AN AGILE SERVICE PLATFORM FOR TELECOMMUNICATION ENVIRONMENTS

Lajos Lange[1], Alexander Blotny[1], Thomas Magedanz[1]

[1]Fraunhofer Institute FOKUS, Kaiserin-Augusta Allee 31, 10589 Berlin, Germany
Email: {Lajos.Lange|Alexander.Blotny|Thomas.Magedanz}@fokus.fraunhofer.de

Abstract — A framework for lowering the borders of service delivery and service composition in telecommunication networks is proposed in this work. In our opinion such an agile framework is needed for two main reasons. At first the cycle of introducing new services in the telecommunication industry usually takes several months, while the general cycles in the Web domain are much faster. Besides, customers want to use a flexible amount of services from different kinds of sources (e.g. Google, Facebook). The proposed agile service environment is intended to interact with services of the traditional telecommunication industry and is enabling the third party development process to support services coming from all kind of domains. Service platforms for Next Generation Networks adapting the principles of Service Oriented Architectures and the initial steps in the standardization towards service delivery frameworks are done. Special requirements that have to be considered are aspects of service life-cycle management and service composition that are critical regarding high performing real-time communication processes as needed in a telecommunications environment. Beyond the requirements of dynamic service deployment, management, evolution and service termination there is the intention of a well performing agile service composition mechanism. In order to fulfill the discussed requirements some related work in this area is presented and the promising IT / Network standards are enlightened. Furthermore, the architectural design of the developed framework is presented. Finally the paper demonstrates a practical use case scenario that shows the dynamic integration and combination of telecommunication services with services of the Web domain.

Keywords – Service Composition, Next Generation Network (NGN), IP Multimedia Subscsystem (IMS), Service Delivery Plaform (SDP), Open Service Gateway initiative (OSGi), Service Component Architecture (SCA), Service Oriented Architecture (SOA)

I. INTRODUCTION

TRADITIONALLY the telecommunication domain has been a closed environment that did not allow a simple access on Telco assets as well as the integration of third party services. In addition the comparably slowly evolving service market of the Telco domain has to compete against the fast, creative and innovative force of the World Wide Web (WWW). The open character of the Internet has generated the synonym Web 2.0 that declares the WWW as a platform, allowing application developers to easily publish services via open Application Programming Interfaces (APIs) using standards like Web Services [36], REST [13] or JSON/XML-RPC [3][4]. Furthermore, the Web 2.0 standards enable Web application developers to create collaborative service mashups, which combine different kinds of services to whole new applications. A related topic driven by the IT business sector is the Service Oriented Architecture (SOA) that comprises principles like flexibility, a loosely coupled service infrastructure or the high grade of service reusability. In addition the Time To Market (TTM) cycles of new services can be reduced. Facing the changed business environment the Telco operators are breaking new grounds to open their domain specific assets towards third parties. However, in order to provide an expedient TTM for new services within the Telco domain an agile service environment for bidirectional service integration is required. In particular this environment should support the export of internal Telco specific services to the Web based service markets, Small and Medium Enterprises (SMEs) or individual third party developers as well as the usage of those external non-telecommunication resources. This paper introduces the two terms Inside-Out (IO) for distributing internal services to the outside world and Outside-In (OI) for integrating external services to enrich the inherent telecommunication services with new functionalities. Figure 1 visualizes how Inside-Out services and Outside-In services are connected with each other. The operator and third parties both have an asset domain. In other words, they have a certain portfolio of capabilities. The enabling layer is the point of distribution of those capabilities to the outside world. Normally a certain service endpoint is specified where the capabilities are reachable (e.g. an URL for a Web Service). At the end the application domain on both sides can benefit from the usage of an enlarged portfolio of services. The operators are able to integrate 3rd party services into their telecommunication services and vice versa.

Similar to the mentioned Web 2.0 service mashups, a SOA offers mechanisms to compose processes on lower abstraction levels to new complex services on a higher abstraction level. The valuable outcome is a rapid and simple development process. These approaches have to be adopted by the Telco sector performing at the same time the typical telecommunication requirements. In the Web application domain services are usually delivered in a best effort constitution, in contrast to the Quality of Service (QoS) assurance in the telecommunication domain. Due to the market requirements the Telcos are already in the process of opening up their environment. Some examples are British Telecom (BT) with the Ribbit platform [30], Deutsche Telekom with the developer garden [31] and Orange has introduced the Orange Partner program [32]. All of them bring up their own proprietary solutions in the absence of an universally accepted standardization.

978-1-4244-8704-2/11/$26.00 ©2011 IEEE

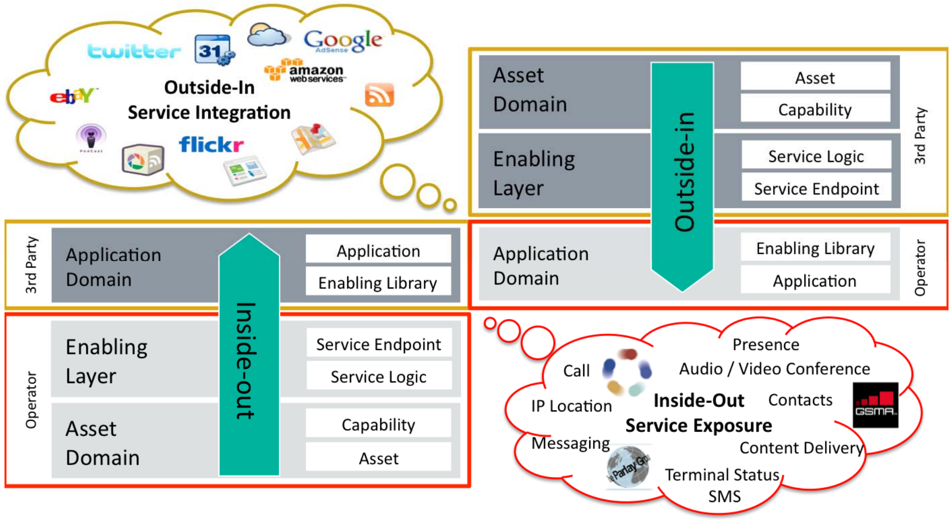

Figure 1. – Outside-In and Inside-Out Approach

This paper presents a Telco tailored service execution environment for NGN and non-telecommunication services by considering the telecommunication and IT standardization bodies. The proposed concept is based on an analysis of the current state of the art in service management, integration, and the publication of services as well as the service composition. In order to address a wide range of potential application developers the framework has to offer a generic solution for distributing a service via various access channels. The design is targeted on a transparent usage of NGN and non-telecommunication services. The exposed prototype demonstrates a possibility for improving the service delivery and composition in order to enhance the ease for developers to build new applications. Besides, the composition of services inside the execution environment can improve the performance on the client side. An example would be resource constraint end devices that could benefit from outsourcing the complex service composition to the backend. Those network centric services are hosted on a platform according to the Software as a Service (SaaS) principle. Third party development enabling is a further and important step of the Telco operators towards open environments as well as innovative products. The consequence is that the end-user can choose from an extended selection of services and the consumer satisfaction increases. An established example in the telecommunication world is the success of Apples App Store [6]. In this case applications or services are spread on the mobile phone (iPhone) whereas our proposed framework gives the opportunity to develop services in the backend.

However, the desire of Telco operators for a flexible and reliable service delivery system is not that novel. The Next Generation Networks (NGN) [17] as a platform for service routing and the IP Multimedia Subsystem (IMS) [10] as a service overlay architecture for NGNs already changed the telecommunication networks. The Open SOA Telco Playground (OSTP) [14] is a testbed of the Fraunhofer FOKUS that is displayed in Figure 2. The proposed service platform in the context of the OSTP is incorporated as Service Broker. In section II a short overview of related work is provided. The main concept and the architectural details are explained in section III. The proof of concept implementation is validated by a use-case scenario in section IV. Finally the paper ends with a conclusion and outlook in future work, which is done in section V.

II. Related Work

This section is intended to give a short overview about the academic and industry activities that have been done in the area of dynamic service management, the SOA programming model Service Component Architecture (SCA) [24] and service composition.

A. *Dynamic Service Management*

In this section the research activities related to the Open Service Gateway initiative (OSGi) [27] environment as a dynamic module system is observed in detail. The aim of the OSGi Alliance is to create a market for universal middleware. "Enhancing Residential Gateways: OSGi Service Composition" [11] proposes a framework to support the composition of OSGi services. It is stated as a great successor in the field of the networked home. According to the authors the framework introduced in this paper improves the installation and reconfiguration process of services. The solution for composing services is based on Business Process Execution Language (BPEL) [25]. The extension of the OSGi framework with a SIP service is discussed in [9]. The authors propose a SIP-based device communication service for OSGi. They believe that a combination of SIP and OSGi helps to support interaction between mobile devices and home network devices. Another reasonable work [16] describes a programming model for multimedia communication in NGNs. The platform is called Instant-X and is intended to provide an API that abstracts from the underlying technology and protocol. The OSGi infrastructure is used to resolve dependencies and for an easy exchange of components. Using this approach it is possible to change the underlying network protocol for multimedia communication without changing the application as the API remains. Furthermore the authors state that this approach aligns with the SOA paradigm of allowing multiple applications to share common multimedia tasks. Our framework concentrates on the highly dynamic composition of services made available by service providers, integrated from the outside world (e.g. Web Services). Additionally, third party developers are able to upload their services into the framework. OSGi fits perfectly the needs for a dynamic integration of certain assets.

B. SOA programming model

SCA is a programming model for SOAs for simplifying the application development. It was developed by an informal industry consortium the Open Service Oriented Architecture (OSOA) [28] and has been handed over for further standardization to the Organization for the Advancement of Structured Information Standards (OASIS) [23]. The paper [20] states out that a combination of SCA and OSGi is very powerful in SOAs. Authors of [18] provide ”A Solution Model for Service-Oriented Architecture”. In this model they adopt a series of widely accepted standards like Service Data Object (SDO), SCA and BPEL. SCA is used to process components via standard interfaces. In [29] a middleware for mobile devices is presented. This middleware is based on SCA because the authors wanted to adapt SOA and component paradigms. The SCA framework supports the idea of SaaS by exporting the software building blocks via multiple protocols as services. In contrast to the middleware on mobile devices the proposed framework in our work runs on the operators side but the ideas are similar. The great claim in our framework is the approach of service availability and usability. A service integrated into the environment can seamlessly be used by every other service. SCA delivers an interesting model for distributing and combining services.

In contrast to most of the ongoing research the prototype in this paper should be a framework, which combines the ability of service composition, service delivery and a dynamic service life cycle management. The ambition is to be as flexible and extendible as possible.

C. Service Composition

Service composition paradigms play an important role in the Information Technology (IT) SOA world and have been reused in the telecommunication sector. Authors in [15] provide an overview of service composition technologies for IP-based telephony and present a graphical Service Creation Environment (SCE). This SCE enables also non-telecommunication experts to create combined services. Authors in [7] present a case study of Parlay X service composition using a model-driven approach and [8] describe a similar approach using a Web-based service creation toolkit to address especially SMEs. The paper [21] examines the orchestration of workflows in real-time communication, while the authors raise the question if the orchestration of these workflows with Web Services is as convenient as Web Services perform in the data environment. The most important requirement is the real-time capability of Web Services. The paper takes a deeper look at three approaches for orchestrating workflows: BPEL, Call Control eXtensible Markup Language (CCXML) [34] and State Chart eXtensible Markup Language (SCXML) [35]. The authors come to the conclusion that CCXML/SCXML are more sufficient for orchestrating fine-grained scenarios as been found in real-time communication. They suggest using CCXML/SCXML in their proposed execution environment. Another study on the usage of SCXML for the real-time composition of network services is provided in [22]. In this work an extended Service Capability Interaction Manager (SCIM) [5] reference architecture with an integrated SCXML based orchestration engine is presented. The proposed framework realizes a simplified integration of combined IMS and non-IMS services into SDP/SOA environments and enhances the flexibility and a rapid development process can be achieved. The existing solutions for orchestration are mainly focusing on addressing specific applications, rather than generic solutions. It is difficult to compare the advantages and disadvantages of various solutions, because they are usually available to certain scenarios or specific domains only. The mentioned activities related to service composition are limited in the usage of network protocols and bound to specific technologies. Nevertheless considering the specific real-time requirements in communications network it becomes apparent that SCXML is a promising candidate as it is emphasized on the real-time execution of composite services in our proposed solution.

III. ARCHITECTURAL APPROACH

The framework proposed in this paper is utilized as a platform for third party service developers. It is possible to reuse and combine services from other developers in order to create complete new services. Our Service Broker solution is based on top of an NGN testbed, the OSTP that is displayed in Figure 2, while it relies on the open source project Open IMS Core [12]. Various service enablers (e.g. Presence, Location, Messaging, etc.) are building an enabler sphere that is exposed through the Parlay X - Enabler Exposure API. This fundamental service environment is interconnected with our Service Broker framework. The invocation of services, offered by the Broker is done by SIP IMS clients through the NGN-based Core network and via Web 2.0 interfaces, like JSON-RPCs, SOAP or REST requests. Incoming service requests will be generalized and analyzed in terms of service selection, configuration and service execution. In more detail the necessary parameters will be extracted and evaluated in order to discover the appropriate service. In case of composed services, the composition script will be loaded into the appropriate execution engine (e.g. BPEL, SCXML). The industry driven standardization bodies Open Mobile Alliance (OMA) [2] and TeleManagement Forum (TMF) [1] have been considered in the architectural design. The specified Service Life Cycle Management (SLCM) handles the different processes of service design, deployment, testing, activation up-to the service termination. The OMA defined the Open Service Provider Environment (OSPE) [26] and the TMF a Service Delivery Framework (SDF) [33].

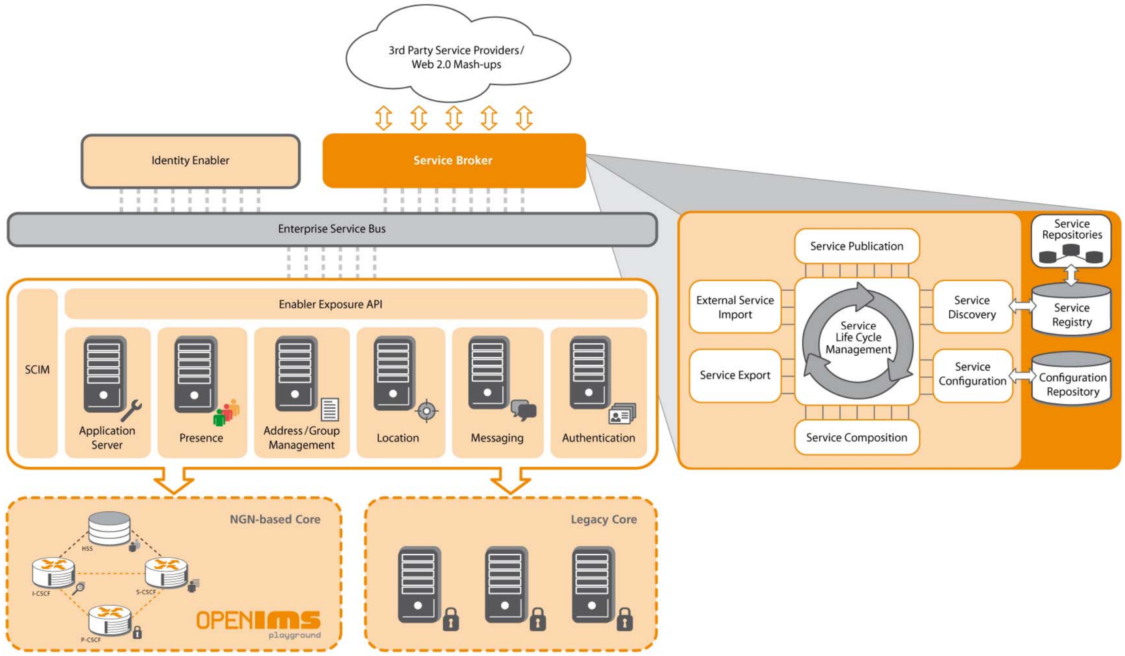

Figure 2. – Service Broker Architecture Embedded into the Open SOA Telco Playground (OSTP)

The Service Broker platform is divided into several functional units as depicted in the right part of Figure 2. In the following all the important parts of the framework and the applied technologies are explained. According to the requirements of a dynamic, modular environment the Service Broker is based on an OSGi service platform that builds the core element of the proposed architecture. The life-cycle management abilities of the OSGi platform apply to the mentioned SLCM requirements. It is used for installing, starting, stopping and uninstalling new software components.

The **Service Export** unit is enabling the operator of the platform to install new services and to host applications of third party developers on the platform. These software building blocks are OSGi compliant Java archive (JAR) files and called bundles.

After the installation procedure the software building blocks can be exported as services via different kind of protocols (e.g. REST, SOAP, RMI, JMS, JSON/XML-RPC), enabled by the **Service Publication**. The objective is to reach a huge amount of potential developers or customers. In addition network operators having usually a limited service portfolio that can be massively increased by innovative service developments and novel value added services. SCA as a programming model for SOA based systems provides also a technical solution for the protocol-based export of software.

The **Service Import** unit supports the integration of externally located network service APIs like REST or Web Services into the framework. Services from the WWW domain, SMEs or communities can be automatically integrated by providing the service endpoint and are available in the framework afterwards. This approach provides an additional way of reusing already existing services.

Using the **Service Composition** unit, all proposed services can be combined and capacitates reusability on very high grade. According to the telecommunication requirements the composition technologies have to cope with asynchronous, stateful and also with long-running procedures, while enabling a rapid development process. Two different types of composition are used, the component-driven composition in particular SCA and the workflow-driven composition with BPEL and SCXML. All models are not only stand-alone approaches. Each of them can be interconnected and developers should be able to reuse those models to rapidly create new composed services. BPEL was chosen because of the highly acceptance in the business world, whereas SCXML is performing very well in comparison to BPEL. The combination of state machines and script based XML syntax complies with our requirements in delivering orchestrated real-time services. A detailed theoretical analysis of those languages can be found in [19]. The component-driven composition mechanism of SCA is focusing on the direct combination of software building blocks in order to make them easily reusable. In this approach components are wired using configuration scripts. A great advantage of that approach is the ability to easily replace one component with another implementation.

The registration and discovery of services is controlled by the **Service Discovery** entity. Developers or service suppliers are able to publish and register new services. Those services got a description and can then be discovered by other developers. A central service registry and repository is storing that service information. The OSGi framework is providing a solution by using the inherent OSGi Bundle Repository (OBR) that automatically resolves required dependencies from the discovered service. Furthermore, a cluster of federated OBRs can be used to retrieve additional resources.

Another functional unit is the fine-grained **Service Configuration** that allows the adjustment of services.

Software components can be configured on service basis as well as on a single user basis. In the framework the service and user configuration is stored by using the OSGi Config Admin a service that is already usable in the OSGi Service Platform. The advantage of using this service is that the configuration parameters are available in the whole platform.

IV. PROOF OF CONCEPT

The following composed service scenario intends to cover all facets of the proposed framework as displayed in figure 3. The platform includes on the Service Provider level an IMS core related SIP based telephony conferencing service and a Short Message Service (SMS).

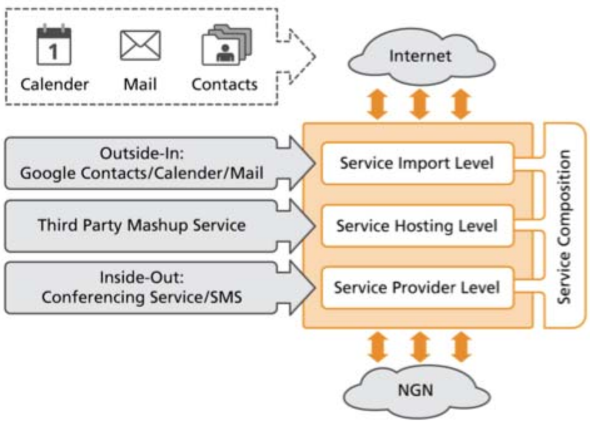

Figure 3 – Architectural Scenario

On the Service Import level the externally running Google service Google Mail, Contacts and Calendar have been integrated. In case of the Service Hosting level a developer installed a third party mashup on the framework that reuses the Google services and the telecommunication services of the provider. The Mashup Service scenario is depicted in Figure 4.

Figure 4 – Mashup Service

First it is retrieving calendar information and checks if an appointment for a telephone conference is available. If there is such an appointment the application initially schedules an SMS reminder some minutes before the beginning of the conference. Secondly it starts the conference at the defined time in the Google Calendar. The service is exposed by the platform exemplary via REST and can be triggered by various Web application technologies.

In order to schedule every planned telephone conference automatically in the backend a SCXML based service composition script is used that checks continuously whether new appointments are available. The user can start and stop the script in the personal configuration area of the Service Broker that is reachable on a website.

Presenting this short use case we wanted to show how easy and flexible the framework allows to integrate new services. The special characteristic of this scenario is the combination of services that are developed by different developers. The loosely coupled service modules in the framework allow a freely connection of all installed assets.

V. CONCLUSION AND OUTLOOK

The composition of services is a common approach of service-oriented environments. The presented framework is a proper solution to interconnect telecommunication and Web based services in a fast and easy way. Service composition scripts can be written in short time and deployed on a Service Broker. The proposed service platform enables Third Party Development on Telco operator environments. Developers can upload new OSGi compliant software libraries and reuse other services that are available on the platform. Furthermore, it is presented how the proposed architecture is tightly integrated into the IMS-based Fraunhofer FOKUS Open SOA Telco Playground solution and how OSGi fits perfectly in the requirements of the service life cycle management. The combination of different composition approaches (e.g. BPEL, SCXML, SCA) allows a powerful and rapid service development. Besides, SCA is shown as an appropriate approach to export and compose services efficiently. In this framework the combination of OSGi and SCA produced a flexible and usable prototype implementation.

In our future work we will integrate further functional components. It is planed to integrate a policy engine based on the OMA PEEM [37] standard that is able to control the access on services as well as to support a service selection process dependent on the user profile information. Another missing but important component is an identity management system. Finally we work on a clusterable Service Broker solution with virtualization support in order to fulfill the requirements for elastic Cloud Computing.

REFERENCES

[1] TM Forum. http://www.tmforum.org/.
[2] OMA (Open Mobile Aliance) http://www.openmobilealliance.org/
[3] XML-RPC. http://www.xmlrpc.com/.
[4] Json-rpc 1.1 specification, working draft, August 2006. http://json-rpc.org/wd/JSON- RPC- 1- 1- WD- 20060807.html
[5] 3GPP. SCIM, 3GPP TS 23.002 specification. http://www.3gpp.org.
[6] Apple. Apple App Store. www.apple.com/de/iphone/appstore.
[7] N. Blum, T. Magedanz, J. Kleessen, and T. Margaria. Enabling Extreme Model Driven Design of Parlay X-based Communications Services for End-to-End Multiplatform Service Orchestrations. In Engineering of Complex Computer Systems, 2009 14th IEEE International Conference on, pages 240 –247, 2-4 2009.
[8] Niklas Blum, Thomas Magedanz, and Horst Stein. Service Creation & Delivery for SME based on SOA / IMS. In MNCNA '07: Proceedings of the 2007 Workshop on Middleware for next-generation converged networks and applications, pages 1–6, New York, NY, USA, 2007. ACM.
[9] A. Brown, M. Kolberg, D. Bushmitch, G. Lomako, and M. Ma. A SIP-based OSGi Device Communication Service for Mobile Personal Area Networks. In Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE, volume 1, pages 502 – 508, 8-10 2006.
[10] Gonzalo Camarillo and Miguel-Angel Garcia-Martin. The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds, Second Edition. John Wiley & Sons, 2006. ISBN: 0470018186.

[11] R.P. Diaz Redondo, A.F. Vilas, M.R. Cabrer, J.J. Pazos Arias, and Marta Rey Lopez. Enhancing Residential Gateways: OSGi Service Composition. Consumer Electronics, IEEE Transactions on, 53(1):87 –95, february 2007.
[12] Thomas Magedanz Dragos Vingarzan, Peter Weik. Design and Implementation of an Open IMS Core. In MATA, pages 284–293, 2005.
[13] Roy T. Fielding. Architectural styles and the design of network-based software architectures. PhD thesis, University of California, Irvine, 2000.
[14] FOKUS. Open SOA Telco Playground. www.opensoaplayground.org/.
[15] R.H. Glitho, F. Khendek, and A. De Marco. Creating Value Added Services in Internet Telephony: an overview and a case study on a high-level service creation environment. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 33(4):446 –457, nov. 2003.
[16] Jan-Patrick Elsholz, Holger Schmidt, Sven Schober, and Franz J. Hauck. Instant-X: SOA for Multimedia Communication in NGNs. Technical Report, Universita t Ulm, VS-R11-2009, November 2009.
[17] K. Knightson, N. Morita, and T. Towle. NGN Architecture: Generic Principles, Functional Architecture and Implementation. Communications Magazine, IEEE, 43(10):49 – 56, oct. 2005.
[18] Jianzhong Lan, Yi Liu, and Yueting Chai. A Solution Model for Service-Oriented Architecture. In Intelligent Control and Automation, 2008.WCICA 2008. 7th World Congress on, pages 4184 –4189, 25-27 2008.
[19] Lajos Lange, Thomas Magedanz, Niklas Blum, and Tiziana Margaria. Assessing Workflow Languages for Composition of Real-time Communication Services. In ICIN 2010, Weaving Applications into the Network Fabric - The Transformational Challenge, Paper already accepted, Berlin, Germany, 2010.
[20] Weibo Li, Yanbing Zhang, and Jie Jin. Research of the Service Design Approach based on SCA OSGi. In Services Science, Management and Engineering, 2009. SSME '09. IITA International Conference on, pages 392 –395, 11-12 2009.
[21] Lin Lin and Ping Lin. Orchestration in Web Services and Real-time Communications. Communications Magazine, IEEE, 45(7):44–50, 2007.
[22] Serge Moro, Sebastien Bouat, Marie-Paule Odini, and John O'Connell.Service composition with real time services. In ICIN 2008, 2008.
[23] OASIS. Organization for the Advancement of Structured Information Standards. http://www.oasis- open.org.
[24] OASIS. SCA Assembly. http://docs.oasis-open.org/opencsa/sca- assembly/sca- assembly- 1.1- spec- cd03.pdf.
[25] OASIS. Web services business process execution language version 2.0, 2007. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.
[26] OMA. OMA Service Provider Environment (OSPE) V1.0. http://www.openmobilealliance.org/Technical/release program/ospe v1 0.aspx.
[27] OSGi Alliance. OSGiAlliance. http://www.osgi.org.
[28] OSOA. Open Service Oriented Architecture. http://www.osoa.org.
[29] D.Romero,C.Parra,L.Seinturier,L.Duchien,andR.Casallas.Ansca Middleware Platform for Mobile Devices. In Enterprise Distributed Object Computing Conference Workshops, 2008 12th, pages 393 –396, 16-16 2008.
[30] British Telecom. Ribbit for developers. http://developer.ribbit.com/.
[31] Deutsche Telekom developergarden http://www.developergarden.com/.
[32] Orange. Orange partner programme. http://www.orangepartner.com/.
[33] TM Forum. TR139, Service Delivery Framework Overview, Release 2.0. http://www.tmforum.org/.
[34] W3C. CCXML. http://www.w3.org/TR/ccxml/.
[35] W3C. SCXML. http://www.w3.org/TR/scxml/.
[36] W3C. Web Services Glossary, 2004. http://www.w3.org/TR/ws-gloss/.
[37] OMA Policy Evaluation Enforcement Management (PEEM), http://www.openmobilealliance.org/Technical/Release_program/peem_v1_0.aspx