Elektrotehniški vestnik 72(1): 45–51, 2005
Electrotechnical Review, Ljubljana, Slovenija

NGN Service Development - Overview and Parlay X Implementation

Luka Zebec, Iztok Humar, Darko Bodnaruk, Andrej Kos, and Janez Bešter

University of Ljubljana, Faculty of Electrical Engineering, Tržaška 25, 1000 Ljubljana, Slovenia
E-mail: luka.zebec@fe.uni-lj.si

Abstract. One of the important advantages that comes with NGN are open interfaces, that enable migration of network intelligence from the telecommunications network elements to the application servers. The usage of Web based interfaces (e.g. Web Services) provide wide community of web developers with access to telecommunications elements functionality. This new concept enables rapid third-party development of innovative services. In this paper we explain the concept of providing Web Services through gateways, used to establish cooperation with legacy telecommunications network elements interfaces. One of such is our implementation of CSTA/Parlay X gateway. CSTA interface was chosen on the network side because it is frequently supported by network elements. Parlay X interface was chosen on the Web side, while it is based on standard Open Services Architecture. We also demonstrate the usage of CSTA/Parlay X gateway by the Parlay X application we developed. Application allows user to respond the sender of currently selected e-mail with a phone call from e-mail client application.

Key words: Service Architecture, Next Generation Networks, Intelligent Networks, CSTA, Parlay, Parlay X, Web Services, Service Gateway

Razvoj NGN storitev - pregled in izvedba Parlay X

Povzetek. Ena poglavitnih prednosti, ki jo prinašajo prihajajoča omrežja naslednje generacije, so odprti vmesniki, ki omogočajo selitev inteligence iz omrežnih elementov v aplikacijske strežnike. Preko spletnih vmesnikov (npr. spletnih storitev) bodo omrežja naslednje generacije skupnosti razvijalcev spletnih aplikacij ponudila dostop do zmogljivosti telekomunikacijskih omrežnih elementov, kar bo omogočilo hitrejši razvoj novih aplikacij. V članku je predstavljen koncept razvoja spletnih storitev z uporabo prehodov, ki omogočajo dostop do vmesnikov klasičnih telekomunikacijskih omrežnih elementov prek spleta. Opisan je CSTA/Parlay X prehod, ki je bil razvit v ta namen. Na omrežni strani smo izbrali CSTA vmesnik, ki je pogosto podprt na omrežnih elementih. Na strani spleta smo izbrali Parlay X vmesnik, ki temelji na arhitekturi odprtih storitev. Predstavljena je Parlay X aplikacija, ki omogoča vzpostavljanje telefonskih klicev s pošiljateljem v e-poštni aplikaciji trenutno izbrane elektronske pošte.

Ključne besede: storitvene arhitekture, omrežja naslednje generacije, inteligentna omrežja, CSTA, Parlay, Parlay X, spletne storitve, storitveni prehod

1 Introduction

In recent years it has become clear that fixed and mobile telephone network will converge with Internet. This is the first time these networks are used together side by side and form a hybrid network that needs a versatile telecommunication services delivery mechanism.

The first concept of service architecture, that was introduced almost two decades ago, is a concept of Intelligent Network (IN). It was most popular way to offer value added services within the telecommunication networks. Unfortunately, this approach kept all network intelligence concentrated at the network operator side. It did not simulate third-party service providers to develop value added services. IN services development requires a high level of expertise in telecommunication networks and protocols, thus limiting the community of developers to a small number of telecommunication services specialists.

On the contrary, the most important advantage of Next Generation Network (NGN) concept, that was introduced afterwards, is their open interfaces and usage of Web based technologies (Web Services). This openness enables flexible and easy service creation for both service providers and third party application developers. NGN approach migrates network intelligence from the network operator towards the network edge - to the application service providers.

Web Services offer highly abstracted interfaces to the telecommunication network element's functionality, based on eXtensible Markup Language (XML). The use

Received
Accepted

of Web Services as an application interface reduces the knowledge required to develop applications to object oriented technologies and XML-based communication.

Applying NGN concept to the interworking with legacy telecommunication systems requires the usage of gateways. This approach is used for signaling (signaling gateways) and media (media gateways), as well as for application interfaces (Parlay gateway). Since Computer Supported Telephony Application (CSTA) interface is generally supported by existing telecommunications, we used this approach to implement CSTA/Parlay X gateway, based on Web Services.

In the related work, mostly Parlay/Parlay X gateways have been developed [1][2][3]. Since Parlay demands additional costs and long integration time, it is not always a suitable solution, particularly for a fast, simple and inexpensive NGN-based interface implementation. The rest of this paper is organized as follows. In Sect. 2 we overview service architectures and present the role of CSTA, Parlay and Parlay X interfaces. Section 3 identifies the possibilities to map the Parlay X Web Services to CSTA services, presents the implementation of gateway, demonstrates development of application, and presents the application functionality. The main implications are summarized in Sect. 4.

2 Overview of service architectures and interfaces

The first concept of service architecture appeared at the beginning of 1980's with the Intelligent Network [4][5]. It was designed to offer value added services in Public Switched Telephony Networks (PSTN). The main idea was to enable external service logic processing by separating service control from service switching. In the IN architecture, service switching is implemented in trunk switches - Service Switching Points (SSPs). SSPs trigger service servers - Service Control Points (SCPs) that are providing services. For signaling between SSPs and SCPs, Intelligent Network Application Part (INAP) protocol is used.

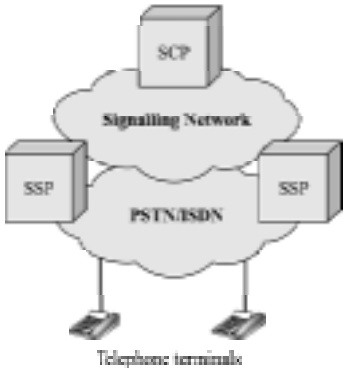

Figure 1. Intelligent Network service architecture

Nowadays, the IN architecture is still in use by the majority of network providers. Examples of IN services include Pre-paid Service, Green Numbers, Televoting etc. Despite the idea of separating services and switching functionality, the IN concept has several drawbacks [4], where the main are lack of openness and the protocol-based application development. For these reasons, the service development was always kept in the domain of telecommunications vendors and operators. Thus, only a small number of services were developed.

The appearance of Internet and IP networks brought a different concept of service architecture, where the service user is in a direct communication with application that provides a service [4]. While network elements are mainly involved in packed data transport issues (i.e. routing, switching, QoS and security), the service logic is located on application servers and, if necessary, on clients. The Internet communication protocols are simple, often text oriented and many platforms for application development exist. This resulted in mass service development, deployment and success.

In the last years, we are faced with the convergence of fixed, mobile and IP networks and services. The networks are interconnected via different gateways (e.g. signaling gateway - SG, media gateway - MG). Common call servers (CS)/softswitches exist to control network elements and different types of application servers (AS) are available for service deployment. Converged services, such as Multimedia Conversation, Messaging and Integrated Video are mostly based on characteristics of the Internet service architecture. Such converged networks are most commonly called the Next Generation Networks (NGN).

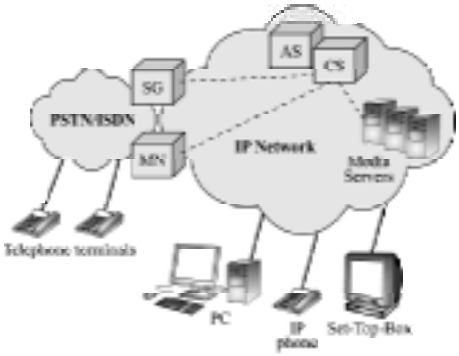

Figure 2. Converged networks architecture

A wide range of open application interfaces and service creation technologies exist in NGN. Generally, they can be classified in three main categories [6]: Application Programming Interfaces (API) based on objects technologies, (e.g. Parlay, Parlay X, JAIN), protocol based interfaces, (e.g. SIP, INAP, CSTA) and scripting languages (e.g. Voice XML, CPL, CCXML).

It is expected that in the future, today's hybrid net-

work will migrate toward IP-based packet network. Consequently, a need to have common service architecture will emerge. Many activities for migration exist. The main architecture is built on IP Multimedia Subsystem (IMS) [7], defined by the European Telecommunications Standards Institute (ETSI) and the 3rd Generation Partnership Project (3GPP), where the Session Initiation Protocol (SIP) is the protocol of choice. From an application development point of view, the Parlay API will also play an important role in IMS.

The migration towards new services architecture will not be instantaneous. Many legacy network elements will still need to operate. Hence, while providing more sophisticated services for users of new technologies, new application interfaces bring the opportunities to also enhance legacy services. In the following subsections we will introduce CSTA, Parlay and Parlay X interfaces, as we use them for providing Web Services through the CSTA/Parlay X gateway that has been developed.

2.1 CSTA

Computer Supported Telephony Application (CSTA) [8] is a standard interface, introduced in mid 90's by European Computer Manufacturers Association (ECMA). It is intended for use in Computer Telephony Integration (CTI). CSTA enables third party applications with an access to the call handling mechanism in telephony systems.

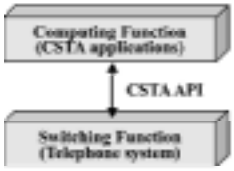

Figure 3. CSTA - functional diagram

CSTA is implemented as an application interface (API) between a Switching Function (e.g. telephone switch) and a Computing Function (e.g. CTI system). CSTA API is disassociated from Switching Function interfaces, operates generically and hides the differences in interfaces from CSTA applications. CSTA API normally communicates using the Ethernet connection with a TCP/IP protocol stack. CSTA messages are structured using the Abstract Syntax Notation One (ASN.1) formal language.

The API supplies an access to different sets of services. Currently, three versions of CSTA standard exist. Phase I [8] comprises Switching Function Services, Status Reporting Services, Computing Function Services and Bi-Directional Services. Phase II [9] extends the above mentioned services with new methods and adds I/O and Voice Units Services. Phase III [10] brings a new classification of services. It adds support for data collection, Call Data Records and terminal capability exchange. It also supports new terminals/clients (e.g. IP terminals, softphones) and new services (e.g SMS, MMS). Recently, ECMA defined XML [11] and Web Services [12] encodings of CSTA.

Many telephony systems today have CSTA Phase I implemented. CSTA application development suffers from the same weaknesses as the IN concept and not many CSTA applications are actually available.

2.2 Parlay

Parlay [13] is a standardized API that enables applications by the access to the network's functionality. It was introduced by the Parlay Group upon an initiative of telecommunications operators, equipment providers, software companies and application providers. Parlay is consistent with the concept of Open Services Access (OSA), supported by the International Telecommunication Union (ITU) and ETSI. The main idea of OSA/Parlay is to create a single API for thirdparty development. Using the API, the applications in a converged network can be built by using standard programming languages and tools.

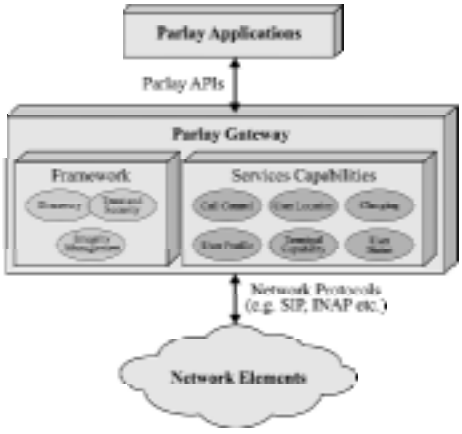

Figure 4. Parlay architecture [14]

Parlay API is typically implemented on a gateway, positioned between application servers and network elements. It consists of Framework and Service Capabilities. Framework offers thirdparty applications functions like security, discovery and management. The Service Capabilities offers the capabilities of the underlying networks: call control, messaging, user interaction, charging etc. [14].

Parlay API is provided in a form of distributed APIs, accessible through a distributed processing mechanism. Mapping on CORBA and Web Services is defined. Although quite a lot of Parlay gateways and application servers were deployed by network operators, the development of services did not achieve the success of an Internet services. The most probable reason is that programming Parlay applications is more complex than programming

Web applications. For example, even a simple Click-To-Dial application requires many method invocations.

2.3 Parlay X

Parlay X [15] is a simplified and highly abstracted Parlay based API. The main goal for introduction of Parlay X was to ensure a simple application creation for developers, especially non-telecommunications experts like Internet applications developers. Parlay X offers the most relevant functionality of Parlay. Upon estimation of Parlay Group [16], 80% of relevant NGN applications can be developed using Parlay X.

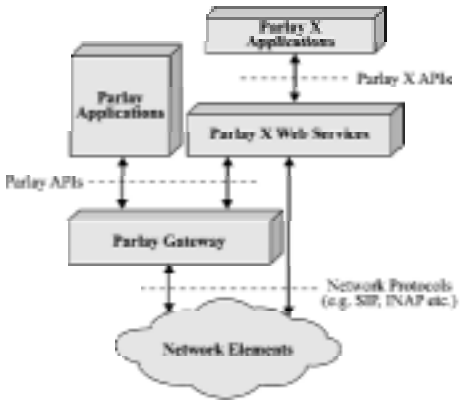

Figure 5. Parlay X architecture [15]

Parlay X API can be a component of Parlay gateway or implemented on a separate gateway, that communicates through Parlay gateway or directly with network elements. Parlay X API is based on Web Services. Communication with the applications is done by exchanging XML (SOAP) messages over HTTP protocol. Parlay X applications are Web based and can be written in any programming language. Consequently, all common Web servers and server technologies can be used for application development; many programming tools already exist and a huge group of potential developers is at hand.

Parlay X (version 2.0) consists of 13 APIs, respectively groups of Web Services. Third Party Call enables applications to initiate calls. Call Notification notifies the applications about initiation of calls and enables them to further impact on a call (routing of a call). Short Message Service (SMS) and Multimedia Message Service (MMS) APIs are intended for messaging. Call Handling API provides applications with a variety of features, like call forwarding, white/black list handling etc. Audio Call API introduces the possibility to play text, audio and VoiceXML messages. Multimedia Conference API deals with conferencing. Presence, Address List Management, Terminal Status APIs are designed for applications like Instant Messaging while Terminal Location API is primarily intended for use in mobile telecommunications. Finally, Payment and Account Management APIs are dealing with billing possibilities.

Although the Web Service application development on the Internet is intensive, the Parlay X type application development is still at the beginning. The reason is slow implementation of Parlay X APIs (gateways) by network operators and the lack of easy and open access to them.

3 CSTA/Parlay X gateway for evolving telephone systems

During the evolution of network architecture, legacy telephone systems will continue to operate. In order to provide their users with NGN-based services and applications, appropriate gateways between legacy and NGN application interfaces can be implemented. Furthermore, such gateways can be used for new network elements, like softswitches and call servers, often supporting legacy interfaces for backward interoperability.

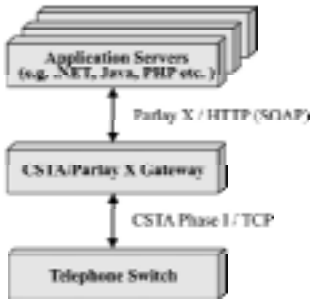

Figure 6. CSTA/Parlay X gateway

Considering the idea of implementing Web Services through a gateway, the CSTA interface, supported by many telephone systems, can be mapped to the Parlay X interface. These mappings are implemented in a CSTA/Parlay X gateway, an intermediate between telephone switch and NGN application servers (Fig. 6).

3.1 CSTA/Parlay X mappings

The first step at CSTA/Parlay X gateway implementation is identification of possible mappings between CSTA services and Parlay X APIs. There are two relevant things to consider regarding mappings, the identification of most relevant Parlay X APIs as well as CSTA Phase and Services.

Considering possible services, applicable for deployment with telephone switches, those already supported by CSTA Phase I should be taken into account first. In Phase I, most of services are engaged with call control, supporting Third Party Call, Call Notification (including Call Direction) and a limited set of Call Handling and Multimedia Conference Parlay X APIs. Method mappings for Parlay X Third Party Call API are shown in Tab. 1.

Parlay X Method	CSTA Methods
makeCall	*Monitor Start (when not already started), MakeCall*
getCallInformation	*Event Report Service (Originated, Established, Connection Cleared etc.)*
endCall	*Clear Connection Monitor Stop (optionaly)*
cancelCallRequest	*Clear Connection Monitor Stop (optionaly)*

Table 1. CSTA/Parlay X mappings for Third Party Call

Regarding CSTA Phase II and III, Audio Call (based on CSTA Phase II) and SMS and MMS APIs can be mapped. In NGN architecture, other Parlay X APIs, not directly linked to CSTA but useful for developing Parlay X applications, can also be provided - either they are implemented on the same gateway (e.g. User Status, based upon event reports) or available across the Internet (e.g. Terminal Location, based on telecom operator's address book).

3.2 Prototype gateway and application

In this subsection, the implementation of CSTA/Parlay X gateway and Parlay X application is described. Both, the gateway and the application were developed using Microsoft operating system and programming tools. They were tested with the SI2000 telephone switch, provided by Iskratel [17].

Windows platform with .NET framework and Visual Studio .NET (VS .NET) was chosen because it provides powerful programming environment. Web Services definition tool that comes with VS .NET enables transparent translation of Web Service Definition Language (WSDL) files, provided by ETSI into programming skeleton, both for Web Services server and client. .NET framework also generates a service description web page with example Web Services method calls.

3.2.1 CSTA/Parlay X gateway

The CSTA/Parlay X gateway provides Parlay X Web Services for thirdparty application development, maps Parlay X Web Services to CSTA Services and handles CSTA communication with the telephone switch. The gateway is based on the previous described mappings, currently supporting Third Party Call Parlay X API.

The gateway comprises two modules, which communicate over a TCP connection. CSTAConnector is used for communication with the switch and API mappings. ThirdPartyCall implements the Parlay X API.

The CSTAConnector is a Windows console application, implemented in C++ programming language. The module maintains the TCP connection with the telephone switch, handles the method calls from ThirdPartyCall, encodes/decodes CSTA messages to ASN.1 notation (by using an API by Objective Systems [17]), maintains the state of the CSTA service and generates responses to method calls, initiated by ThirdPartyCall module.

ThirdPartyCall is a .NET Web Services application, implemented in C# programming language. The module handles HTTP/SOAP communication with Parlay X applications, implements Third Party Call API (Web Services) methods and handles communication with the CSTAConnector.

3.2.2 Parlay X application

The Parlay X application Call Sender is a Click-To-Dial application, which enables the user of Microsoft Outlook E-mail client to establish a call with the sender of the currently selected e-mail by clicking a button.

The Parlay X application is implemented as an Add-In application for Microsoft Outlook. Upon a skeleton code, provided by VS .NET, we implemented the code for MS Outlook button generation, added the web reference to the remote WSDL method (Parlay X Web Service) on the gateway and created a configuration file for user preferences. Configuration file contains the user's phone number (calling party), prefixes for the different number types (business, home, mobile etc.), belonging to an e-mail sender (called party) and the selection priority between these numbers.

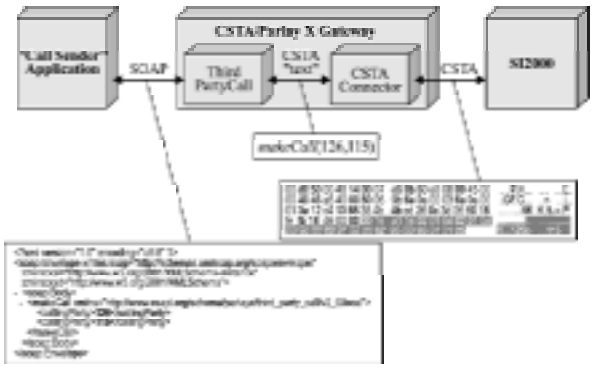

Figure 7. MakeCall method call in CSTA/Parlay X gateway

The main part of implementation was programming the code that handles the button click. The code includes reading of the e-mail address of the selected message, searching over contact, the selection of appropriate telephone number (regarding to configuration file), initiating the remote Web Service method call *makeCall* on CSTA/Parlay X gateway and further call handling. The call handling code includes a new Web Service call, where *getCallInformation* method supplies the application with the status of the call. Regarding the call status, the code for user interaction is invoked that include calling of remote *endCall* and *cancelCallRequest* methods.

The application works as follows. When a user clicks the Call Sender button, an entry in the address book

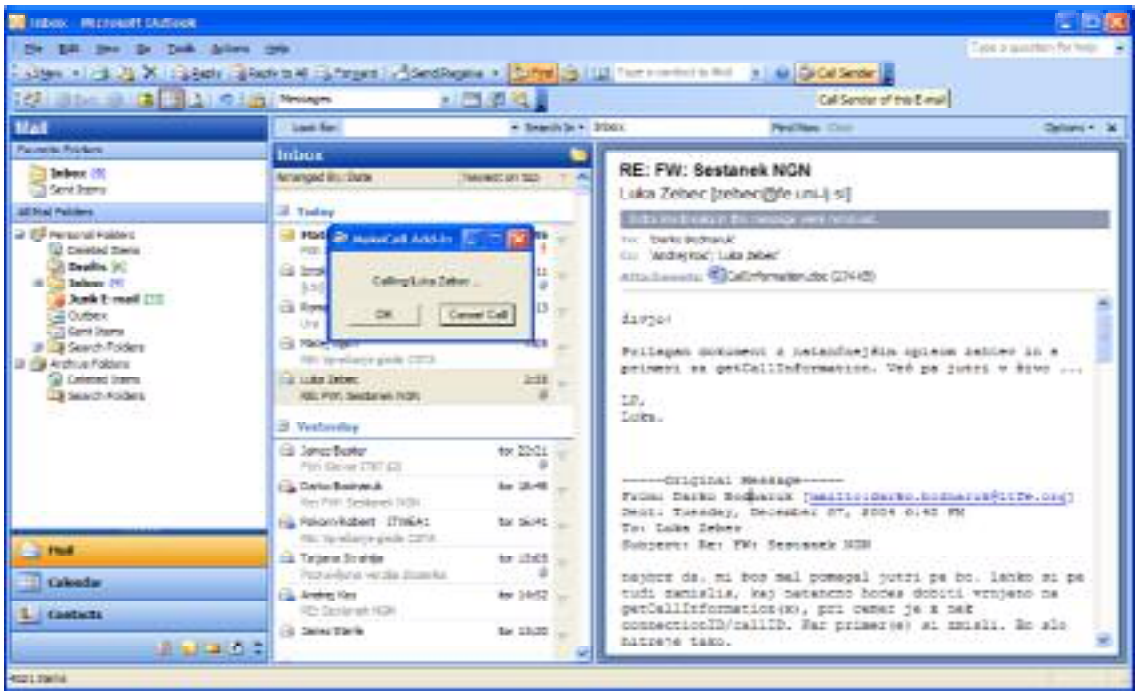

Figure 8. Usage of Parlay X application

matching the currently selected e-mail message is first found. A phone number of sender is selected in contacts and the call between the user and the e-mail sender is established. A message box is shown to the user, showing the state of the call and providing him with options to close the box or cancel the call. Fig. 8 shows the usage of the application.

4 Conclusion

NGN architecture brings a concept of open application interfaces using Web based technologies, which offer highly abstracted XML-based interfaces to the telecommunication network element's functionality.

This paper explains a telecommunications service and application architecture and development, emphasizing the usage of CSTA/Parlay X gateway for providing legacy telephone switches with an application interface, based on Web Services.

In our work, we employed Parlay X interface, because it is intended to play an important role in NGN architecture. On the network side, the CSTA interface was chosen, while being supported by many telephone switches. To connect these two technologies, we have developed CSTA/Parlay X gateway. The gateway was tested with SI2000 telephone switch, provided by Iskratel.

Furthermore, we explain the mappings between CSTA and Parlay X application interfaces and provide information about gateway implementation. In order to demonstrate the usability of the gateway, some applications have been developed. We present the implementation of Call Sender application, which allows the user to respond the sender of an e-mail with a phone call directly by clicking a button in his/her e-mail client application.

Our future work will include development of additional applications, which rely on CSTA/Parlay X gateway. We also plan to implement additional Parlay X APIs, in extension to already implemented Third Party Call API.

5 Acknowledgement

The research and development work was supported in part by Iskratel and the Ministry of Higher Education, Science and Technology of the Republic of Slovenia.

6 References

[1] Practical Experiences of Parlay X and Parlay Web Services, Eamonn Murray, 2002, http://www.parlay.org/docs/jul2002mm/15AePONA.ppt.

[2] Marconi, Application Mediation Gateway, 2003, http://www.parlay.org/news/news/L1670-Applications_Mediation_Gateway_Brochure.pdf.

[3] T. Magedanz, D. Witaszek, K. Knuttel, Service Delivery Platform Options for Next Generation Networks within the national German 3G Beyond Testbed © Fraunhofer FOKUS, 2004, http://www.fokus.fraunhofer.de/.

[4] E. Bertin, E. Bury and P. Lesieur, Intelligence distribution in next-generation networks, France Telecom R&D, 2001.

[5] P. Kuuppelomki and A. Mustonen, Open architectures in telecommunication and IP convergence, Architecture solution for telecommunication network convergence. Tampere University of Technology, Pori, Series A 43/2002, pp. 32-41, 2002.

[6] C. A. Licciardi and P. Falcarin, Technologies and guidelines for service creation in NGN, 8th ITU International Conference on Intelligence in Networks (ICIN 2003), Bordeaux, France, April 2003.

[7] Lucent Technologies, IP Multimedia Subsystem (IMS) Service Architecture, June 2004.

[8] Services for Computer Supported Telecommunications Applications (CSTA) Phase I, Ecma-179, June 1992.

[9] Services for Computer Supported Telecommunications Applications (CSTA) Phase II, Ecma-217, June 1994.

[10] Services for Computer Supported Telecommunications Applications (CSTA) Phase III, Ecma-269, June 2004.

[11] XML Protocol for Computer Supported Telecommunications Applications (CSTA) Phase III, Ecma-323, June 2004.

[12] Web Services Description Language (WSDL) for CSTA Phase III, Ecma-348, June 2003.

[13] ETSI, Open Services Access (OSA); Application Programming Interface (API), Part 1: Overview; (Parlay 5), 2004.

[14] State of the art overview, Lucent Technologies, KPN Research, Telematica Instituut, November 2002.

[15] Parlay X Working Group, Parlay X Web Services Specification, June 2004.

[16] S. Davis and M. Cookson, View of the future, Parlay Septemberfest, September 2001.

[17] Iskratel, SI2000 TDM, February 2005, http://products.iskratel.com/?book=produkti/SI2000_TDM.htm.

[18] Objective Systems, ASN 1 API's, October 2004, http://www.obj-sys.com/products_asn1apis.shtml.

Luka Zebec received his B.Sc. and M.Sc. degrees in electrical engineering from University of Ljubljana, Faculty of Electrical Engineering, Slovenia, in 2000 and 2003, respectively. He joined the Laboratory of Telecommunications at the same university in 1999. At present, he is employed as a Research associate at the Faculty of Electrical Engineering in Ljubljana. He has extensive research experience in the analysis and design of convergent Internet applications and next generation telecommunications systems.

Iztok Humar received his B.Sc. and M.Sc. degrees in electrical engineering from University of Ljubljana, Faculty of Electrical Engineering, Slovenia, in 2000 and 2003, respectively. He joined the Laboratory of Telecommunications at the same university in 1999, where he is currently working towards his Ph.D. degree. His main research topics include Internet Telephony, Network Traffic Analyses and Modeling, QoS Management and Network Management. He is a member of IEEE and student member of the IEICE.

Darko Bodnaruk graduated in computer science at the University of Ljubljana, Slovenia in 2002. He is currently working at the Laboratory for telecommunications and working towards a M.Sc. degree in economics at the same university. His research topics include embedded systems, multimedia and convergent Internet applications.

Andrej Kos graduated from the University of Ljubljana, Slovenia in 1996 and was awarded his Ph.D. degree in telecommunications in 2003. At present, he works as a Senior Researcher at the Faculty of Electrical Engineering in Ljubljana. He has extensive research and industrial experience in the analysis, modeling, and design of advanced telecommunications elements, networks, and systems. His current work and research focuses on next generation packet switching and management. His other fields of expertise include mobile communications and intelligent networks. He is a member of the IEEE, IEICE, and the Telemanagement Forum.

Janez Bešter received his Ph.D. degree in the field of telecommunications from University of Ljubljana, Slovenia. He is currently the Head of Laboratory for Telecommunications and an Associate Professor at the Faculty of Electrical Engineering in Ljubljana. His work focuses mainly on planning, realization and management of telecommunication systems and services as well as application of information technologies to education. Janez Bešter is a member of IEEE, ACM and also an associate member of IEICE.