control can be implemented. In short, once SCML server receives a service request via JCC, it performs service logic and sends the appropriate message to network control layer (with the help of network protocol) via JCC.

IV. Distributed Software Creation Tools

The objective of presenting distributed processing technologies is to provide means and standards required for production and utilization of components for distributed

software creation. With these in hand, utilization of services in local or remote distributed software will be provided and in consequence 3rd party operators can interact with network users based on service and independent of network infrastructure. Below three relevant technologies are considered:

1) **RMI Technology:** RMI [15] provides Java programmer with mechanisms so that they can execute different object methods on a remote virtual machine (JVM). Software objects can completely be passed or returned like a parameter. This feature means that a Java programmer with the help of RMI, can transfer new codes in the network and execute them in remote virtual machine in a dynamic way. This unique feature has allowed this technology to be used in distributed system creation. In a distributed environment, RMI clients can have access to new version of Java services and there is no need to distribute programs to clients.
2) **CORBA Technology:** This technology provides the means to call remote software object methods and transfer various simple data type or data structures consisting of simple data type. CORBA [16] services are defined via an interface written with IDL. Different IDL for various programming languages have been devised and unlike RMI, this technology is not limited to a specific language and with it many distributed software systems in distributed environments can be related. CORBA technology because of its easy usage, flexibility, portability (in different languages) and speed in real time applications has become popular among programmers for linking different software platforms for a system development.
3) **COM Technology:** After introducing different technologies by Microsoft like OLE (Object Linking + Embedding) for linking various MS-office documents and OLE2 for linking all windows supported documents, COM technology [17] was introduced by Microsoft. At first, this technology with respect to distributed components did not offer a substantial capability. But with presenting of Win-95 and Win-NT and necessity of distributed components and the relation between these components, Distributed COM (DCOM) and its improved version with the name of COM+ was introduced. Both DCOM and CORBA work based on client/server model for their communication between components and therefore have similar behavior. DCOM compared with other technologies have less popularity and is mostly used in .NET platforms.
4) **WEB Service Technology:** Similar to other technologies, WEB Service [18] is utilized for distributed applications, with one special and fundamental difference that its infrastructure and functionality is based on Internet application. This feature has made this technology very popular and widely used. This technology with the help of common web based standards and protocols, like HTTP and XML, provide the capability to call services located on various servers. Currently this technology in many SDP architectures is used as a key interface for connection with 3rd party operator servers. Similarity, Parlay X standard based on this technology for telecom service creation has been devised. Other advantage of using this technology is that it supports software platforms like J2EE and .NET.

V. 3rd Party Operator Service Model

Nowadays, moving towards next generation networks is in its initial phase. Telecom operators are devising methods and state of developments of various layers for easy migration. Therefore a unified and an accurate reference can not be found which explains all layers of network. Ambiguity in service layer and new concepts is becoming more colorful because of lack of experience with these entities. As indicated before, Participation of 3rd party operators in creating and presenting services upon an integrated infrastructure is the objective of this layer. In the diagram below, firstly a general model for connection between 3rd party operators and network based on current platforms is presented and then more descriptions will be given.

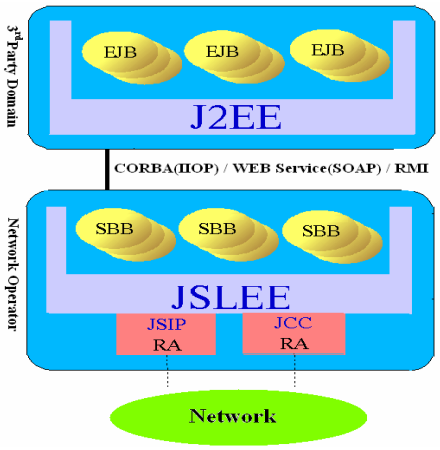

Fig. 5 General Service model for relation of 3rd party operator and network capability [7]

JSLEE [19] is a platform based on Java and is used as providing resource networks (SBB) for 3rd party operators and is located on network operator server (OSA SCS Application Server). The kinds of SBBs that are installed in this server are named as service building blocks in parlay standards (SCS) and are implemented as Java classes. As parlay standard is open, any 3rd party operator with the help of these APIs, can use block functionalities for own service creation. Also in this server, with using JAIN and protocol relevant APIs, required network connections can be setup. A suitable platform for 3rd party operators is J2EE. This platform is known as the best application server these days. Service logic is implemented by 3rd party operator as EJB and installed in J2EE [20]. The existence of distributed software

mechanisms in this platform, guarantees the connection between EJBs (Service Logic) and SBBs (Service Building Blocks Provided by Network Operators). 3rd party operators with the help of OSA SCS application server (network operator server) which resides on IMS core, makes connections with network and users. More explanation on 3rd party operator server with OSA SCS application server will be presented.

In order to implement the mentioned model, the first step is to provide IP connection between servers. This connection can be created by the network operator with the creation of a private IP network which guarantees network security and quality of service. In case existing IP network is used, this connection should be set up utilizing common methods like VPN, L2/L3 MPLS VPN, VPLS and IP SEC. Note that the way IP connections between 3rd party severs and network are made, has no effect on overall model which is concentrated on service presentation. What is important is the existence of a secure IP connection between servers with a good quality of service.

As it is indicated in section 3 of this paper, there are two methods for creating services based on API and script. Therefore this model should have compatibility with any 3rd party operator or any method. Those operators who use API tools need a CORBA, RMI or WEB service server so that they can call remote methods. As it is seen in fig 6, service logic, service execution mechanism consist of sessions in the network for service presentation, is located on 3rd party operator server and network operator by calling standard APIs on its own server and mapping service to API.

Network operator by calling standard APIs on its own server, maps services to APIs via using JSLEE profiles which causes the service logic to be performed on the 3rd party operator server. Results are then returned to network operator server with the help of network adoption block, which is the same as JAIN protocol API, and results are then converted to network sessions. CORBA service provider provides above mentioned mechanism for parlay API and RMI service provider provides it for JAIN API. Web service is also used in connection with parlay X. Software platforms based on J2EE technology and with support for all three service provider, are the best option for the implementation of this mechanism.

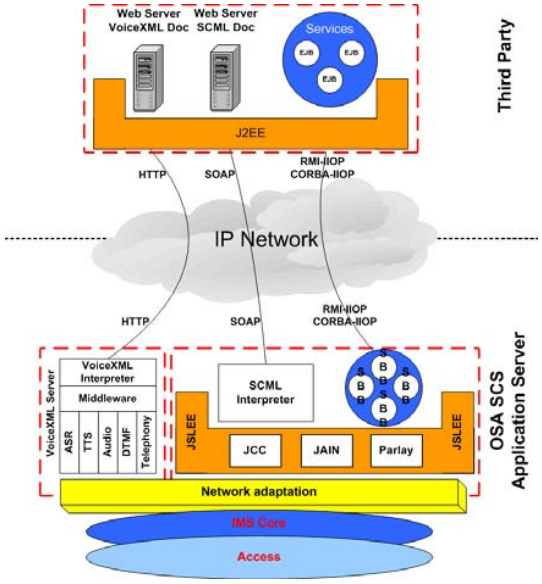

Fig. 6 Service model for 3rd party partnership

3rd party operators, who utilize scripts for providing services, have a simpler mechanism. These operators provide their services based on XML format and whenever the network user request for such a service, its XML document is provided to network operator. This mechanism is very similar to processes that are performed in Internet. SCML interpreter and voice XML/CCXML play the same role in network operator as Internet Explorer does for Internet. When specific service (with a specific URL) is requested by a user, SCML or voiceXML/CCXML server with the help of HTTP or SOAP protocol, receive the relevant XML document or message from web service provider and by interpreting that request, server the user.

In voiceXML/CCXML technology, 3rd party operators write the required service text with common editors and deploy it on their own web server and when the network operator makes a request for that, the relevant document is sent for the network operator and the existing voiceXML/CCXML interpreter on network server will interpret and with the help of basic service blocks like speech recognition, sound file recording, DTMF and etc. will provide the required service to the user. Network operator server will play voiceXML gateway for service creation and execution utilizing this mechanism. VoiceXML gateway is usually located on a separate OSA SCS application server.

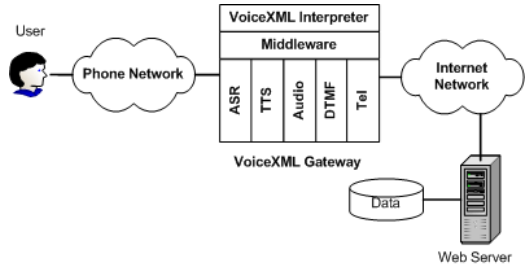

Fig. 7 VoiceXML System Architecture

In SCML technology, as shown in Fig 8, there are 3 different ways for providing services. There are three possibilities here: 1) the SCML interpreter could reside on the 3rd party server itself, 2) the SCML interpreter could reside on an Application Server, or 3) the SCML interpreter could reside on both sides.

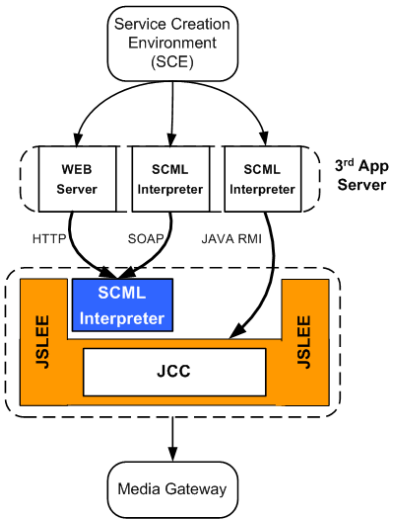

Fig. 8 SCML System Architecture

In case 1, the required service will be interpreted and executed on the 3rd party operator server and SCML interpreter makes remote calls to the JCC interface (e.g. using Java RMI). In the case 2 in which only SCML interpreter resides on network operator server, SCML document is received and executed by network operator with the help of HTTP protocol. In the case 3, web service will provide the means for service execution. In this case, 3rd party SCML interpreter send and XML message via SOAP [21] protocol for network SCML interpreter and then XML document is received and executed by network SCML interpreter.

VI. Conclusion

In this paper a model for relating different 3rd party operators to network operators was put forward and needed requirements for making connections and service creation between these operators was revealed. From a scientific point of view this model is regarded as a flexible and easy model for implementation because it uses common mechanisms and platforms like HTTP protocols, SOAP and J2EE which is highly used by service provider these days. Other advantages of this model are its scalability and redundancy. In this model, 3rd party operators can use text or API technology to create and provide their own services without any limitations. In this model, there is no limitation to the number of 3rd party operators involved, and the only prerequisite for the presence of 3rd party operators in this model, is the existence of a secure IP connection between the servers.

References

[1] Byungsun Lee, et al, *Next-generation. Open Network. Forum. Report*, NONF, Dec, 2001.
[2] Hanwook. Chung, *Trend. and Tech. of Wire/Wireless. Convergence. Service, Institude of Information. Technology. Assessment.* Journal. vol. 13, 2002.
[3] Chunghun Choi, Youngmee Shin, et al, "Design and Implementation of the Integrated Service Creation Environments", Korea Institute of Communication Sciences Journal vol. ",pp. 198-202, 1996
[4] M. Poikselka, A. Niemi, H. Khartabil, G. Mayer, *The IMS: IP Multimedia Concepts and Services*, 2nd Edition, 2006.
[5] N. Silva, M. Monteiro, S. Rego, H. Jorge, *Service Oriented Architectures for convergent Service Delivery Platform*s, P1652, 2006.
[6] Parlay Group, (2005), A Guide to the Parlay Specifications, [Online]. Available: www.parlay.com
[7] Sun Microsystem, JAIN and Java in Communication, 2004, [Online]. Available: http://java.sun.com/products/jain
[8] SunMicrosystem, (2001), JAIN JCC Specification, [Online]. Available: http://jcp.org/en/jsr/detail?id=21
[9] SunMicrosystem, JAIN SLEE API Specification, [Online]. Available: http://jcp.org/en/jsr/detail?id=22
[10] Sun Microsystem, (2002), the JAIN API:Integrated Network API for Java Platform, [Online]. Available: http://java.sun.com/products/jain
[11] Paolo falcarin,carlo alberto licciardi , Analysis of NGN Service Creation Technologies, Eurescom Project P1109, 2005.
[12] Scott McGlashan, Daniel C. Burnett, Jerry Carter, et al., Voice Extensible Markup Language (VoiceXML) Version 2.0, W3C Recommendation, 2004
[13] RJ Auburn, Voxeo, Voice Browser Call Control: CCXML Version 1.0, W3C Working Draft, 2007.
[14] Bakker, J.L., Jain, R., Next Generation Service Creation Using XML Scripting Languages, 2002.
[15] Java Remote Method Invocation (Java RMI), Available: http://java.sun.com/javase/6/docs/technotes/guides/rmi/index
[16] Common Object Request Broker Architecture (CORBA) specification, www.corba.org.
[17] A. Alessandra, D. McCartney, (2001), Next Generation Networks: the service offering Standpoint, P1109, [Online]. http://www.w3.org/2002/ws/
[18] JAIN SLEE Implementation Experiences, JNETX Communication & Commerce, 2005
[19] SunMicrosystem, (2005), J2EE and Telecom, [Online]. Available: http://java.sun.com/products/jain
[20] W3C, (2007), Simple Object Access Protocol (SOAP) 1.2 specification, [Online]. Available: www.w3.org/TR/SOAP
[21] T. Megedanz, Tutorial about SOA concepts in Telecommunications, 2007.