

# A framework of service components modelling for multimedia distribution over broadband network

<sup>(a)</sup> *O. Martikainen*, <sup>(b)</sup> *V. Naoumov*, <sup>(c)</sup> *K. Samouylov*,  
<sup>(c)</sup> *M. Zhidovinov*

<sup>(a)</sup> *Helsinki University of Technology*  
*Otakaari 1 room Y228C, FIN-02150 Espoo, FINLAND,*  
*Tel. +358 9451 2174, Telefax +358 9451 3293,*  
*E-mail Olli.Martiainen@hut.fi*

<sup>(b)</sup> *Lappeenranta University of Technology*  
*P.O. Box 20, FIN-53851 Lappeenranta, FINLAND,*  
*Tel. +358 5 621 2828, Telefax +358 5 621 2899,*  
*E-mail Valeri.Naoumov@lut.fi*

<sup>(c)</sup> *Peoples' Friendship University of Russia*  
*P.O Box 9, 117419 Moscow, RUSSIA,*  
*Tel./Telefax +7 095 952 2823,*  
*E-mail Konstantin.Samouylov@mx.pfu.edu.ru,*  
*Michael.Zhidovinov@mx.pfu.edu.ru*

## Abstract

The paper focuses on the middleware design issues for multimedia applications on top of IP switching environment over ATM networks. It describes the JVOPS Framework and the corresponding JVOPS System Architecture.

## Keywords

Media distribution, service modelling, service architecture, broadband network

## 1 INTRODUCTION

With rapid growth of the information industries it seems to be clear that new bridging technologies between different actors of multimedia services provision over broadband networks will be developed and piloted up to the end of the century. These technologies will reflect major shifts at the telecommunications market, which now can be defined as follows

- *shift in the networks infrastructure* towards broadband backbone and access networks;
- *shift in the service value creation* first from the mobile and Internet services;
- *shift in the content industry* where content and/or brokerage service providers will play a significant role in value creation.

One of the enabling factors of these changes is the Internet becoming a basis for service and content creation, and another is the liberalization of telecommunication markets where all main players should join efforts in the creation of network architectures considerably flexible and simpler than today.

The key issue for future networks is to provide a wide range of multimedia services by means of distributed intelligence over user terminals and different service control and provider nodes. The basic idea is presented in the Principles of Intelligent Network Architecture [1] and means that the operation and provision of new services should be independent of core network functions. It is well assumed [2] that modern service architecture must satisfy to the following requirements:

- *support for wide range of services* – support for telecommunication, information and management services;
- *rapid service deployment and provision* – reduce development and deployment cost by components reuse;
- *support for multi-player environment* – open environment for interoperability among multiply providers/operators;
- *universal service access* – access to service independently of physical location and terminal used;
- *independence from the network infrastructure* – service evolution independently of underlying transport and computing technologies.

A number of research and development projects have been already started to support multimedia telecommunication systems implementation. The TOVE (Transparent Object-oriented Virtual Exchange) project [3,4] aims at developing network architecture that would satisfy mentioned above requirements. The research in the Calypso project [5] will focus on how service control should be implemented for media distribution over TOVE broadband network architecture.

In this paper we propose an object-oriented framework architecture for Service Logic Execution Environment modeling and Service Components prototyping in the media distribution control environment.

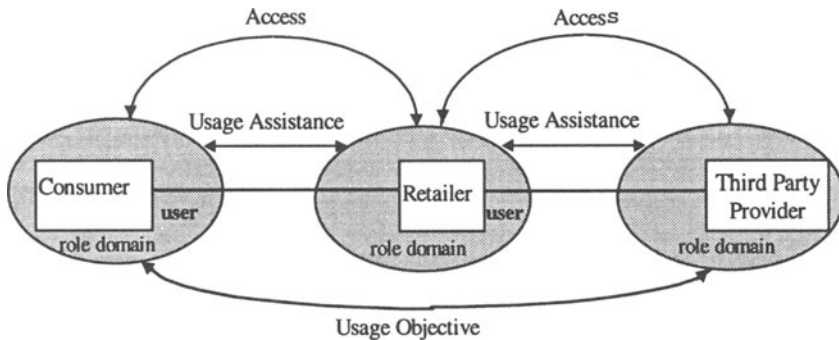
## 2 MULTIMEDIA SERVICES DISTRIBUTION MODEL

The service architecture proposed by TINA-C consortium [2] demonstrate three generic aspects of the system:

- *generic User/Provider paradigm* to govern access between stakeholders;
- *flexible session (“call”) model* supporting multimedia/multiparty services;
- *open interfaces* to allow for third party development applications, interoperability among stakeholders, global personal and terminal mobility.

User/Provider paradigm is universally applicable to any situation where a User makes use of service offered by a Provider and defines roles of actors and relationship between them in business scenario.

The User/Provider relationship is separated into “Access part”, concerning the establishment and maintaining of interaction and “Usage part”, concerning the actual use of the capabilities associated with interaction. The separation of Access and Usage parts of service is illustrated at Figure 1.



**Figure 1** Separation of Access and Usage part of User/Provider interaction in multi-player environment

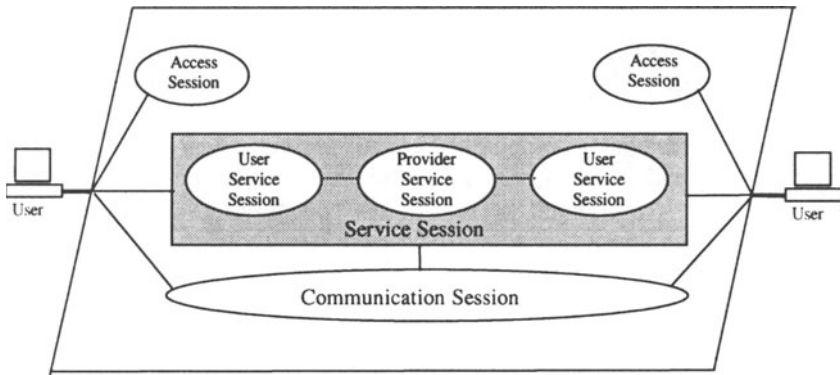
The Access part concerns to the following activities:

- initiate dialogue between User and Provider;
- exchange of domain and equipment information to facilitate service provision;
- establish and maintain a secure, trusted association between User and Provider.

The Usage part is divided into Usage Assistance – use of services to assist the User/Provider association (such as customization of settings, subscription services, accounting and billing, etc.) and Usage Objective - use of services concerning the primary aim of User/Provider association (such as Media on Demand, Video Conferencing, etc.).

We describe relationship between parties in service provision and their evolution over time by terms of sessions, which are defines as context for relating service provision activities between stakeholders and related to the allocation of resources

which are necessary to perform these activities. Three types of sessions are shown in Figure 2.



**Figure 2** Sessions of User/Provider interactions

*Access session*, related to Access part of User/Provider relationship, identifies a secure, accountable, manageable association between Users and provides a context for a User to access services offered by Provider.

*Service sessions*, related to Usage part of User/Provider relationship, provides context for one or more Users to support execution of services. Service session provides the concept of “call” and allows to Users possibility of negotiation through services session, e.g. negotiate what parties are involved, which specific communication configuration are selected, what QoS parameters are required, etc. It is composed of User Service Session part, which represents local view for each user and Provider Service Session part, which represents global view.

*Communication session*, related to Usage part of User/Provider relationship, provides context for one or more Users to support the telecommunication capabilities necessary for service sessions. The mapping of service sessions onto communication sessions is fully flexible (Call/Connection separation).

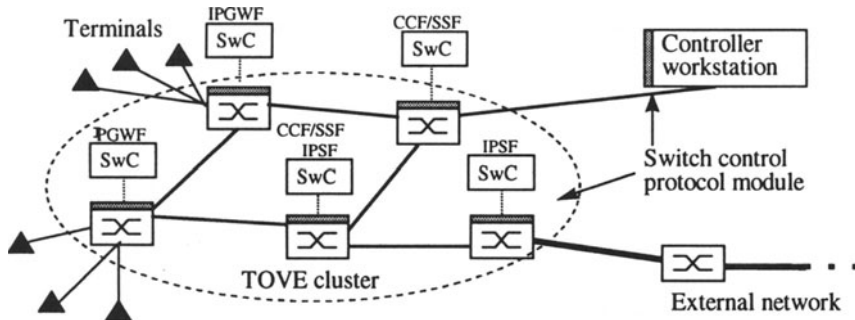
From computational viewpoint this architecture can be described by means of interacting service architecture components - entities in computational viewpoint that can be mapped to computational objects or to sets of interacting computation objects. These components are forming a framework for service development and provision, providing abilities for deriving new service components by existing via specialization (adding new features) and composition (putting together several components).

So, in the scope of service architecture in broadband multimedia distribution network it is important to define both refined architecture model based primary on principles of TINA-C service architecture and generic *Service Components Framework*, that can be used by designers for rapid development of wide range of multimedia provision services through specialization and composition of reusable components.

In next section we consider an example of multimedia distribution scenario for TOVE-based network architecture which will also clarify our vision of the model described above.

### 3 MULTIMEDIA DISTRIBUTION SCENARIO

ATM based networks, such as B-ISDN, is going to be dominant transmission technology on which the information infrastructure will be build. Therefore integrated solutions, where IP routing and switched connections are provided will be needed in future broadband multimedia capable networks. The TOVE network architecture satisfies the requirements listed in the previous sections and provides all necessary facilities to separate control from switching as it shown in Figure 3. Switch controllers (SwC) are used to execute specific Control Functions, e.g. Call Control Function (CCF) and Service Switching Function (SSF) needed in establishing ATM virtual circuits, IP Gateway Function (IPGWF) and IP Switching Functions (IPSF) needed in IP routing.

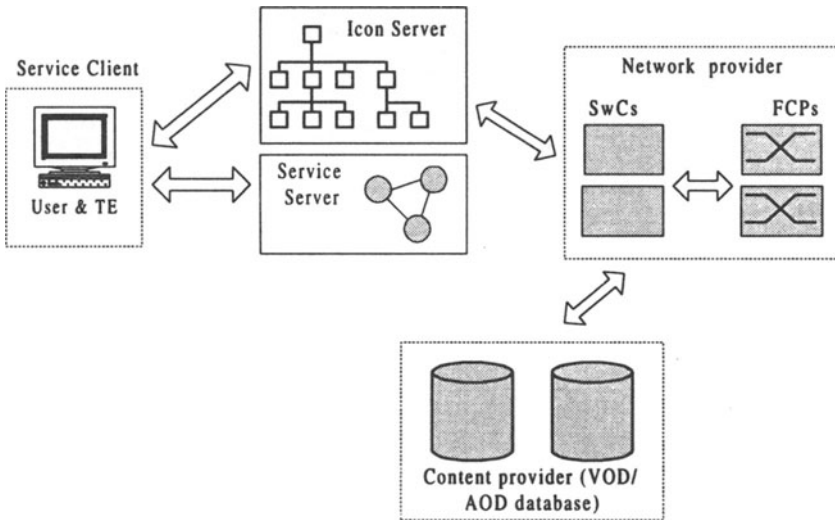


**Figure 3** TOVE architecture

Using the TOVE underlying architecture a multimedia distribution scenario can be proposed where main actors are:

- Service Client;
- Content Provider;
- Icon Server;
- Service Server.

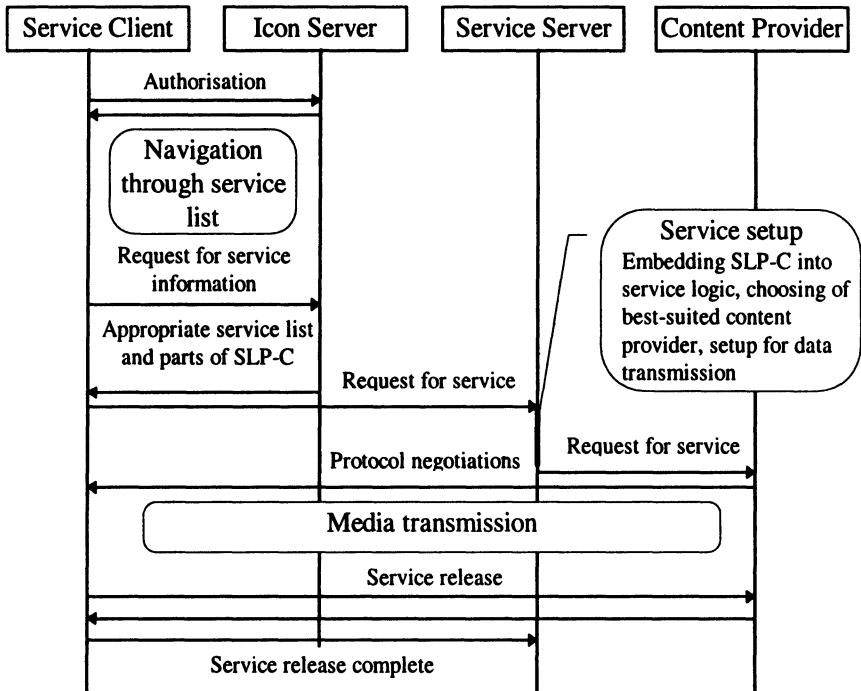
The scenario is based on principles of multimedia service provision model proposed in [6]. The model is described in Figures 4 and 5. The service provision mechanism is illustrated as an example for Multimedia on Demand type services.



**Figure 4** Multimedia service provision system architecture

- The Service Client is represented by User and Terminal Equipment (TE), where the last one is a device with different multimedia capabilities. The TE can be available for the use by many users and provides a level of service that customized to the User.
- The Icon Server represents available services such as AOD/VOD or different multimedia documents.
- The Service Server is a distributed computer controlled system, which provides to Service Client a concrete service with guaranteed QoS parameters.
- The Content Provider is responsible for service content provision on the basis of corresponding databases and for regular renovation of information at the Icon Server environment.
- The Network Provider provides standardized network services as it described in [4].

More details of negotiating between main actors of scenario are shown by means of arrow-type diagram in Figure 5.



**Figure 5** Multimedia service provision scenario example

#### 4 SERVICE COMPONENTS FRAMEWORK IMPLEMENTATION ASPECTS

The Service Components Framework development is based on TINA-C generic principles. However we will not follow strictly the architecture and interfaces solutions provided by the TINA-C and the OMG CORBA. Our approach is based also on the solutions available in Java Platform which at the moment is presented by JavaBeans component architecture [7], Java RMI [8] and JDBC [9] APIs as forthcoming Java Electronic Commerce Framework and Java Media Framework [10]. Our contribution on top of the Java Platform is the development of the middleware for Service Components Framework.

The middleware for Service Components Framework is called JVOPS (JavaVOPS Framework), by analogy with CVOPS [11] and OVOPS [12] Virtual Operations Systems known as sophisticated frameworks for telecommunication applications programming.

## 4.1 JVOPS Basic Concepts

Distributed systems are hard to build. They require careful thinking about problems that do not occur in local computation. The primary problems are those of partial failure, greatly increased latency, and language compatibility. The Java Platform provides a number of tools (e.g. Java RMI, Java IDL, etc.) that support a general approach for distributed computation using techniques natural to the Java language and environment.

These tools, powerful as they are, do not make distributed computation systems easy to design - they merely make them possible to approach. JVOPS is intended to be an environment with fundamental possibilities for Java-based applications in distributed reliable computing system [14 - 16]. JVOPS supports the development of distributed applications, providing transparent communication mechanism, process migration, load balancing and different degree of reliability to application. It can be used by designers for rapid development of wide range of distributed application through specialization and composition of reusable components.

An interactive distributed application is represented in JVOPS as a family of interacting components called *jTasks* by analogy with *vtasks* and *otasks* in CVOPS and OVOPS, respectively. The *jTasks* interact with each other with one way synchronous or asynchronous transporters representing method calls or messages. The transporters are routed through communication channels representing the *jTask* connections. The following abstract classes and principles are used:

- *jTask* with subclasses *jAdapter*, *jMux*, *jBus*, *jProtocol* and *jFactory* representing active components,
- *jTransporter* with subclasses *synchTransporter* and *asyncTransporter* representing synchronous and asynchronous interactions between components,
- *jScheduler* which manages threads and the execution of *jTasks*,
- *jArch* for System Architecture specification and instantiation of *jSchedulers*, *jTasks* and their interconnections in each given subsystem.

Different underlying technologies such as RMI, transactional RMI, CORBA IIOP or are encapsulated in *jAdapter* instances.

The *jProtocol* class may contain an Extended Finite State Machine (EFSM) abstraction and is characterized by the current states of its instance variables set, behavior function and input/output channels. The *jTask* and *jTransporter* classes have been introduced from the OVOPS++ extension of OVOPS [3 - 4]. JavaBeans component architecture can be applied in the implementation of *jTask* subclasses.

## 4.2 JVOPS System Architecture

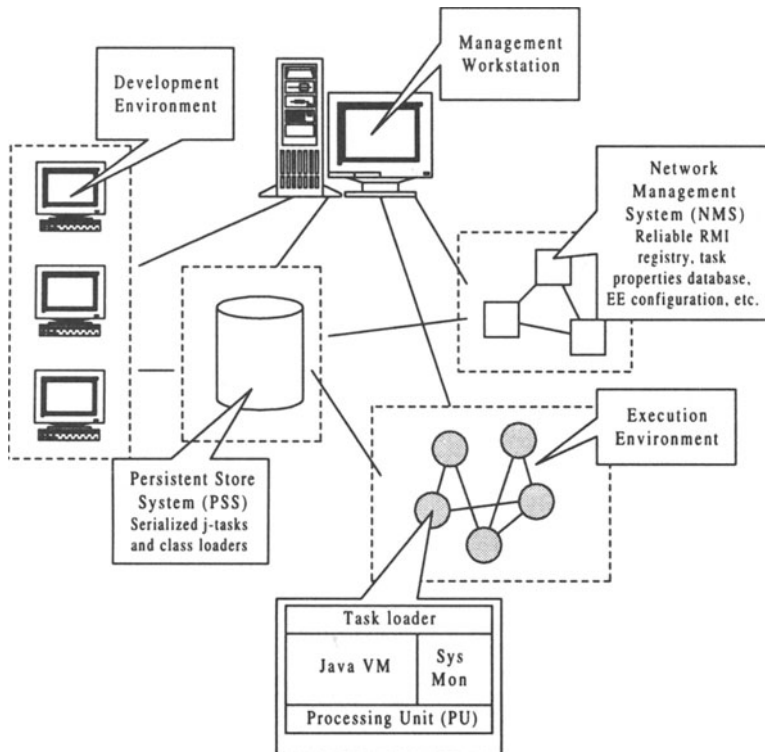
JVOPS System Architecture is shown at Figure 6 and consists of the following subsystems:

- Network Management System (NMS);
- Persistent Store System (PSS);
- Execution Environment (EE);



- Development Environment (DE);
- Processing Units (PU);
- Management Workstation (MW).

Each subsystem in the JVOPS System Architecture contains at least one jArch and jScheduler and the required number of jTasks.



**Figure 6** JVOPS System Architecture

The Execution Environment is represented as a set of PUs which are active at the moment over network of nodes in DRC environment [13].

The PU is a basic computing element of the JVOPS application and is described as a Java Virtual Machine with components depicted in Figure 6.

The Network Management System provides the control of j-task functioning and communicating, load balancing, reliability control, task migration between nodes etc.

The Persistent Store System supports the data storage necessary for all kind of EE system needs.

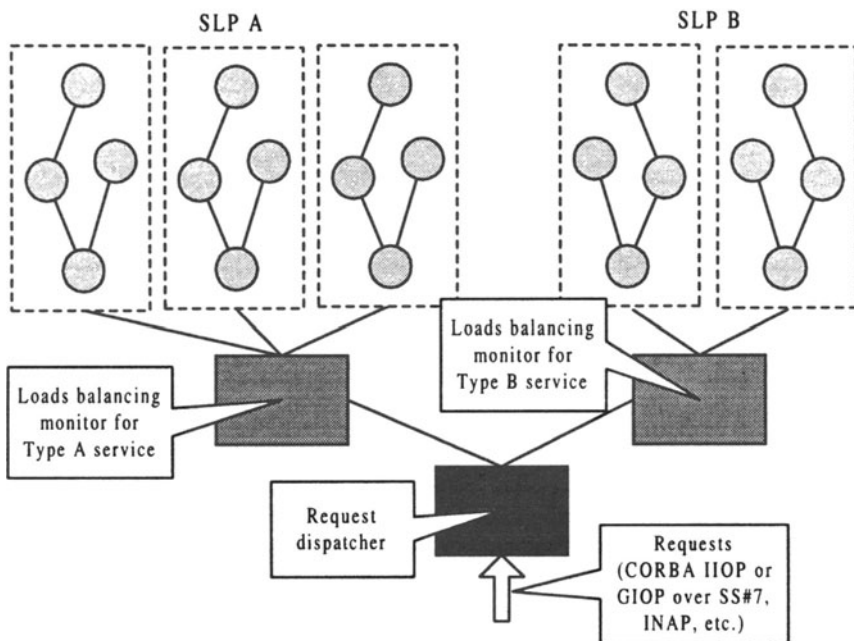
The Development Environment provides the possibility of j-task programming and its integration into application software. After verification and testing j-tasks are to be passed to PSS system and becomes ready for loading into EE system.

### 4.3 JavaSLEE and JavaSCE

An important part of JVOPS concept is a modern way for service creation and execution over DRC environment.

For intelligent applications Service Component Framework should provide Service Logic Execution Environment (SLEE) for service application and Service Creation Environment (SCE) for service modeling and creation. The SLEE and the SCE are forming a basis for Service Component Framework which is built on top of JavaVOPS Framework.

A general architecture of services application in Java SLEE environment is illustrated in Figure 7. The JavaSLEE provides Service Logic Programs (SLP) entities with all means for normal functioning in JVOPS environment , e.g. load balancing , request dispatching etc. Every SLP in JavaSLEE is a set of jTasks with its internal logic and external interfaces. The structure resembles the IN SIBs logic with Points of Initiation and Points of Return as external interfaces.

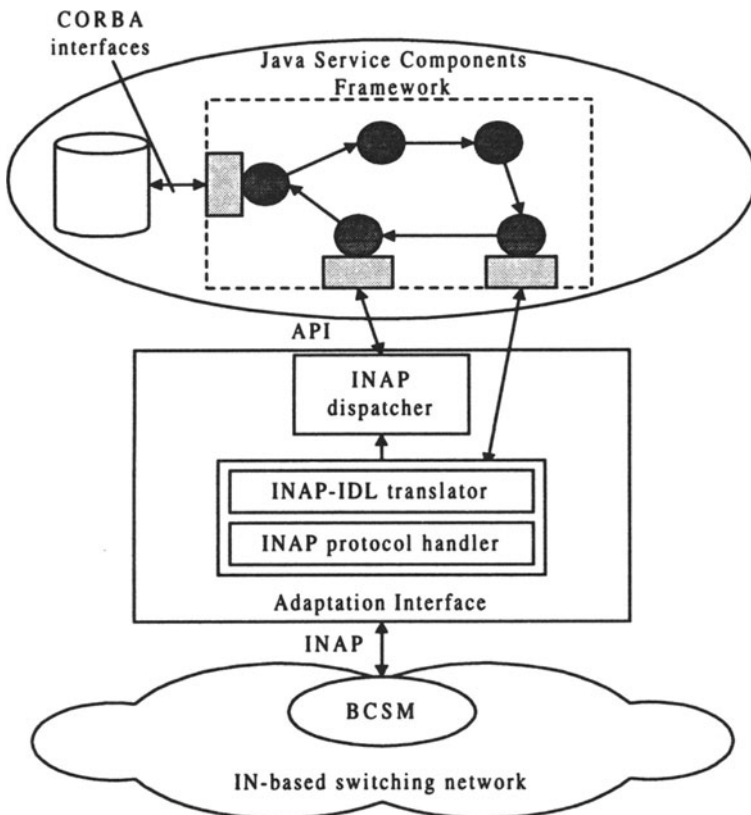


**Figure 7** Services application in JavaSLEE

## 5 EXAMPLE

As an example let us consider the problem of IN integration with new object-oriented platforms. To solve this problem three basic approaches have been proposed in EUROSCOM P508C project [16]. They are

- encapsulation of IN software into CORBA objects;
- the use of external gateway;
- use of SS#7 as transport for interORB communication.



**Figure 8** Example of external gateway architecture in Service Component Framework

Figure 8 shows the Service Component Framework application to the second approach and illustrates the Service Component Framework service architecture on top of IN based-switched network. Such kind of solution can allow to Public Network Operators to keep the existing BCSM and INAP interfaces in operation as long as possible/ required, therefore saving investments into previous technologies.

## 6 REFERENCES

1. ITU-T Recommendation I.312/Q.1201, Principles of Intelligent Network Architecture, 1992, Geneva.
2. TINA-C Deliverable. Service Architecture. Version 4.0. 1996
3. Puro V-M., Koponen P., Rasanen J., Nummisalo P., Martikainen O. (1996) TOVE in Universal Mobile Telecommunications System, *Proc. 2nd Workshop on Personal Wireless Communications (Wireless Local Access)*, Frankfurt am Main.
4. Puro V-M., Koponen P., Rasanen J., Nummisalo P., Martikainen O.(1996), TOVE in Broadband Multimedia. *TOVE report, Helsinki Univ. of Tech.* Helsinki.
5. Martikainen O. et al. (1997) CALYPSO - System Components and Tools for Media Distribution over Broadband Networks. *Project description, Helsinki University of Technology*. Helsinki
6. Martikainen O., Naoumov V., Samouylov K. (1996) Call Processing Model for Multimedia Services, *Intelligent Networks and New Technologies (V.B.Iversen and J.Nørgaard eds), Chapman & Hall*, London.
7. JavaBeans™ API Specification. Version 1.0-A. *Sun Microsystems Inc.*, 1996.
8. Java™ Remote Method Invocation Specification. Revision 1.4, *Sun Microsystems Inc.*, 1997.
9. The JDBC™ API Version 1.10. Part 1 - Interfaces, Part 2 - Classes and Exceptions. *Sun Microsystems Inc.*, 1996.
10. Java™ API Overview. URL: <http://www.javasoft.com/products/api-overview.html>
11. Malka J., Ojanpera E. Reference Manual for CVOPS 4.0 (1992). *Technical Research Centre of Finland, Telecommunications Laboratory*. Helsinki
12. Martikainen O., Puro P., Sonninen J. (1995) OVOPS, Object Virtual Operations System for Distributed Application Development. *Proc. INDC 94*, Funchal, Portugal.
13. DRC Project Home Page. URL: <http://mds.jpl.nasa.gov/drc>
14. Anderson T. E., Culler D. E., Patterson D. A., and the NOW Team (1995) A Case for NOW (Networks of Workstations). *IEEE Micro*, Vol. 15, No. 1, February 1995.
15. Ingham D. B., Caughey S. J., and Little M. C. (1997). Supporting Highly Manageable Web Services. *Proceedings of the Sixth International WWW Conference*, Santa Clara, USA..
16. CORBA as an Enabling Factor for Migration from IN to TINA: A EURESCOM P508 Perspective. Final Draft, Version 4.1, *EURESCOM*, 24 December 1996.

## 7 BIOGRAPHY

**Olli Martikainen**, PhD from Helsinki University and MS from Helsinki University of Technology. He has been doing research in several positions in Helsinki University of Technology (1976 - 1982, 1991 - 1997), Oxford University (1980 - 1981), Technical Research Centre of Finland (1982 - 1985, 1989 - 1991), Nokia Electronics (1985 - 1986), Nokia Research Centre (1986 - 1988), Lappeenranta University of Technology (1988 - 1989). In 1991 - 1996 he has been research director and vice president at Telecom Finland. Currently he is professor at Helsinki University of Technology. His main areas of interest are telecommunication software methods and tools, network architectures, performance analysis and new industrial and economic structures in telecommunications.

**Valeri Naoumov**, PhD from Computer Centre of Russian Academy of Sciences (1979) and MS from Peoples' Friendship University (1972). He has been doing research in several positions in Peoples' Friendship University (1973 - 1996), Helsinki University of Technology (1984 - 1985), Institute for Problem of Information Transmission of Russian Academy of Sciences (1992 - 1994), Deutsche Bundespost Telekom (1994 - 1995). Since 1996 he is professor at Department of Information Technology of Lappeenranta University of Technology. Areas of expertise: queuing theory, traffic models, programming, computational algorithms, intelligent network, telecommunication protocols implementation. He has written many technical articles and two books.

**Konstantin Samouylov** received the MS degree from Peoples' Friendship University of Russia and the PhD degree from Moscow State University in mathematics in 1978 and 1985 respectively. From 1986 to 1992 he taught courses in queuing theory and conducted research in CCS network, teletraffic and performance analysis. Since 1993 he began work in broadband and intelligent networks. Dr. Konstantin Samouylov currently is a head of Telecommunication Systems Laboratory in PFU, Moscow. He has written a number of journal and conference papers in the areas of queuing theory, performance of CCS network and IN software development.

**Michael Zhidovinov** received his MS degree from Moscow State University in mathematics in 1994. Since 1993 he has been doing research in several positions in Peoples' Friendship University, Moscow. His main areas of interest are telecommunication software methods and tools, network architectures, software engineering.