

Submission to the Broadband & Multimedia Workshop  
November 11-12, 1996, Zagreb, Croatia

Cover Sheet

**Paper code number**

P14

**Title**

Multipurpose Middleware for Broadband Multimedia Applications

**Authors**

István Cselényi,  
Christer Gisgard,  
Péter Haraszti,  
Alexander Latour-Henner,  
István Szabó,  
Gergely Záruba

**Corresponding author**

István Cselényi  
High Speed Networks Laboratory,  
Technical University of Budapest, Dept. of Telecommunications and Telematics,  
H-1111 Budapest, Hungary, Sztoczek u. 2. 208.  
Tel: +36-1-4633110  
Fax: +36-1-4633107  
Email: cselenyi@ttt-atm.ttt.bme.hu

# Multipurpose Middleware for Broadband Multimedia Applications

István Cselényi<sup>1</sup>, Christer Gisgard<sup>2</sup>, Péter Haraszti<sup>1</sup>,  
Alexander Latour-Henner<sup>2</sup>, István Szabó<sup>1</sup>, Gergely Záruba<sup>1</sup>

<sup>1</sup>Technical Univ. of Budapest, Dept. of Telecommunications and Telematics,  
H-1111 Budapest, Hungary, Sztoczek u. 2. cselenyi@tnt-atm.tnt.bme.hu

<sup>2</sup>Telia AB, Telia Instrument, S-123 86 Farsta, Sweden, Email: alexander.latour@telia.se

**ABSTRACT:** *ATM networks are currently being installed and tested as part of pilot projects. The various functions of the network components must be investigated under a wide range of operating conditions [1, 2]. One of the key functions in B-ISDN is signaling. Because of the flexibility of an ATM based B-ISDN the signaling functions will be more complex than in existing systems. Before the introduction of a new multimedia service, its resource requirements should be investigated [3, 4, 5]. There are local resources like memory, disk requirement and computational power, and global resources like buffer size of network elements, allocated bandwidth, etc. The recent multimedia applications are inadequate for testing the requirements of local and global resources, since they focus on the user side and do not provide alterability in resource allocation. Therefore a new platform is required on the ATM terminal which can use any global and local resources in an efficient way.*

*The aim of our work was to design and develop a flexible environment which is adequate to make experiments on the requirements of new multimedia services from the terminal's perspective. The main features of the system and some considerations made during the development of the first realized service, video conference are described.*

**Keywords:** distributed multimedia, ATM, QoS analysis, network performance, ATM signaling

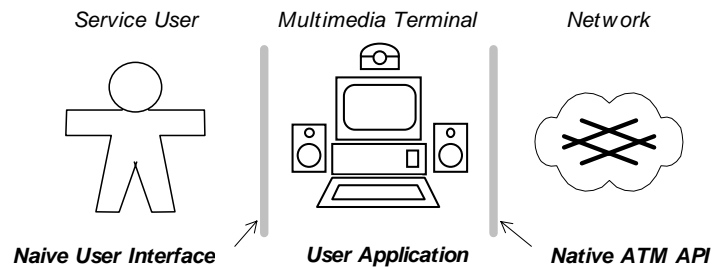
## 1 Introduction

*Experimental Multimedia Middleware for ATM (EMMA)* is a platform which supports experimental work focused on service and call control functions of a multimedia terminal. It complements the *Signaling Emulation (SIGNE)* project at Telia with components necessary for set-up and management of multimedia services [6].

The goal of EMMA project is to extend the functionality of an ordinary multimedia terminal, i.e. a workstation equipped with multimedia tools and an ATM interface card, at several levels in order to investigate several areas, such as local/network resources and signaling protocols required by the service and call management.

At *application level*, several user applications are required, which have a wide range of alterable teleservice and bearer service parameters for experiments. The telecommunication service and the multimedia application representing it to the user are described by a common modeling methodology. At *presentation level*, an easy-to-use naive user interface is developed for users, who are not familiar with the meaning of parameters determining the ordered service. At *transport/network level*, a native ATM extension is necessary, which makes it possible to by-pass the TCP/IP or UDP/IP protocol stack for raw information flow (audio, video), i.e. provides a direct access to ATM. As an other extension at this level, the ATM shaping rate is to be adjusted adaptively, according to the behavior of the traffic source and the multimedia application.

EMMA is designed according to the considerations above. Its main structural parts are a *Naive User Interface (NUSI)*, an *User Application* and an EMMA middleware with *Native ATM Application Programming Interface (NAPI)*, as it is depicted in Fig. 1. The NUSI is designed in JAVA, since the object-oriented methodology and the WWW approach are beneficial for an upper layer protocol. Video conference with White Board is the first user application built for EMMA. The NAPI is based on a specific ATM card from Fore Microsystems, which provide traffic shaping and a programming interface.



**Fig. 1.** Main parts of the EMMA terminal

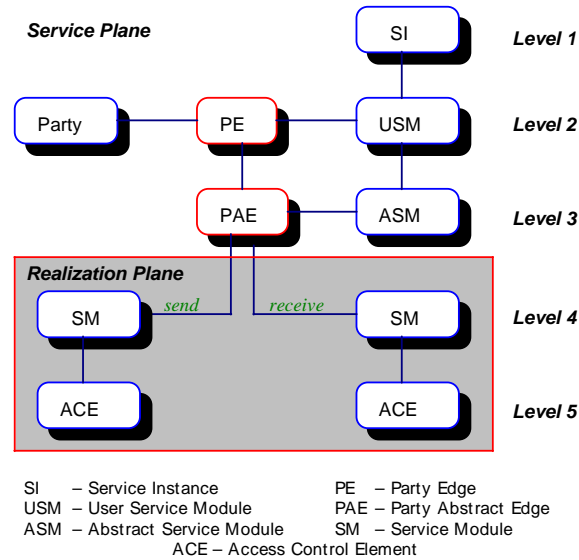
## 2 Flexible Service Description

The B-ISDN TeleCommunication Services (TCS) should be analyzed in order to evolve a general model, which can form the basis of services provided on the EMMA/SIGNE platform. Our model is based on the activities of RACE R2044 Multiservice Applications Governing Integrated Control (MAGIC) project [7], which was extended by Telia. This model is built up from the media representing the service and their relations and contains the role, authority for call modification, and relation to the media of participants. The way of storage and transport of information is also defined. The description of the service contains the processes used for call set-up and call management. The available broadband services are classified and the description of a service from each service class is given in the RACE MAGIC document. The RACE MAGIC methodology provides a good basis both for design of signaling system and test of B-ISDN services.

### 2.1 Service Description Framework

There is a set of attributes defined in RACE MAGIC model for service description. The service is built up from service objects which comprise several attributes. The service objects – which represent class definitions according to the *object oriented methodology* – are placed into five levels in a hierarchy.

When the service is realized as an actual call named Service Instance (SI), then the attribute types will also be realized as an instance of attribute. Figure 2. presents a model of the generic Telecommunication Service Description (TCSD). It shows the objects and their relationships.



**Fig.2.** Telecommunication service description

There are optional and mandatory service objects. An object can be realized if every mandatory object has been built up in the hierarchy below it. The service objects are given in the following list. The numbers denote the hierarchy level. The three upper layer form the *service plane*, the two lower the *realization plane*.

1. Telecommunication Service Layer contains attributes used to describe the overall operation of the service.
- 2.a Party object is used to describe the different behavior of the various types of a service. Different parties can use different media.
- 2.b User Service Module is the basic building block of a service. It provides a typing function combining one or more information types of a service into a single grouping understandable to a user.
- 2.c Party Edge is used to define the relationship between the Party types that may take part in the service and the USM that comprise the service.
- 3.a Abstract Service Module represents the basic information types that may be used in a multimedia service and essentially encapsulates the attributes of the application layer of the service.
- 3.b Party Abstract Edge object is used to describe the relationship between the ASMs and the Party Edge, and maps send and receive to SMs.
4. Service Module represents the protocols used to implement a basic information type of a multimedia service. Protocols from the ATM Adaptation Layer to the presentation layer are specified. The SM is used to implement the service indicated in the ASM.

5. Access Control Element defines the characteristics of the bearer connection that is used to transport a SM across the access. It encompasses the user information protocols in the protocol stack of the Physical Layer and the ATM Layer.

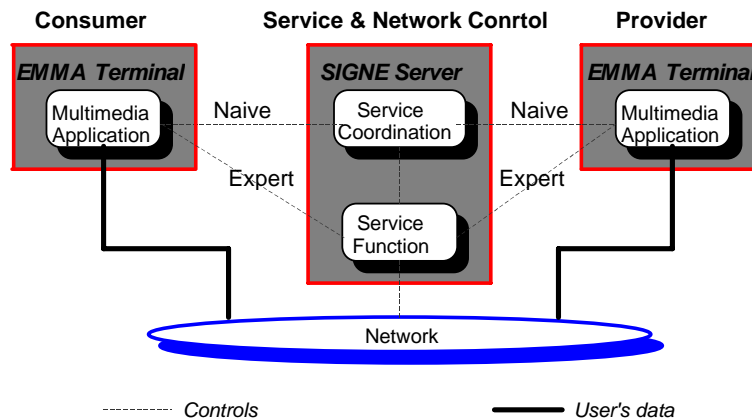
### 3 Flexible Terminal

The development of telecommunication moves toward the evolving of a global information network, which provides beside the bearer phone services advanced forms of human-human communication, education and entertainment possibilities and enables to the people to work and arrange everyday business, like shopping and banking, from home.

Since Internet is said to have a very good chance to take the role of the global information network, it is important that the EMMA terminal can access the server of network operator (SIGNE) through the Internet. The World Wide Web is more and more popular every day and the majority of users has good skills in the empire of HTML links. The appearance of Java enables the creation of interactive WWW pages, thus it is a natural choice that EMMA provides a user interface developed in Java for the customers.

EMMA is realized on a general purpose workstation. This system independence can be increased by selecting a flexible software environment, i.e. Java.

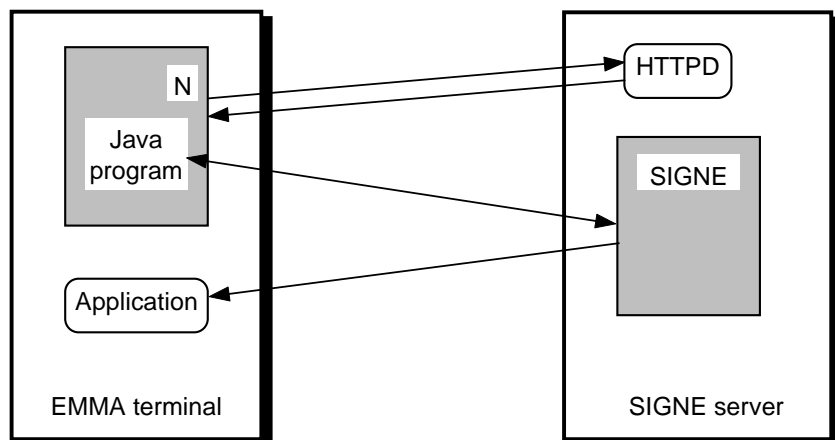
The SIGNE project is focused only on the service and network control. The "Provider" is the provider operating the service and network control, i.e. not the provider of multimedia services or the network itself. The "Consumer" represents any actor using the service and network control. See Fig. 3.



**Fig. 3.** Service and network control entities and their relationships

The operation of EMMA consist of the following main steps (Fig. 4):

- The user access to the server of the network provider by his WWW browser, selects and orders a call. SIGNE provides a user interface implemented in Java to help the naive customer.
- According to the order and the available resources, ATM connections are built up among the EMMA terminals of the participants.
- The applications required for the ordered service are started by a "bouncing mechanism" which is necessary because of security purposes. The terminals use ATM links for transferring the multimedia information.



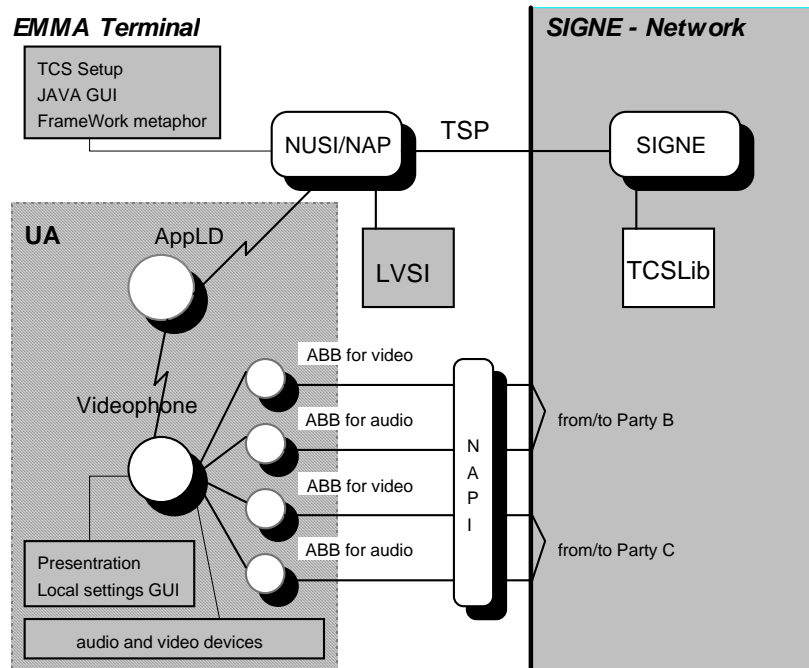
**Fig. 4.** *EMMA system environment*

### 3.1 Parts of the EMMA Project

The structure of EMMA terminal and the relationship between its parts is shown in the Fig. 1. The main parts are:

1. Naive User Interface/Network Access Program (NUSI/NAP) for service consumer and provider in order to easy use of the User Applications (UAs).
2. *User Application* (UA) providing several multimedia services to the user.
3. *Native ATM Application Programming Interface* (NAPI) which provides a network access to the UAs. The NAPI is based on the basic functions of the Fore System's SBA-200E ATM adapter card.

The soul of EMMA is an ordinary multimedia workstation with an ATM network interface card. However, the connection of these components with the network side (SIGNE) is more complicated (Fig. 5.) This figure introduces the functional structure of our experimental system in the first development phase. Beside the main parts other units can be seen. The *Application Demon* (AppLD) executes the User Application after establishment of the call. The *Local View of Service Instance* (LVSI) stores the description of telecommunication service with the actual attributes set-up by participants. The *Telecommunication Service Protocol* (TSP) determines the communication between EMMA and SIGNE, since *TCSLib* is a library of several TCS Descriptions. The EMMA interacts with the user through the metaphors of the Framework, written in Java. The UA is built up of *Application Building Blocks* (ABB), i.e. well determined, basic functions for handling several media.

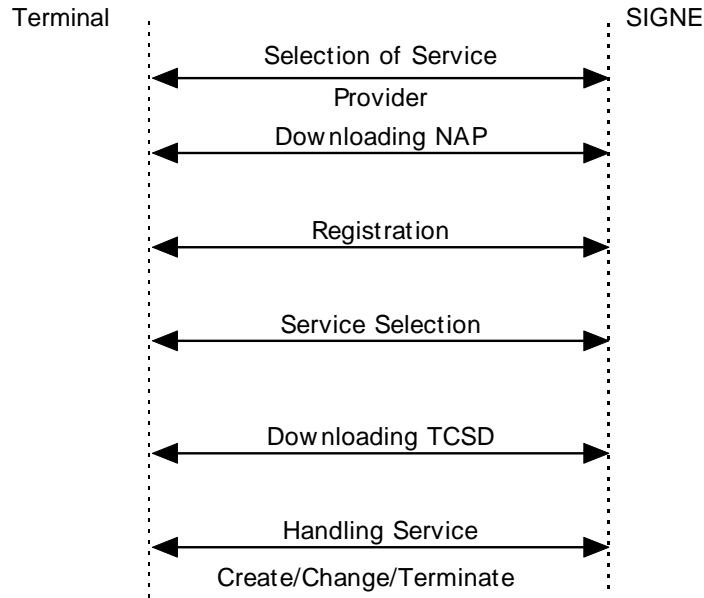


**Fig. 5.** Connections between the terminal and network side

### 3.2 High Level Signalling

One of the key functions in B-ISDN is signaling. However, *service oriented messages* are not supported yet by currently standardized signaling (capability set 1). Thus we designed an own protocol family. The *Telecommunication Service Protocol* (TSP) denotes the protocol between NET subsystem and SIGNE. It is operating over TCP/IP and consists the signaling required for call establishment. Ordering of the service and steadily control of call parameters are considered as service management in this paper. The user actions are coordinated by SIGNE.

According to our concept, there will be no differences among the parties during connection set-up. Every participant can join to the session following the same manner, i.e. after filling out the Login Panel and registration. It ensures a certain security to the system, since everybody should use a password not only the initiator of the conference. From the other hand, every user can apply the same environment independently on being invited or initiator.



**Fig. 6.** Main steps of communication between terminal and network

The major steps of connection set-up are the following (Fig. 6):

- *Login and registration* by clicking on Multimedia Services WWW Page in the Web browser.
- *The Network Access Program (NAP) is downloaded* and asks for user name and password, which will be verified and notified by SIGNE.
- *Service Selection* happens by clicking on preferred service type and Create (video conference available in recent EMMA phase) or on Wait for Call button, which options are provided by NAP.
- In case of *Create*, *TCSD model is downloaded* and interpreted by NUSI. The dialogue boxes and framework icons necessary for configuration of the call are presented by NUSI. The user specifies his/her wishes and the data structure are sent to SIGNE for evaluation, which informs the invited participants. The called parties have right to accept, reject or downgrade the initial service. Their answers are reviewed and the available service description is sent to the initiator, who decides about the entire call. SIGNE broadcasts the result to every involved parties.
- In case of *Wait for Call* mode, NAP waits for invitation from SIGNE and presents it to the user.
- NAP forwards the user's answer to SIGNE and goes back to wait state until SIGNE informs it about the decision of the initiator.

Beside the connection set-up, steps of changing service parameters, parties, terminating of the call, etc. are designed as well.

## 4 Building Process of the EMMA Videoconference

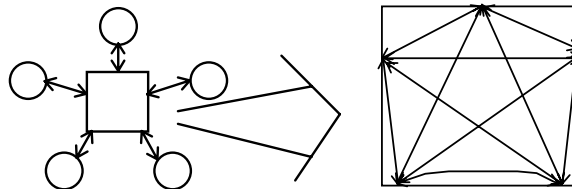
Videoconference with white board is implemented as the first application in the EMMA - SIGNE environment. The number of participants could be two but theoretically there is no upper limit. From the other hand, a practical upper limit is given by the workstations used by the participants. With the recent technology, we can say that the maximum number of participants is in the order of ten, so ten people are able to see and talk to each other. At a videoconference there is generally one initiator and the other participants are waiting for an incoming call. The other possibility is, when two or more currently existing (small) videoconferences would like to connect to each other. In this case there is one initiator, but the arrangement of the other participants is not so trivial.

What are the basic topology types of a conference? As depicted in Fig. 7. the one is the star shaped topology, where all participants are connected with one and the same - the central - machine (of course, this should be used by a participant too), other possibility is the full graph topology, where all of the participants are connected with all the others.



**Fig. 7.** Basic conference topologies

Since ATM is “faster” than the workstations, the full graph were better solution. On the other hand, the physical construction of the network is more for the star topology. Further more the star topology has significant disadvantages in the view of us. Firstly, there has to be a central machine with more orders of magnitude faster and better than the others. Secondly, the sensitivity of the system is questionable, the central machine is a bottle-neck. The solution is the so called star shaped full graph topology, what is practically a full graph (Fig. 8).



**Fig. 8.** Star shaped full graph topology

In the next part we will take a closer look on the AppLD, in the sense how are more AppLDs on more machines available to get in connection to each other, namely how does the protocol work.

### 4.1 Steps of Building up an Application

The demon is a process, what runs permanently, and watches a so called well-known port. If a person wants to make a connection to an other station, he has to send a message to his own

demon to get a connection to the other demon (the called party). In the simplest (and not fully correct) case a demon must be able to receive three kind of messages given in table 1.

Nr.	direction	parameter	short description
1.	user → demon	remote host address	a user wants to make a call.
2.	local demon → remote demon	local port number	invitation to make a connection
3.	local demon → remote demon	third party's host address	invitation to make a connection with a third demon

**Table 1.** *Message types received by a demon*

Steps of building up an application:

1. The user sends a message to his own demon (type 1.), that he would like to make conversations with an other terminal (user). The demon controls the *videophone* processes, and tells them to launch a new ABB video and audio process, whenever there is no connection to this machine yet. The socket number of the launched processes will be told the AppLD. The AppLD can now connect to the other AppLD, and tell the socket number of his ABB processes (message type 2).
2. Whenever the machine had already connections with others and the user has the authority to invite people to the conference, then the AppLD has to send to each of the partner AppLD-s a message type 3, so they know that a new machine is to connect.
3. If a demon gets a message type 2, then it launches its own ABB-s and the connection is made. If it had already connections with others, then it sends to each of the partner AppLD-s a message type 3, so they know that a new machine is to connect.
4. At a case of getting message type 3, a demon will act like in case 1.

## 5 Future Work

The first phase of the project is close to the end and the range of possible experiments is wider and wider every day. Among the future ideas, a multimedia application authoring tool and a cell rate control mechanism for the network interface are to be implemented. We also plan to adapt the system to other computer platforms, such as PC, special ATM multimedia devices (e.g. AVA 300 from Nemesys) and mobile terminals. The distribution of centralized functions is still an open issue, although there are some promising solution to this goal, like CORBA for Java.

### Acknowledgments

The reported research activities are made in the joint research program between Telia Research in Sweden and High Speed Networks Lab, Department of Telecommunication and Telematics, Technical University of Budapest in Hungary. We would like to thank our colleagues in Sweden and in Hungary for the help during the design and implementation of this system.

## References

- [1] H. Haapasalo, I. Norros, T. Raij, "Some Experiments From ATM Traffic Measurements", *The Twelfth Nordic Teletraffic Seminar*, Espoo, Finland, August 22-24, 1995.
- [2] S. Molnár, I. Cselényi, N. Björkman, C-G. Perntz, M. Boda, "ATM Traffic Measurements and Analysis on a Real Testbed" *10th International Teletraffic Congress Specialist Seminar*, Lund, September 17-19, 1996
- [3] E. Aarstad, R. Venturin, "Traffic Characterization of the JVTOS Multimedia Application", *The Twelfth Nordic Teletraffic Seminar*, Espoo, Finland, August 22-24, 1995.
- [4] S. Singh, S-S. Chan, "A Multi-Level Approach to the Transport of MPEG-Coded Video Over ATM and Some Experiments" *GLOBECOM'95*, Nov. 13-15, 1995.
- [5] J. Pugaczewski, W. Walling, M. Wiklund, "Multimedia Traffic Characterization from the High Definition Distance Learning ATM Trial", *20th Conference on Local Computer Communications*, Minneapolis, Minesota, Oct. 16-19, 1995.
- [6] I. Cselényi, N. Björkman, P. Haraszti, I. Szabó, C. Gisgard, A. Latour-Henner "Experimental Multimedia Middleware for ATM" *Hungarian-Swedish Networkshop '96 on ATM Networks*, Balatonfüred, May 28 June 1, 1996
- [7] RACE Project R2044: MAGIC- Multiservice Applications Governing Integrated Control, Service Description Framework and B-ISDN Service Descriptions, 3rd Deliverable, EC, 1992.