

OSA ParlayX Gateway Architecture for 3rd Party Operators Participation in Next Generation Networks

A.H Darvishan , H.Yeganeh, K.Bamasian, H.Ahmadian
Iran Telecommunication Research Center (ITRC), Iran
darvish@itrc.ac.ir, yeganeh@itrc.ac.ir , kbamasi@itrc.ac.ir, ahmadian@itrc.ac.ir

Abstract—In this paper a practical architecture for creation of OSA ParlayX Gateway in service layer of an integrated networks, according to IMS architecture will be presented. OSA ParlayX Gateway is a suitable platform for third party operators participation to provide integrated services in NGN. Third parties without any knowledge of network technologies, by using standard ParlayX APIs on top of Web Services platform which provided by network operators on OSA ParlayX Gateway node, can create new services. In this paper a layered software architecture using .Net framework for performing ParlayX transactions on network sessions (using SIP protocol) will be presented. This architecture provides faster implementation of services for third parties and high scalability of network.

Keywords—IMS, ParlayX, 3rdParty, WEB Service, SOAP

I. INTRODUCTION

Next generation networks with the intention of converging existing mobile-networks, data and PSTN-in service and control layer, with the aim of “Equal service presentation in integrated networks”, have provided a new approach to telecom operators. The idea of similar logical and functional behavior in control and service layers of different networks (despite use of different protocols and interfaces in reality), has created a new perspective in telecom world. Service based perspective, is an idea which limits technology at user access point and converges different networks at service and control layer [1][2]. Therefore telecom world, independent of diversity in access type and with the intention of telecom networks evolution is moving towards convergence and creation of an integrated platform for receiving service. Consequently the most important effect of convergence is reduction of economical pressures that originated from creation of different platforms and also creation of Job opportunities for operators, especially low income operators, who find business opportunities as third party operators without creating costly telecom platforms.

ParlayX by providing APIs at different levels of Call Control on Web Services platform provides a suitable opportunity for third party operators that they can participate in providing services for NGN. Third party operators without any knowledge about network or Call Control protocols which used in a network, can implement their services and execute

them via OSA ParlayX Gateway node by concentrating on service software or executing SDL(usually called service logic). OSA ParlayX Gateway converts Web Service transactions from third party operators, to network sessions among network users. OSA ParlayX Gateway provides a simple platform for new service creation and presentation by creating a single layer between network and service providers[3].

In section 2, position of OSA ParlayX Gateway in next generation network model- concentrating on IMS model- will be considered. Then, ParlayX standard will be described. After that, a software architecture based on .Net framework with quick and simple implementation capability will be introduced and finally, some samples of service implementations that using this software architecture will be presented.

II. IMS-TISPAN MODEL

IMS [2] introduces a standard structure for next generation telecom networks, and intends to present all services of PSTN’s services and mobile networks on a unified platform.. The idea of IMS requires conversion of all existing networks into a single all IP network, in which all services and media (voice, video, music,...) are provided to users in an integrated structure. The most important features of IMS networks are:

- The network core is independent from access layer technology.
- IMS network is responsive at user and service layer
- IMS network can create new services based on standard structures quickly.
- IMS has a structure based on redundancy and scalability.

IMS layer, or better defined the control layer of IMS network, consists of logical nodes for setting up multimedia sessions, routing calls, authentication and charging. Control layer consists of logical nodes for setting up connections among different access networks with IMS layer and service layer. Service layer consists of logical nodes for creation and execution of services that using open standards. One of the most valuable features of IMS architecture is its reusability of many existing functions in the network to present new services.

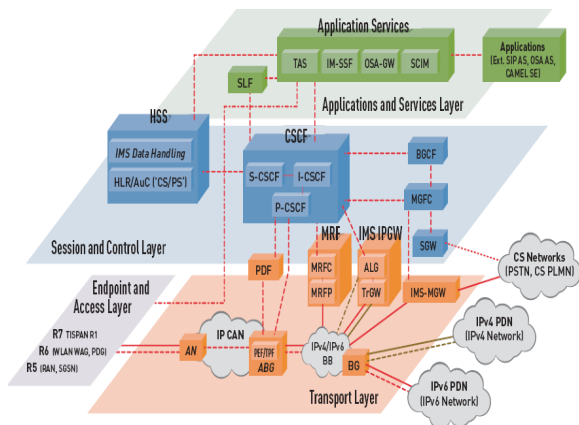


Figure 1: IMS Model [2]

In fact by this feature, a third party operator has a capability for using network resources to creation and presentation of services. This capability is provided by a node called OSA SCS Application Server in service layer. In the architecture presented in this paper, this node is used as user interface between network and third party operator. In general, in IMS service layer three types of Application Server (AS) are defined for creating and executing services. These nodes can be physically placed in one or different servers. These ASs are:

SIP Application Server: Provides SIP based services, e.g. Messaging and Presence. This node is usually used by network operator and presents basic services to users.

IN-SSF Application Server: Provides access to IN services through INAP protocol and with intelligent network nodes.

OSA-SCS Application Server: Provides opportunity for service creation by third party operators with supporting interfaces like OSA/Parlay.

3GPP, ETSI, and Parlay group has composed and selected ParlayX standard for OSA-SCS Application Server. In fact, in IMS model, third party operators present their services in network by using ParlayX standard on Web Service platform. OSA-SCS Application Server with ParlayX standard is called OSA ParlayX Gateway and its position is shown in Fig2. In

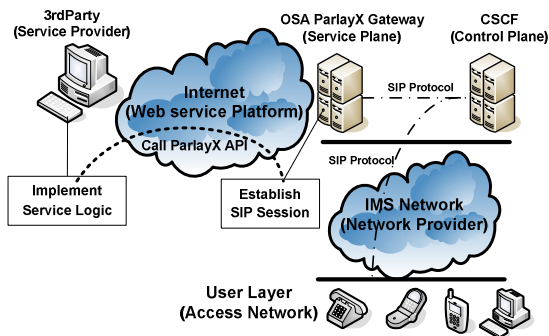


Figure 2: OSA ParlayX GW

section 4 of this paper, practical software architecture for this node is presented.

As it is shown, service providers independently implement service logic or Service SDL. Each service is a collection of network sessions between different users, and service logic is

a description of these sessions for a service. Session setup is the responsibility of OSA ParlayX Gateway, and Service Providers present their required service sessions to OSA ParlayX Gateway. OSA ParlayX Gateway by using SIP protocol and via CSCF node setup sessions with users. In the next part, we will explain the ParlayX standard.

III. PARLAYX STANDARD

One of the most important objectives of next generation networks is the ability of new service creation in service layer independent of lower layers or network infrastructures. Therefore, for the use of third party operators, tools for creating, specifying and executing of created services have been devised. The most important of all is ParlayX standard which has been standardized for service layer in next generation networks by parlay group, ETSI and 3GPP. In this standard, by defining different APIs for network session setup and termination, a platform is created to implement service logic irrespective of session setups by network operators. Software implementation of these APIs is done by network operators which using standard programming languages in the form of Methods and Classes. Third party operators, then call these Methods in their service logic by using Web Service and distributed software infrastructures which used in ParlayX. So the necessary sessions for their services are created. Using this method clearly shows that third party operators don't need any knowledge about network and should only issue session setup and disconnect request to their network operators[3].

In ParlayX standard, a collection of different APIs in the form of Call Control, SMS Handling, Call Handling, and so on have been defined, and third party operators can use them to perform their network interactions with their users. These APIs have been devised by joint cooperation of ETSI, 3GPP and Parlay in a standard collection called ETSI TS 129 [4][5]. A few of these standard formats like Third party call (ETSI TS 29 199-02) for setting up calls among users, Call Notification (ETSI TS 29 199-03) for reporting call status to third party operator, Short Messaging (ETSI TS 29 199-04) for sending and receiving of SMS are shown in Fig 3. Software libraries for these APIs are implemented by network operator on OSA ParlayX Gateway node. For using this standard in service creation and presentation, any standard programming language may be used. Web Service [7][8][9] uses HTTP and SOAP protocol- a language based on XML- for creating distributed softwares and therefore has many structural similarities with Internet mechanisms. This characteristic is a reason for deployment and popularity of this technology and easy usage of it. Web Service technology by using of Internet standards and protocols, like HTTP and XML, provides service call capabilities on different servers, and in many service architectures is considered as a key interface for connection with third party operator servers. Web Service and standard set of ParlayX are suitable means for IT operators to participating in creating and presenting of Telecom services.

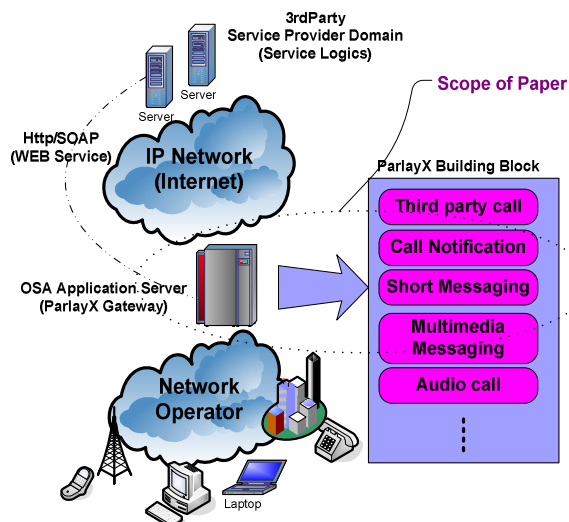


Figure 3: Standard ParlayX

IV. PRESENTATION OF PARLAYX OSA GATEWAY SOFTWARE ARCHITECTURE

Figure 4 shows an architecture for OSA ParlayX Gateway software implementation. This implementation is based on c#

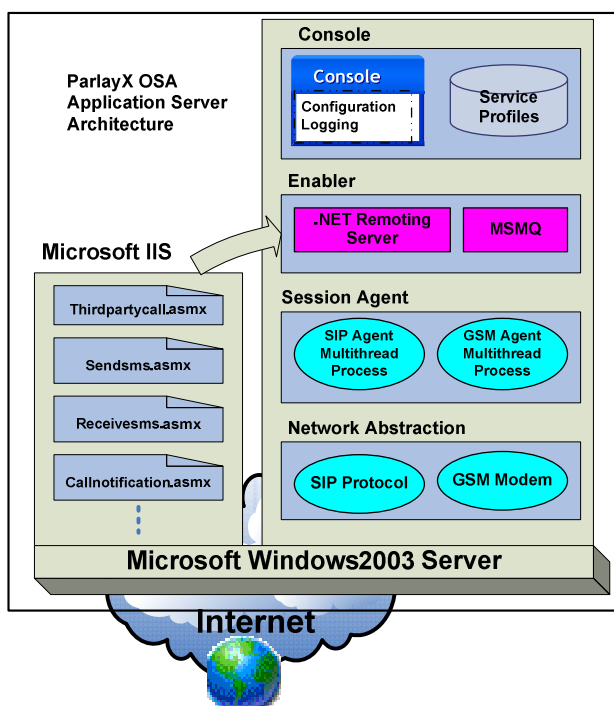


Figure 4: OSA ParlayX Gateway architecture

programming language and Microsoft .Net environment. In this architecture the latest software technologies that are introduced in .Net have been devised. The utilized OS is Window-2003.

The software architecture consists of two parts.

a) Microsoft IIS

IIS is an application that responds to Internet request-requests based on HTTP protocol. This application acts as Application Server inside the windows operating system. Naturally utilizing this software as HTTP service provider can perform a Web Service tasks which in turn shortens implementation time and justifies final product from reliability point of view. It should be mentioned that installing Microsoft .Net version 3 with IIS version 6 which is only found in Win-2003, it is possible to support Web Services (meaning it can respond to SOAP calls).

b) Console Program

Console Program consists of softwares like system monitoring and logging, configuration, service information database, Enablers for connection to Published ParlayX documents in IIS and software agents for network interactions and network protocols. Enablers with the use of Remoting and MSMQ mechanism-devised in Microsoft .Net- process received requests from IIS. In documents which are published in IIS, ParlayX APIs are implemented therefore executing softwares related to these APIs are implemented in console program and session agents. Enablers provide the connection between these APIs and session agents. Session agents process network requests that are requested by Third party operators via ParlayX APIs in separate threads, and make connection with network by using protocols in Network Abstraction part. In general, Enabler and session agent are standard ParlayX libraries which are performed in OSA ParlayX Gateway element. In Fig 5, different inter-node protocol connections in network are shown. Third party operators make connection with IIS via Internet using Web Service mechanism (HTTP/SOAP Protocol).

IIS by interpreting SOAP requests which consists of calling ParlayX APIs, transfers requests to Enabler in Console Program using Remoting mechanism with TCP protocol, via that, SIP [10] requests for call setup or connection are transferred to SIP Server of VoIP network. In the next part, implementation of Wakeup and Click-to- Dial service by a Third party operator and protocol exchange between different nodes are presented.

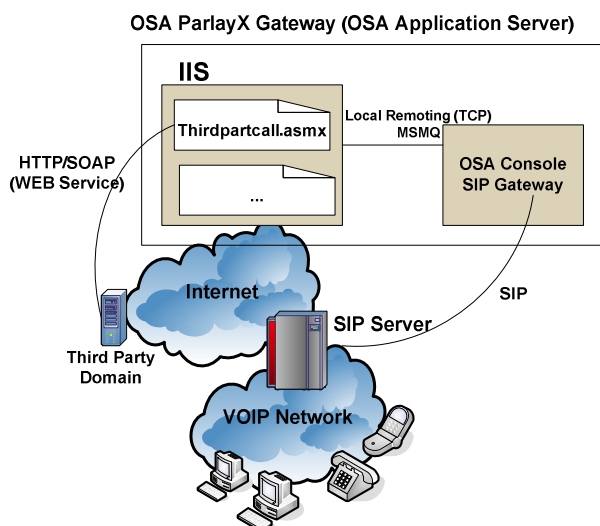


Figure 5: OSA ParlayX Gateway Protocol exchange

V. SERVICE IMPLEMENTATION USING DEvised ARCHITECTURE

From a third party operator's view utilizing the devised software, two telecom services, Wakeup and Click-to-Dial, should be considered. Protocol connections and scenarios for both services will be presented. In both of these services Third Party Call APIs (ETSI TS 129 199-02) shall be used. It is worth mentioning that implementation of Wakeup Service logic is performed in third party operator server and implementation of network interactions is performed in OSA ParlayX Gateway.

a) Wakeup

In this service and according to user request, a connection is established from network to user at the specified time. This call reacts like an Alarm. Users usually use web to register themselves and the required time in third party server. The APIs that are used by third party operators to implement this service are:

MakeCallSession[5]: This API is used among users to create Call Sessions. In Wakeup Service, third party operators by calling this API at the specified time via web service, requests call setup from network to service user. The input parameter for this API is service user address and API output is an ID for consequent call status.

GetCallSessionInformation[5]: This API is used to trace call status. In Wakeup Service, In case that user pickup the phone or does not pickup phone after a time elapse (during time elapse user phone is ringed many times), then the call has to be terminated. Third party operator periodically calls this API to ascertain call status and act accordingly. The input parameter for this API is the received ID from MakeCallSession which is an indication of the call. API output will show the call status.

EndCallSession[5]: This API is used to terminate a session.

The input parameter to this API is the received ID from MakeCallSession.

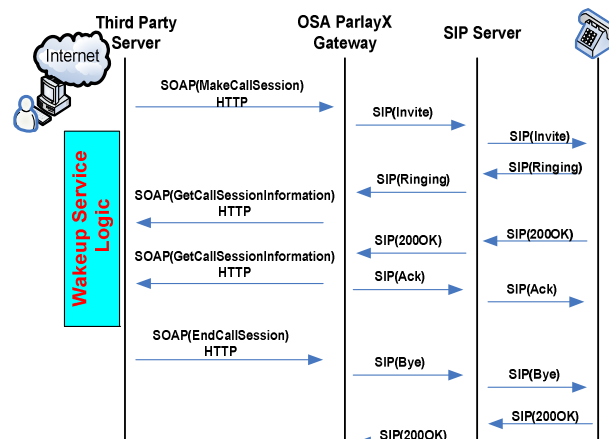


Figure 6: Wakeup Sessions

In Fig 6 a scenario for creation of Wakeup Service and its protocol exchanges are shown. The user can refer to third party operator web site and pay for charges and then register himself and configure Wakeup Service according to require time. At the specified time, Wakeup Service logic of third party operator server will call MakeCallSession API. Calling this API via HTTP protocol and SOAP whose descriptions is presented in Fig 7, will provide its request to OSA ParlayX Gateway.

```
Hypertext Transfer Protocol
POST /Thirdpartcall.aspx HTTP/1.1\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client
Protocol
Content-Type: text/xml; charset=utf-8\r\n
SOAPAction: "ThirdPartyCall/MakeCallSession"\r\n
Host: 10.1.40.3\r\n
Content-Length: 346
Expect: 100-continue\r\n
Connection: Keep-Alive\r\n
eXtensible Markup Language
xml?>
"version="1.0
encoding="utf-8"
MakeCallSession Invite
<?
soap:Envelope>
"/xmlns:soap="http://schemas.xmlsoap.org/soap/envelope
"xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
<"xmlns:xsd="http://www.w3.org/2001/XMLSchema
<soap:Body>
MakeCallSession>
<"xmlns="ThirdPartyCall
<UserID>
<string>
84274200
</string>
</UserID>
MakeCallSession/>
</soap:Body/>
</soap:Envelope/>
```

a) MakeCallSession Request (HTTP/SOAP)

```

eXtensible Markup Language
xml?>
"version="1.0
"encoding="utf-8
<?
soap:Envelope>
"/xmlns:soap="http://schemas.xmlsoap.org/soap/envelope
"xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
<"xmlns:xsd="http://www.w3.org/2001/XMLSchema
<soap:Body>
MakeCallSessionResponse>
<"xmlns="ThirdPartyCall
<MakeCallSessionResult>
1
<MakeCallSessionResult/>
<MakeCallSessionResponse/>
<soap:Body/>
<soap:Envelope/>

```

b) MakeCallSession Response (HTTP/SOAP)

```

Session Initiation Protocol
Request-Line: INVITE sip:84274200@80.191.28.172 SIP/2.0
Method: INVITE
Resent Packet: False
Message Header
Via: SIP/2.0/UDP 10.1.40.3:1272;branch=z9hG4bK30903763
Transport: UDP
Sent-by Address: 10.1.40.3
Sent-by port: 1272
Branch: z9hG4bK30903763
Max-Forwards: 70
From: <sip:84274000@80.191.28.172>;tag=1641070879
SIP from address: sip:84274000@80.191.28.172
SIP tag: 1641070879
To: <sip:84274200@80.191.28.172>
SIP to address: sip:84274200@80.191.28.172
Call-ID: 605020959-141530256-1063807152
CSeq: 1 INVITE
Sequence Number: 1
Method: INVITE
Contact: <sip:84274000@10.1.40.3:1272>
Contact Binding: <sip:84274000@10.1.40.3:1272>
URI: <sip:84274000@10.1.40.3:1272>

```

c) SIP Transaction

Figure 7: Wakeup Service

In OSA ParlayX Gateway, IIS software receives HTTP message and after processing SOAP body inside the HTTP message, calls MakeCallSession API. The Remoting mechanism lodged inside the API, calls the Enabler implemented in console and by creating a thread related to SIP session, an Invite message is sent to SIP server. The SIP server will then setup the required connection with the service user. During this session setup, third party operator can follow call status by calling GetCallSessionInformation API. At the end by picking up the handset and receiving 200k message from OSA ParlayX Gateway, conversation phase is setup and third party operator can realize this by continually calling GetCallSessionInformation API. In this state third party operator can request call termination by calling EndCallSession API from OSA ParlayX Gateway and this request is executed by sending SIP protocol: Bye message.

b) Click to Dial

In this service the user refers to the third party operator's web site and by registering his customers, a request for call setup among them is made. In this service, a call connection among many users via Internet can be setup. Fig 8 shows the scenario for this service. Similar to Wakeup Service, third

party operator can request call connection among customers by calling MakeCallSession API from OSA ParlayX Gateway. OSA ParlayX Gateway will also setup the necessary sessions via SIP protocol.

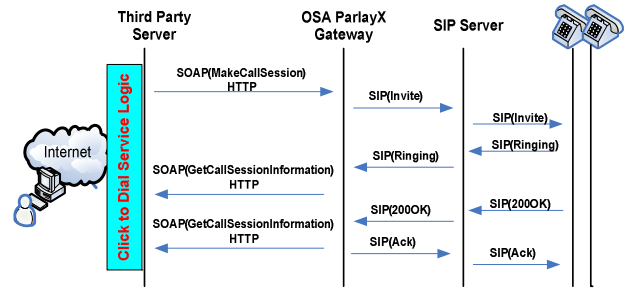


Figure 8: Scenario for Click to Dial

```

Hypertext Transfer Protocol
POST /ThirdPartyCall.aspx HTTP/1.1\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client
Protocol 2.0.50727.1433)\r\n
Content-Type: text/xml; charset=utf-8\r\n
SOAPAction: "ThirdPartyCall/MakeCallSession"\r\n
Host: 10.1.40.3\r\n
Content-Length: 399
Expect: 100-continue\r\n
Connection: Keep-Alive\r\n
eXtensible Markup Language
xml?>
"version="1.0
"encoding="utf-8
<?
soap:Envelope>
"/xmlns:soap="http://schemas.xmlsoap.org/soap/envelope
"xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
<"xmlns:xsd="http://www.w3.org/2001/XMLSchema
<soap:Body>
MakeCallSession>
<"xmlns="ThirdPartyCall
<UserID>
<string>
84274200@80.191.28.172
</string>
<string>
84274201@80.191.28.172
</string>
</UserID>
<MakeCallSession/>
<soap:Body/>
<soap:Envelope/>

```

a) MakeCallSession Request (HTTP/SOAP)

```

Session Initiation Protocol
Request-Line: INVITE sip:84274200@80.191.28.172@80.191.28.172
SIP/2.0
Method: INVITE
Resent Packet: False
Message Header
Via: SIP/2.0/UDP 10.1.40.3:1275;branch=z9hG4bK1826673897
Transport: UDP
Sent-by Address: 10.1.40.3
Sent-by port: 1275
Branch: z9hG4bK1826673897
Max-Forwards: 70
From: <sip:84274000@80.191.28.172>;tag=697883054
SIP from address: sip:84274000@80.191.28.172
SIP tag: 697883054
To: <sip:84274200@80.191.28.172@80.191.28.172>

Session Initiation Protocol
Request-Line: INVITE sip:84274201@80.191.28.172@80.191.28.172
SIP/2.0
Method: INVITE
Resent Packet: False
Message Header
Via: SIP/2.0/UDP 10.1.40.3:1275;branch=z9hG4bK2028508007
Transport: UDP
Sent-by Address: 10.1.40.3
Sent-by port: 1275
Branch: z9hG4bK2028508007
Max-Forwards: 70
From: <sip:84274000@80.191.28.172>;tag=229849727
SIP from address: sip:84274000@80.191.28.172
SIP tag: 229849727
To: <sip:84274201@80.191.28.172@80.191.28.172>

```

b) SIP Transaction

Figure 9: Click to Dial Service

VI. CONCLUSION

The mentioned architecture is implemented in ITRC NGN pilot and can offer various services like Wakeup, Email to SMS, call notification by Busy, No Answer,... . Utilizing Microsoft IIS for Web Service support, will reduce implementation time and will provide reliability to software collection. using of Remoting and MSMQ not only free IIS resources to faster response to HTTP protocol messages but also integrate processes needed for session setup activated by calling different ParlayX APIs.

REFERENCES

- [1] N. Silva, M. Monteiro, S. Rego, H. Jorge, *Service Oriented Architectures for convergent Service Delivery Platforms*, P1652, 2006.
- [2] M. Poikselka, A. Niemi, H. Khartabil, G. Mayer, *The IMS: IP Multimedia Concepts and Services, 2nd Edition*, 2006.
- [3] Parlay Group, *A Guide to the Parlay Specifications*, 2005, <http://www.parlay.com>
- [4] ETSI TS 129 199-1 V7.2.0 (2007-06), *Open Service Access (OSA);Parlay X web services;Part 1: Common*
- [5] ETSI TS 129 199-2 V7.4.0 (2007-06), *Open Service Access (OSA);Parlay X web services;Part 2:Third Party Call*
- [6] Paolo falcarin,carlo alberto licciardi , *Analysis of NGN Service Creation Technologies*, Eurescom Project P1109, 2005.
- [7] <http://www.w3.org/2002/ws/>
- [8] W3C, *Simple Object Access Protocol (SOAP) 1.2 specification*, 2007, <http://www.w3.org/TR/SOAP>
- [9] T. Megedanz, *Tutorial about SOA concepts in Telecommunications*, 2007
- [10] RFC3261, *SIP Protocol*, IETF 2002
- [11] T. Megedanz, *Tutorial about SOA concepts in Telecommunications*, 2007
- [12] Sun Microsystems, *JAIN and Java in Communication*, 2004, <http://java.sun.com/products/jain>