# Service Execution Environment for Digital Multimedia

*Juhana Räsänen, Petteri Koponen, Olli Martikainen*

*Laboratory of Telecommunications Software and Multimedia*
*Helsinki University of Technology*
*{Juhana.Rasanen, Petteri.Koponen, Olli.Martikainen}@hut.fi*

## Abstract

Broadband networking concepts and technologies, such as open signalling, IP switching and xDSL technologies are experiencing a rapid evolution whereas the service concept in traditional ATM networks has not evolved at the same pace. As opposed to the broadband IN-based approach we have composed one vision of a flexible broadband media services and network architecture.

The key issue is flexibility in the creation and deployment of new services by service providers that operate independent of the network operator. This can be achieved by introducing an architecture where an open network control interface and a Service Execution Environment (SEE) are provided for service applications. In this environment the service itself can control the network to set up media service sessions.

A Java-based SEE is suggested for implementing such an environment. The Java SEE would provide the basic functionality (such as media stream set-up/teardown, user and service authentication and billing) on top of which the service providers could implement their services. This enables a flexible distribution of service logic between clients, servers and network nodes.

## 1. Background

The Telecommunications Software and Multimedia Laboratory of Helsinki University of Technology has studied ATM networks and technology in the TOVE project [1], which has produced a prototype of an ATM switch controller with an ITU-T compliant UNI (User-to-Network Interface) signalling during 1996. The signalling software is run in an external workstation, which is connected via a 155 Mbps link to an ATM switch. The prototype software is implemented using object-oriented methods in the Linux environment. A further goal for the project is to add NNI (Network Node Interface) signalling and IN (Intelligent Network) functionality to the prototype. Currently the software is available for the project partners only, but it will become publicly available after the project is finished.

## 2. Motivation

Our hands-on experience of ATM signalling and the delayed standardisation of an IN service model for B-ISDN (Broadband Integrated Services Digital Network) as well as concerns of its applicability to heterogeneous digital multimedia services networks encouraged us to seek alternatives. We list below some trends in broadband networking that in our opinion call for a new, more flexible network architecture than the standard IN based B-ISDN/ATM network.

- *Domestic broadband access* over fixed and wireless networks is expected to become commercially feasible within 5 to 10 years, when access network technology and transmission capacity costs decrease enough. This opens a completely new market for the distribution of a variety of services over a single network. The services range from Internet connectivity to basic telephony service, TV, VoD and audio stream distribution as well as to some completely new applications like interactive television. Also mobile access is likely to be a part of this networking environment.

- *Internet* along the TCP/IP protocol suite and applications on top of that can be listed as an independent trend due to the momentum the Internet has gained. As the Internet can be considered the de facto technology of global networking, there is little doubt that also future networks should provide the same services, functionality and coverage as the Internet does today.

- *Heterogeneous networking* in terms of services, protocols, control functions and devices is the result of the two trends described above. A very flexible network architecture is essential for supporting a combination of a digital multimedia distribution network and a global Internet-like packet-switched network. ATM addresses this requirement at the data transport level by providing several traffic classes and the QoS (Quality of Service) concept for applications with different requirements. However, the telecommunications oriented call and connection control model of B-ISDN does not lend itself easily to Internet-like networking and the introduction of new features is difficult due to rigid and complex protocol stacks. The rigidity can be seen even in the more traditional fields of telecommunications, an example of which is adding mobility functions to B-ISDN, which has proven difficult [2].

We conclude that although ATM does provide a cost-efficient data transmission technology, the network architecture should be designed for easier creation and deployment of new services, which would help the service and software industries develop. Fortunately such a trend can be seen within the industry and research community. We list below some issues that in our opinion indicate an ongoing change in the telecommunications infrastructure.

- The separation of hardware switching functions from the call and connection control functions in ATM networks is being studied by several groups [1, 3, 4]. It is an important factor in creating a flexible network infrastructure, because by implementing the control functions independently of the physical branching, routing or switching equipment, the same hardware can easily be used in different applications.

- The field of broadband networking is expanding from the business sector and corporate networks to the mass market of domestic access. Several manufacturers and network operators already study xDSL, CableTV and ATM integration in this context. At the same time the role of the service and content providers is growing, which can be seen in some pilot projects carried out in the Internet, e.g. MediaNet by Telecom Finland Medialab [5]. Another factor here is the opening of the telecommunications competition in Europe in 1998, which will let new, innovative players enter the market. Also the existing operators must be able to adopt new technologies and business practises as well.

- The growth rate of the Internet indicates the demand for a global digital multimedia services network. ATM has already been successfully deployed into the core and into some customer premises networks of the Internet, but as high-speed networking gradually migrates closer to the edges of the network, the limitations of the present IP over ATM solutions become clearer. Seamless integration of the TCP/IP suite and ATM technology becomes necessary, and some innovative solutions for this have been developed [6, 7].

- The field of telecommunications software engineering is adopting new technologies and a new generation of communications software middleware (e.g. [8]) and development tools is emerging. Promising approaches grow from object-oriented methods and component software models (design patterns, frameworks, OMG CORBA) and the distribution of intelligence using mobile code (Java) [9, 10, 11, 12]. Software development also benefits from the separation of control and switching discussed above, because standard computing environments and development tools can be used instead of proprietary embedded systems development tools.

The rest of this document will tie together the loose ends outlined in Chapter 2 by addressing the problem of creating a platform for a broadband services network. The goal is to facilitate the service and user interaction and to build service creation, management and accounting subsystems as well as authentication, security and billing mechanisms. The platform provided by the network operator should be open so that it would be easy to create and deploy new services and content by independent service operators. Chapter 3 describes a layered control architecture at a conceptual level and the model is applied in Chapter 4, which gives a description of a proposed implementation of an open broadband network platform.

## 3. Three-layer Control Model

One of the trends in the broadband networking listed in Chapter 2 was the separation of control from routing and switching. So far the primary goal has been to extract the signalling[1] software from the switching fabric to an external workstation. This increases both the flexibility and scalability of ATM networks by allowing any standard workstation to be used for running the signalling software. It is easy to

---

[1] By 'signalling' we refer to the general procedures of opening and managing virtual connections through an ATM network rather than the specific procedures of standard ATM signalling.

upgrade an external switch controller if the signalling load increases and it is even possible to let more than one workstations control one switch or a single workstation control a cluster of switches. The fact that the controller workstation can be physically separated from the switching fabric allows the switches to be very simple without any moving parts like hard disks, which reduces the maintenance needs.

The above separation of switching and control is adequate for separating specialised switching hardware functions from the signalling functions that can be implemented with specialised software running on general-purpose hardware. To further structure the software part needed to implement the signalling functions, we define three distinct levels of control. Individual switching fabrics are controlled at the Fabric Control Layer (FCL), end-to-end connection control resides at the Network Control Layer (NCL) and overall service control is taken care of at the Service Control Layer (SCL). (Figure 1)

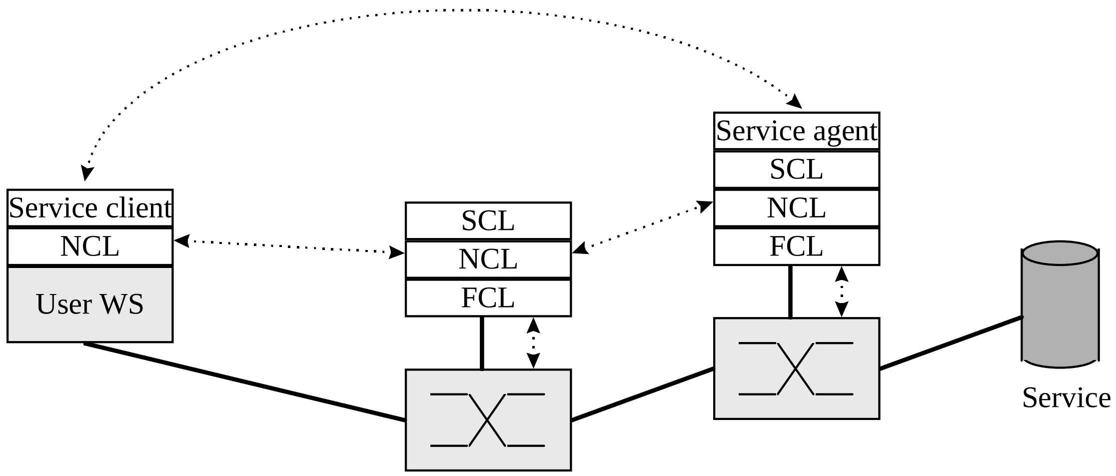


**Figure 1** Layered control model for broadband networking

### 3.1 Fabric Control Layer

FCL is closest to the switching hardware, its function being to provide the network layer a hardware independent control interface for managing connections in an ATM switch. The communication at the FCL occurs only locally between the switching hardware and the software controlling it. FCL might be implemented for example as a wrapper to GSMP (General Switch Management Protocol) [13] or some proprietary switch control interface provided by the switch vendor.

### 3.2 Network Control Layer

Whereas FCL is concerned with the control of individual switching fabrics, NCL handles end-to-end connections which consist of chains of connection legs between network nodes. Connections may be simple point-to-point virtual channels, or they may branch (point-to-multipoint) or become rerouted from a physical link to another (handovers, mobility). The communication at the NCL occurs between peer nodes, and an implementation should provide at least a routed network layer protocol for transporting data between network nodes, a routing protocol to exchange network topology information and a protocol for reserving and establishing connections through the network.

### 3.3 Service Control Layer

The highest level of control is the SCL where entire service sessions are managed. Service sessions consist of one or more individual connections (e.g. audio and video streams together with a bidirectional data connection could form one multimedia service session) and they typically include user authentication in the beginning as well as other necessary negotiations to set up the session. The communication at the SCL occurs between service clients and service agents, which in turn use the connection set-up service of the NCL to establish connections. We point out that at this level the communication methods between clients and agents can be service specific instead of using predefined protocols like INAP (Intelligent Network Application Part), because the service data is transported across the network transparently using the general transport service of NCL.

## 4. Calypso Architecture

The Laboratory of Telecommunications Software and Multimedia has started a new project called Calypso, aiming at developing a prototype implementation of a network architecture that was outlined in the previous chapters. The intended application domain of the Calypso architecture is in providing broadband access for domestic and small-and-medium-sized business customers. This domain has been the main focus of the design due to the reasons elaborated in Chapter 2, and it must be emphasised that Calypso is not intended to be a global replacement for the IN/B-ISDN architecture.

The design has started from service scenarios and progressed towards an architecture that would realise those scenarios. One important goal has been to reuse existing, preferably publicly available software components and designs as much as possible to enable rapid prototyping of the proposed ideas. This means that the research efforts will be directed to getting experience from early prototypes, leaving less concern for non-critical performance and quality issues of individual subsystems, which in our opinion are solved better after the general design has been verified and found reasonable. In practice the initial goal is to assemble a complete working system in which some elements like routing or connection admission control may be less than optimal.

### 4.1 Calypso Network and its Elements

Figure 2 gives an example of a Calypso network subsection. A typical Calypso network would be a relatively small (in the order of hundreds of users) customer access network consisting of a group of access switches and a connection to a high-performance backbone network. The users would be connected to the network using for example xDSL modems over the telephone wire. Typically the user terminal device would be a set-top-box or a PC with similar functionality. One access switch should be able to handle at least 100 users, probably more, so there could be for example one switch located in the cellar of each block of flats, and the switch controllers would be located remotely in the network operator’s premises. The backbone network is beyond the scope of the Calypso architectural domain, but it could be for example a very-high-speed all-optical network using WDM (Wavelength Division Multiplexing) or other photonic switching technologies. The performance should be sufficient to serve a large metropolitan area.

The Service Execution Point (SEP) is the central network element of the Calypso architecture. It is the physical entity (workstation) that executes all the software components, excluding a minimal control and management module integrated into the switching fabrics (e.g. GSMP slave module). Typically there is one SEP per fabric, but other configurations are also possible using the VE (Virtual Exchange) concept[2] developed in the TOVE project. SEP includes all the control layers introduced in Chapter 3. The proposed implementations are described below.

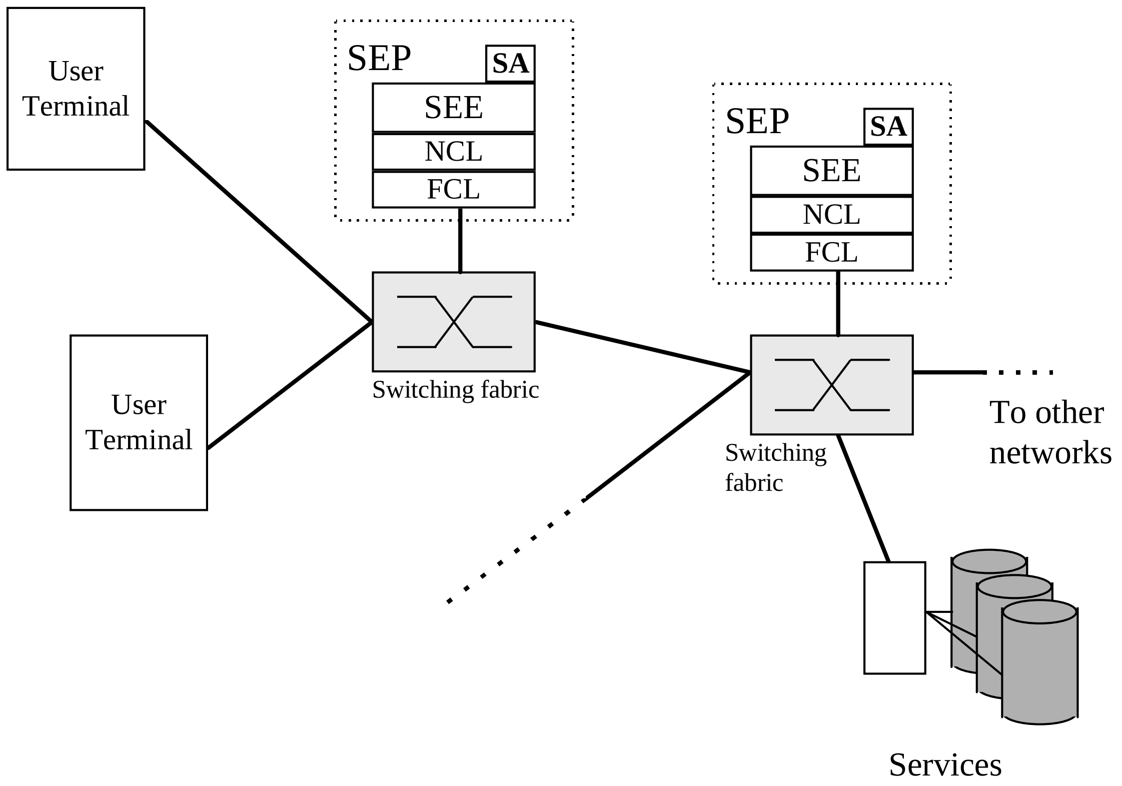


**Figure 2** An example of a Calypso network

## 4.2 Calypso FCL

The VE module developed in the TOVE project will be reused in Calypso FCL. Currently only experimental FSR (Frame Synchronised Ring) hardware developed in the Technical Research Centre of Finland [14] is supported, but also GSMP support for commercially available fabrics is under development.

---

[2] A cluster of fabrics can be managed by a single VE instance that provides a view of a single, virtual switch for the higher software layers. Thus, using VE as the Calypso FCL, a single SEP is able to manage one or more switching fabrics.

### 4.3 Calypso NCL

Calypso NCL is based on the reuse of the TCP/IP protocol suite. The idea is to overlay an IP network on the Calypso network, so that all network nodes and user terminals can be addressed using IP addresses and control data transported between them over TCP/IP connections. Initially we plan to implement this by configuring SEPs to act as IP routers that connect to each other and to the user terminals via Classical IP over ATM [15] PVCs. We point out that to keep the IP performance requirements of the SEPs relatively low, the NCL IP network is used only for transporting control data. The IP routing service to the Internet can be implemented using a separate set of VCs that are switched directly to the dedicated routing server.

Using IP has several immediate benefits. First of all, we can reuse well-understood technology with existing and extensively tested implementations instead of making our own implementation of network protocols. All standard IP tools are also available, e.g. DNS (Domain Name Service), routing protocols, SNMP (Simple Network Management Protocol), HTTP (HyperText Transfer Protocol) so there is no need to find equivalents for them.

In addition to providing a routed network protocol, one requirement of the NCL was to provide a protocol for establishing and managing end-to-end streams through the network. Our idea is to use routed IP so that the stream establish requests are carried in IP packets from the ‘subscriber’ node to the ‘provider’ node, and *before any SEP forwards the request, it reserves resources for the stream from the switching fabric it is controlling*. In the case of multicast the request is forwarded until a switch already receiving the multicast tree is found, at which point a new branch can be connected to the existing tree. This stream request method is similar to the RSVP (Resource Reservation Protocol) under study in IETF [16] and we are following the development of RSVP and will study its applicability in our work.

### 4.4 Calypso SCL

The SCL layer implementation in Calypso architecture is called SEE (Service Execution Environment). The SEE is a platform for the management and execution of Service Agents (SAs). In SEE's context the services include both the supplementary or value-added services and most of the traditional bearer services such as signalling. This means that in addition to services, also the control functions can be updated, introduced, and removed dynamically, which enables domain- and service-specific control functions. These control functions, or SAs in the Calypso architecture, could implement handover and location update procedures supporting mobility, or they could include light-weight procedures for establishing VCs that could be used in redirecting media streams in residential access networks.

In the Calypso architecture the SAs are Java programs managed and executed by the SEE. The management includes, for example, providing means to distribute the SAs safely over the network, and allowing them to access a SA-specific subset of the SEE's resources. The SEE also executes the SAs and monitors their execution. The management and execution functions are strongly interconnected; for example, the SEE checks at run-time whether an SA is allowed to access a resource depending on the security level the management functions have given to the SA.

The SAs and most of the SEE are going to be implemented with Java due to its support for code mobility, native TCP/IP support, and a wide range of development tools available. The SEE will provide the SAs with a class framework, SA repository, authentication and other security functions etc. However, we intend to keep the SEE relatively lean so that the SAs would implement most of the SEP's functionality.

The SAs are either fixed or mobile. The fixed SAs cannot be transferred over the network and they can include native (non-Java) code because of efficiency reasons or existing implementations that are to be reused. However, also the fixed SAs implement the interfaces required for management and monitoring purposes, that is, the native code must be wrapped in the corresponding Java classes. The mobile SAs are more flexible in the sense that they can be dynamically injected into the SEE, although this requires quite advanced security mechanisms. How these mechanisms will be implemented is still an open question at this phase of the project.

The user's interface to the services is expected to be browser-based. Users access the services from their personal service pages that are stored by the network and updated when users purchase access to new services. Services are represented as service icons, which encapsulate Java programs that are able to communicate with the respective SA located in the SEE. This implies that the SAs using HTTP, which means that an HTTP server becomes an integral part of the SEE. One interesting piece of work that could be used in implementing this is the World Wide Web Consortium's Jigsaw [17], which is an extendible web server written entirely in Java.

## 5. Conclusions, Future Work

The concept of Calypso-like multimedia services and service icons has already been presented already in [18, 19], but due to the slow progress of broadband IN architecture standardisation there has been no practical platform on which the ideas could be experimented. Calypso introduces a completely new broadband service architecture which uses existing components whenever possible to provide an open platform for rapid service creation and deployment.

Spring 1997 has been the design phase of the architecture and the implementation is scheduled to start during summer 1997 and we hope to complete a preliminary Calypso platform by the end of 1997. This prototype would, of course, lack most of the non-critical features like authentication, billing, etc., but it should be able to establish on-demand connections using the IP based implementation outlined in Chapter 4. A further application of Calypso architecture will be in the MediaPoli project, in which a broadband access network for the 1000-3000 customers at the Helsinki University of Technology campus area will be created. The MediaPoli network is open for research projects, hardware and software vendors and service providers for piloting new services and technologies in a real-life environment with a demanding, technology-oriented user base.

## References


[1] Puro V-M, Koponen P, Räsänen J, Nummisalo P, Martikainen O. *TOVE in Universal Mobile Telecommunications System*. Proceedings of the IFIP workshop on Personal Wireless Communications, Frankfurt, December 1996, Verlag der Augustinus Buchhandlung (Aachen), pp. 103-111.

[2] Mitts H. *Architecture for wireless ATM*. Licentiate thesis, Helsinki University of Technology, Espoo, September 1996

[3] Lalgudi H, Chen X, Kumar V. *A generic software platform for local ATM networking*. Proceedings of IEEE Globecom '95, November 1995, pp. 1967-1971.

[4] Leslie I, Crosby S, Rooney S. *The Distributed Control of ATM Networks Project*.

`<URL:http://www.cl.cam.ac.uk/Research/SRG/dcan/>`

[5] Telecom Finland Medialab. *Telecom Finland TF - MediaNet*.

`<URL:http://www.ml.tele.fi/>`

[6] Newman P, Minshall G, Lyon T, Huston L. *IP switching and gigabit routers*. IEEE Communications Magazine, January 1997, pp. 64-69.

[7] Rekhter Y, Davie B, Katz D, Rosen E, Swallow G. *Cisco Systems' tag switching architecture overview*. IETF RFC 2105, February 1997.

[8] Lazar AA, Lim KS, Marconcini F. *Realizing a foundation for programmability of ATM networks with the binding architecture*., IEEE Journal of Selected Areas in Communications, September 1996, pp. 1214-1227.

[9] Gamma E, Helm R, Johnson R, Vlissides J. *Design patterns, elements of object-oriented software*. Addison-Wesley 1995.

[10] Hüni H, Johnson R, Engel R. *A framework for network protocol software*. OOPSLA 1995.

[11] Martikainen O, Puro P, Sonninen J. *OVOPS, Object Virtual Operations System for distributed application development.* Proceedings of INDC 94, Funchal, April 1994.

[12] OMG. *The Common Object Request Broker: architecture and specification, revision 2.0,* Object Management Group, Framingham, MA, July 1995.

[13] Newman P, Edwards WL, Hinden R, Hoffman E, Ching Liaw F, Lyon T, Minshall G. *General Switch Management Protocol specification, version 1.1*. IETF RFC 1987, August 1996.

[14] Raatikainen P. *Analysis and implementation of a high-speed packet switching architecture - the frame synchronised ring*. Doctorate thesis, Helsinki University of Technology, Espoo, June 1996.

[15] Laubach M. *Classical IP and ARP over ATM*. IETF RFC 1577, January 1994.

[16] IETF. *Resource reservation setup protocol (rsvp) charter.*

`<URL:http://www.ietf.cnri.reston.va.us/html.charters/`
`rsvp-charter.html/>`

[17] World Wide Web Consortium. *Jigsaw overview*.

`<URL:http://www.w3.org/pub/WWW/Jigsaw/>`

[18] Martikainen O, Molin K. *Broadband intelligent network project*. Proceedings of the IFIP workshop on Intelligent Networks, Lappeenranta, August 1994, Chapman & Hall, TJ Press, pp. 270-279.

[19] Martikainen O, Naoumov V, Samouylov K, *Call processing model for multimedia services*. Proceedings of the IFIP conference on Intelligent Networks and New Technologies, Copenhagen, August 1995, Chapman & Hall, TJ Press, pp. 241-251.