Service and Connection Management Architecture for Distributed Multimedia Applications

Choong Seon Hong, Hirofumi Abe, Dai Kashiwa and Yutaka Matsushita
Faculty of Science and Technology, Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223 Japan
E-mail: {cshong, hirofumi, kashiwa, on}@myo.inst.keio.ac.jp

Abstract

This paper addresses a novel networking architecture model on DPE for various multimedia services suitable to high speed networks and for the flexible and rapid introduction of services. In this model, applications are assembled from software "building blocks" which copywith information, service and network providers. Each building block provides a layered view, enabling the effective management of multimedia network resources and services according to the concept of TMN and TINA.

In this paper, we also propose the use of a directory system and its naming structure for the management of user profiles and session profiles, and a control model of effective multimedia logical device objects that uses a stream process approach. Our model is implemented on a DPE platform that provides various transparencies. For the purpose of flexible service provision to users, this architecture presents Ad hoc service building blocks such as a video on demand building block and a CSCW building block.

1 Introduction

Multimedia computing technology has greatly advanced in the last few years. So there is currently considerable interest in developing an architecture to interconnect multimedia applications. Major research in those area includes TINA (Telecommunications Information Networking Architecture)[1], INA (Information Networking Architecture)[2][3], and so on. These networking architectures provide solutions for critical business problems by enhancing the environment of the existing networks to handle multimedia communication and multimedia information, reducing delays and costs in introducing of networking services. The architecture for introducing new services is not concrete yet; we present here extensions. We especially propose the adoption of special purpose computational objects (in this paper, we call computational objects "building blocks"). These objects have open interfaces to general purpose computational objects to provide potential connectivity and to support an environment capable of flexible multimedia service provision and the rapid introduction of new multimedia services. This modeling architecture also has the objectives of promoting application modularity and portability.

In this architecture, the service and resource management functions specified by TMN (Telecommunications Management Network)[4][5] and TINA [6][7][8] are achieved by the cooperation of logically distinct building blocks (BBs). If we classify those building blocks according to their roles, group call building blocks, connection control building blocks and transport networks belong to service/network providers, and terminals including group call agent building blocks that play a role of video on demand (VOD) server, for instance, are operated by information providers. Ad hoc service building blocks belong to service providers' domain.

We implemented this architecture on DPE, which provides a uniform execution environment for distributed applications in order to take advantage of distribution transparencies defined in RM-ODP[9]. A list of transparencies that impose requirements on building block interfaces are as follows:

- Access transparency that enables interworking across heterogeneous computer architectures and programming languages. This transparency requires that building block interfaces are machine and programming and language independent.
- Failure transparency that requires that some failures are masked from the users of building block interfaces. This imposes requirements on the hardware and/or additional software mechanisms for replication, checkpointing and recovery.
- Location transparency that requires that interfaces do not reveal information about interface location. This implies that building block interface references contain location independent identifiers that enable the DPE to locate an interface instance.
- Relocation transparency that requires the propagation of information about changes in object location. This requires that the DPE can trace the location of a building

0-8186-7216-1/95 $04.00 © 1995 IEEE

block interface reference, in case it has changed location.

In the remainder of this paper, section 2 describes an ODP approach to the design of a networking architecture model. Section 3 then addresses the overall networking architecture that we propose. Following this, Section 4 and 5 explain how to manage the user and session profiles with the naming structure that we propose, and discuss signal interaction between building blocks, respectively. Section 6 describes a scenario for introducing a new service. A multimedia terminal architecture is discussed in section 7, and realization and evaluation are contained in section 8.

2. Networking specifications using ODP approach

For the purpose of addressing multimedia service networking specifications, we used the basic concepts of Reference Model for Open Distributed Processing (RM-ODP) and extended some definitions to adopt those in a novel networking architecture. Specifications are described from enterprise, information, computational and engineering viewpoints. OMT graphical presentations[16] for each viewpoint are expressed for only the enterprise and the information specifications. The others are described in detail in several sections following this section.

Enterprise specification: The enterprise specification describes the objectives, policies and requirements of multimedia services. The requirements and policies of multimedia services are obtained from the parties involved.

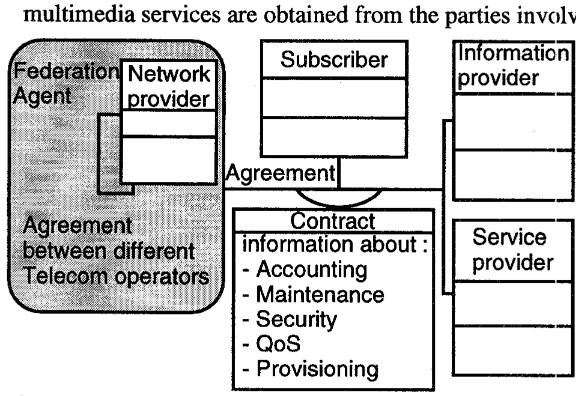


Figure 1: Enterprise representation for multimedia networking service system

They can be classified according to their roles:

• User: a person or machine who uses services in order to satisfy communication needs.

• Subscriber: a person or organization that contracts services offered by service providers or network providers.

• Network provider: an organization that commercially manages networking services and provides the service execution platform offered to subscribers according to contractual agreement.

• Service provider: an organization that commercially provides service packages and manages their services for subscribers by means of network providers.

• Information provider: an organization that provides information to be managed by means of network providers and to be used by users.

In this paper, user and subscriber are described as "user". An OMT graphical representation of the enterprise viewpoint for multimedia service networking is shown in Fig. 1

Information specification: Information specification of the multimedia networking services describes semantics and requirements for processing service information. The specific information is visible in terms of the following:

• The information object classes involved in application.

• The information activities that constitute the application.

• The constraints on the changes of state that can take place in information objects.

The information viewpoint for multimedia networking services represented by invariant schema using OMT graphical representation is shown in Fig. 2.

Computational specification: Computational specification describes interactions between open distributed systems' components that are remote from each other. The details of the infrastructure which supports these interactions are not visible in the computational viewpoint. They are described in the engineering viewpoint. In this paper, this specification is discussed through the networking architecture overview and interactions between building blocks.

Engineering specification: The engineering description is related to the infrastructure required to support selective distribution transparent interaction between objects. This engineering specification exhibits the behavior described in the computational specification. The engineering rules contains the following:

• The structure of communication channels.

• The structure of systems for resource management.

The above concepts are to enable specification of internal interfaces within the infrastructure, enabling the definition of distinct conformance points for different transparencies, and the possibility of standardization of a generic infrastructure into which standardised transparency modules can be placed. In our architecture, the realization of a specification from the engineering viewpoint is achieved by using an ATM transport network, an ANSAware platform, Parallax XVideo, and so forth.

3. Networking architecture overview

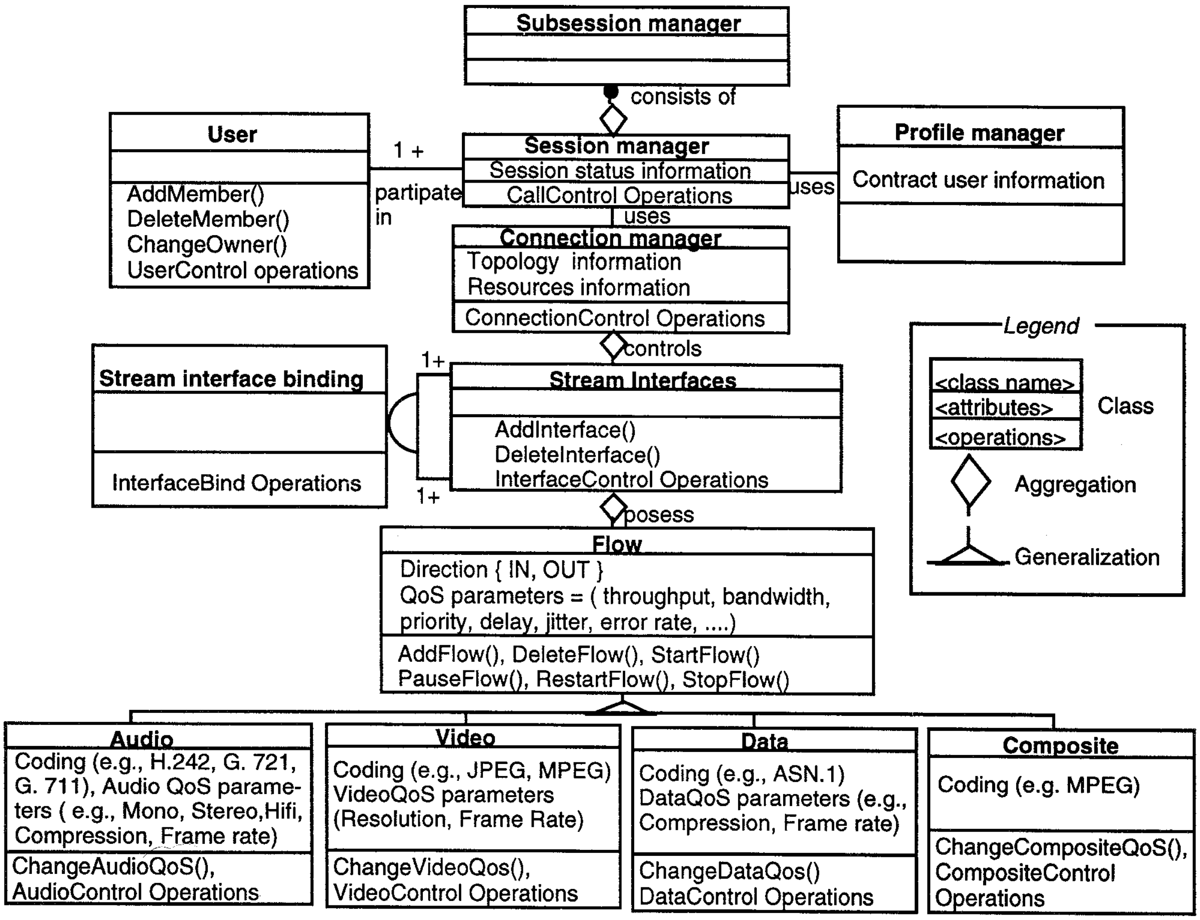


Figure 2: Invariant schema for multimedia networking service systems

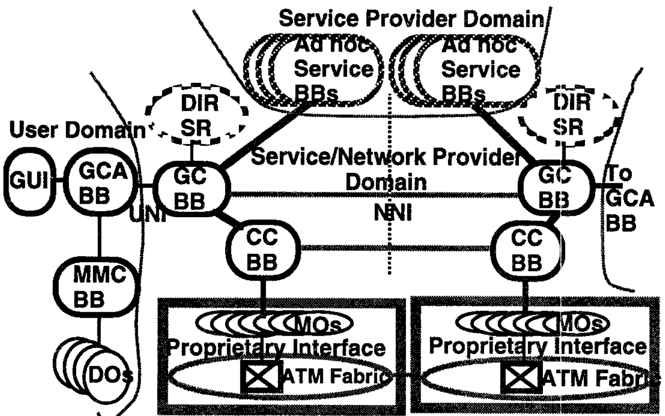
The networking architecture for multimedia service we propose is composed of several functional building blocks on a DPE in order to support flexible service provision based on a high speed network. The simple view of our system is depicted in Fig. 1, describing the interaction among network components (building blocks, servers, multimedia network elements, etc). A group call building block plays the role of service activation. It links the users of the service so that they can interact with each other and share service entities (e. g., CSCW BB). This provides operations that allow users to join or leave a service session. For certain services it may also offer operations to suspend and resume involvement in a service. A connection control building block is a service-oriented abstraction of connections in the transport network. This building block maintains the state of the connections of a particular service session, such as communication paths, end-points and a quality of service characteristic. A connection control building block is only required when streams between computational objects are required.

The purpose of these concepts is to separate out different concerns and to promote distribution of functionality. The separation of service and connection sessions support the division of activities of the service from the set of connections that exist. There are two essential reasons for this split. Firstly, not all services will require the use of the transport network. In these cases, services will be provided by the DPE. Secondly, even if a service establishes connections on the transport network, other activities may be taking place, and the users may be involved; there is not necessarily a one-to-one correspondence between those that are involved in a service and those that have transport connections as part of the service.

In this section, we describe in detail each of building blocks that are constructed according to concepts discussed above.

3.1 Group call agent building block and group call building block

Group call agent building block (GCA BB) provides



GCA BB:Group Call Agent Building Block MMC BB:Multimedia Control BB MOs:Managed Objects DOs:Device Objects GC BB: Group Call BB CC BB:Connection Control BB DIR SR:Directory Server UNI:User Netwok Interface NNI:Nework Node Interface

Figure 3: Configuration of networking architecture proposed

the customized view to a user and passes the GCA BB's messages generated by means of user requests to the group call building block (GC BB). A set of messages can be transferred to the GC BB. Some message types for multimedia conferencing are : Call_request, Add_member, Join_member, Add_service, Delete_service, Delete_ member, Close_call and Drop_member.

GC BB receives and analyses user requirements (adding or deleting an endpoint, and requesting retrieval, etc.) received through the GCA BB, manages the user profiles, and represents states of the present call (or session profile). Call states include the participants of the current session, the session name, call ownership, call initiating time, QoS level for each medium and the bandwidth of the user's contract. The results of analysis based on the messages invoked by the originating endpoint are passed to the connection control building block (CC BB) for establishing a communication connection. If the terminating endpoints are not in the network managed by the local GC BB, the originating GC BB sends the user's requirements to the terminating or intermediate GC BBs corresponding to other endpoints. If a GC BB receives a user's requirement for a special purpose service, it passes the user's message to an Ad hoc service BB. A GC BB corresponds to a user agent, subscription manager and service session manager for a normal call (e.g., phone call, video conferencing call) in the TINA context[10].

3.2 Connection control building block

This functional building block establishes, modifies and releases the connections requested through a GC BB. A CC BB enables the transfer of the logical address to the physical address. Also, this BB manages the information about the network's topology. Using the topology information, CC BB provides the optimal routing as well. For the purpose of effective resource control, this BB adopts an object oriented approach. In the TINA context, this building block plays the role of communication session manager, connection coordinator and connection performer.

Multimedia resource control model. : In order to help flexibly generate new objects and effectively manage the connection, and effectively meet to users' requirements, we present a multimedia resource control model. This scheme prepares several controlling objects so that the managed objects (MOs) can handle real resources in the ATM fabric. The managed objects that control network elements are indirectly controlled through management proxy for these objects. Those objects are described by using a logical connection graph. The objects we defined for multimedia connection are multicaster, mixer, link, *QoS* converter, endpoint, and so forth.

As an example, if a CC BB is requested to establish a three-party audio connection from GC BB, the CC BB first creates the objects, then connects endpoints and the mixers/multicasters. Finally, the mixers and multicasters are connected to each other.

Accordingly, using this scheme, the request of multipoint-to-multipoint connections for each service type (e.g. audio, video, message, delivery service) from a GC BB can be flexibly constructed by controlling the logical objects. Especially, if it is needed to supply the different *QoS* copying with the request of a user and the terminal capability, this scheme can use the *QoS* converter.

3.3 Ad hoc service building block

Ad hoc service building blocks are used to provide the special purpose services such as video on demand (VOD), home shopping, CSCW to users. These building blocks may be provided by service providers. Therefore, the network provider has to prepare a standard interface between the GC BB and Ad hoc service BBs to promote the development of special purpose services. Ad hoc service BBs can be categorized into two classes: the dedicated class which is maintained during the entire session life time (e.g., for CSCW) and the semi-dedicated class which is just maintained while selecting a service (e.g., for VOD, home shopping). Each Ad hoc service BB manages the special purpose service subscriber's information, the contents of service and its domain to support the user's selection service, the cooperative work domain, and so forth.

4. The management of user and session

4.1 Directory system

Primarily, the X.500 directory[11] provides a mapping facility from user-oriented name to machine-oriented name which preserves the information of the objects in the real world. The user-oriented name can be mapped to several characteristic objects which have meaning in control and management in the virtual and real worlds. Namely, given the name for an object, we can easily retrieve and modify the attributes of the object, and create or delete subordinate objects and their attributes.

In this paper, we present an example that elaborates our proposed networking architecture. The detailed description for naming objects will be the next sub-section. The Distributed directory system (DDS) is composed of Directory User Agents (DUAs) and Directory System Agents (DSAs), and includes Directory Information Base(DIB).

• DUA : Each user is represented in accessing the DUA, which is considered to be an application process. DUAs may also provide a range of local facilities to assist users' queries and interpret the responses.

• DSA : This is an OSI application process belonging to the directory. It provides access to the DIB (described below) for DUAs and/or other DSAs. The directory can be distributed over an arbitrary number of DSAs. Each DSA is responsible for a subset of entries in the DIT (Directory Information Tree). This fragment of the DIT is called the naming context of that DSA. A DSA may use information stored in its local database or interact with other DSAs to carry out requests.

• DIB : The contents of a DIB describe both *static* and *dynamic* aspects from the perspective of network management and application work.

In Fig. 3, the directory servers (DIR SRs) are interconnected in the intracarrier and intercarrier domain to allow access by a DUA to information that is being managed by remote DSAs.

4.2 Naming structure of entries

In our proposal, a DDS is used for constructing the user and session profiles used in a phone call, a conferencing call and the other services. As a client for a DSA, the DUA requests necessary information to a DSA. The DSA holds static and dynamic information. The static information those involves objects whose contents are rarely changed during an operation. Examples of such information are objects related to some addresses and objects' attributes (e.g., user name).

The information which is related to an address is called the "user profile". This profile is created when a user subscribes to use a network. On the other hand, dynamic information is the objects that which changes very frequently. Examples are call identification and state in of a network resource. This information are called a "sessionprofile". This profile is created during a session and deleted with the close of the session. We constructed

the DIT in order to make the two profiles that are described above. The DIT structure is shown in Fig. 4. ADD_CL and User_address entries are static information

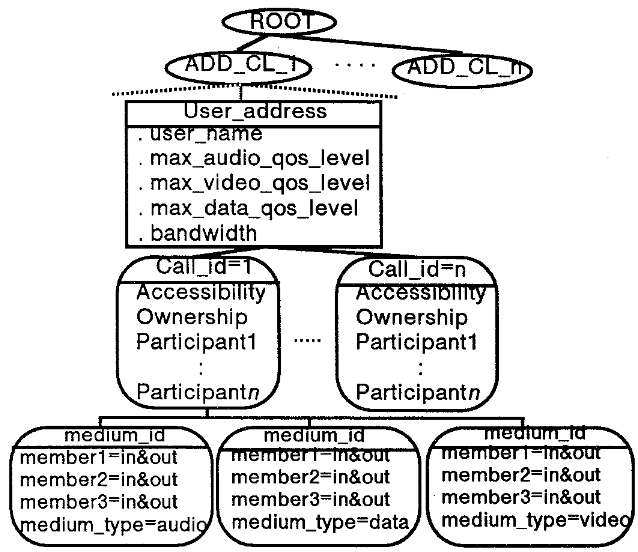


Figure 4: Naming structure of directory service system

not to be changed during a session operation. Those entries are components that construct a user profile and are composed according to the ITU-T numbering plan (E.164). The Call_id, a subordinate entry of User_id and the Medium_id entry compose a session profile. In the figure, a user has multiple call identities which have only meaning in one user address.

Each call identity(Call_id) is composed of several media (audio, video, data, etc) and a medium has multiple users' information. The definition of information to be managed by media identities is summarized as follows :

$$\text{Medium\_type1} = \text{Member1}^{\text{in\_QoS}}_{\text{out\_QoS}} + \cdots + \text{Membern}^{\text{in\_QoS}}_{\text{out\_QoS}}$$

$$\vdots$$

$$\text{Medium\_typen} = \text{Member1}^{\text{in\_QoS}}_{\text{out\_QoS}} + \cdots + \text{Membern}^{\text{in\_QoS}}_{\text{out\_QoS}}$$

in_QoS and out_QoS mean a user's incoming *QoS* level and outgoing *QoS* level, respectively. In case of delivery on demand service, the delivery server's in_QoS value would be 'none'. The *n* in Medium_type*n* and Member*n* are determined by a user's terminal capability and the network provider's policy, respectively. So, the above definition can deal with both symmetric and asymmetric characteristics. The CC BB uses this information for establishing the connections.

5. Signal interactions between building blocks

In this section, we discuss the interaction of signal messages. The signaling protocol which is adopted in our architecture is shown in Fig. 5. This signaling procedure

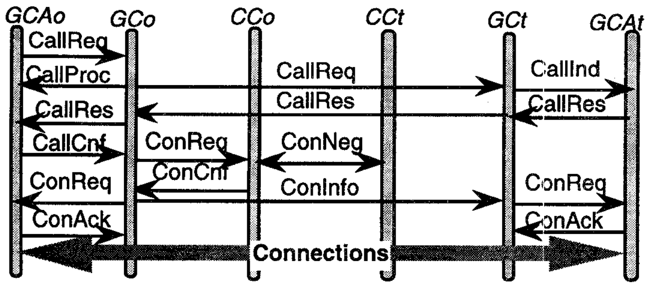


Figure 5: Signalling flows between building blocks

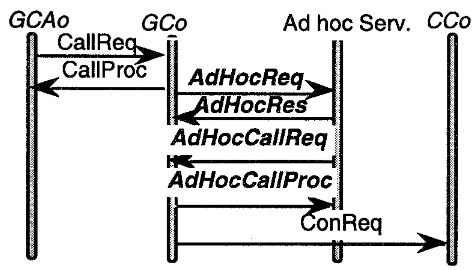


Figure 6: Signals added for Ad hoc services

doesn't include that for using Ad hoc service BBs. The signaling flow related to Ad hoc service BBs is described in Fig. 6. In this figure, the signal messages designated by ***bold italic*** type are added for Ad hoc services. The GC BB differentiates whether a required call is a phone call, normal conferencing call or a call for Ad hoc services. In those two figures, *o* and *t* stand for the originating and the terminating building block, respectively.

In our architecture, *QoS* negotiation is carried out in 2 phases. The first phase is achieved by negotiation between GC BBs based on users' contracts and terminal capabilities, the second one is performed between CC BBs for the purpose of checking the availability of network resources.

6. Scenario for introducing of new services

We introduce an Ad hoc building block which plays a role as Video on Demand (VOD) service agent . This building Block is called VOD BB in remainder of this paper. As an agent, VOD BB has the functionality to provide to users for whom it has the subscription profile, content available from the information providers (or agents), the agents' characteristic information, and so on. The VOD BB is simply added by adopting interfaces defined between a GC BB and an Ad hoc service BB described in section 3. The control and management of the video delivery server is not discussed in this paper. Using the proposed architecture, a VOD BB is introduced as follows:

We assume that the local GC BB can directly access the VOD BB without passing to another GC BB and each BB uses a trader for importing the reference interface that is needed to access another BB.

❶ A user agent (GCA BB) requests to the network (GC BB) to receive VOD agent services. ❷ The GC BB sends a message to a VOD BB that provides the decision-making service for VOD. ❸ The VOD BB sends a message to the GC BB for setting up connections that support the interactive works between the user and the VOD BB, and the the GC BB sends a message to the CC BB to establish the connections (❸') for supporting the user's interactive service. If a user selects a service based on the information provided by an agent in the VOD BB, the related parameters for connections (video server address, service name, QoS for selected service, etc) are sent to the GC BB. ❹ The

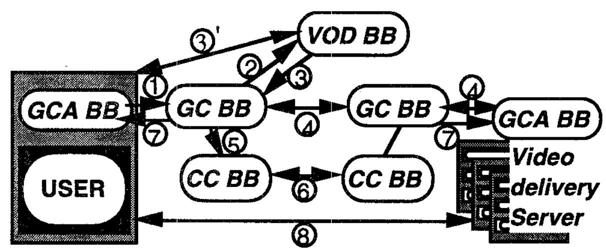


Figure 7: Scenario for receiving a video delivery service

local GC BB performs QoS negotiations with a remote GC BB that manages a video delivery server and with the video delivery server through the remote GC BB. ❺ If the response from remote GC BB is that video delivery server is available, the GC BB requests the establishment of a connection to the CC BB. ❻ The CC BB that received the connection request from the GC BB checks whether the resources chosen through VOD BB by the user and the result of their negotiations in ❺ are available or not. If the resources are available, the CC BB negotiates with the other CC BB in order to establish connections. ❼ If network elements' resources as a result of negotiation between CC BBs are available, each GC BB sends a connection request message for creating terminal resources and connecting to networks, respectively. ❽ After the complete of connections, a service from the video delivery server is provided to a user.

The configuration of the above scenario is illustrated in Fig 7. For the second example of an Ad hoc service BB, adopted for the CSCW BB. We prototyped the CSCW BB using the proposed networking architecture.

7. Multimedia terminal architecture

The multimedia terminal mechanism for managing multimedia resources and users' requirements uniformly treats the various types of multimedia resources as logical devices and a convenient set of logical device manipulation primitives. To realize this mechanism, we adopted GCA BB and MMC BB in the multimedia terminal. We described GCA BB in the section 3.1. In this section, we discuss MMC BB.

7.1 Multimedia control building block

MMC BB is a functional building block that manages the resources for a terminal. This BB receives control messages from a GCA BB, prepares necessary device objects, and manipulates those objects. There are several classes of device objects.

Connection model of MMC BB and device objects (DOs) : an MMC BB is composed of 2 levels. The lower level handles data between DOs. The higher level manages all DOs. A DO sends the data to the lower level of the MMC BB as an Up Call, then the DO receives the data from the MMC BB as a Down Call. By this principle the MMC BB directly controls the data flow, and the problem of synchronization between media can be solved. The management of the stream that is described in the last part of this section is performed in the higher level of the MMC BB.

Device object : An MMC BB creates, connects and controls DOs. DOs are classified according to the kinds of media involved and their functions. Logical devices such as windows and files also can be handled as DOs. Using these principles, the MMC BB can manage the terminal resources uniformly. DO classification is as follows:

- Classification by means of input/output function
  - Source class : DO class generating information
  - Sink class : DO class consuming the information
- Classification by kind of media
  - Audio class, Video class and Data class (not continuous data)

The concept of stream process : An MMC BB manages a set of DOs that are immediately related as a "stream process". The control messages that are sent by a GCA BB mostly are handled as a stream process. There are some reasons why we adopted the idea of stream process:

- Generally, a set of DOs related to the same data stream are operated in the synchronization state. For such kinds of DOs, the operation for controlling DOs normally does not need to be performed on an individual DO. So, after DOs are connected, operations on stream processes can be performed effectively.
- The set of DOs that belong to the same data stream is required to be scheduled by a predefined procedure; for example, from source to sink for simplicity of operation.

In order to achieve this, we introduced the concept of stream process. Operation commands on stream process include add, close, start, stop and resume. The group of DOs that must be synchronized is defined as a stream process. A programming example consisting of a stream process for effective management of DOs is shown in Table1. To enhance execution performance of MMC BB, mixer and multicaster, we used the thread[12].

Table 1: Example C++ listing for stream process

```
// Creation of device object
DefaultMic   mic(100);
DefaultSpeaker  spk(100);
// Connection of  device object
mic.Connect( &spk );
// Creation of  stream process
Stream stream(15);
// Registration device objects  to stream process
stream.AddDevice( &mic );
stream.AddDevice( &spk );
// Request of resource reservation to all device
objects
        // registrated to stream process
if ( stream.Open() == mmcFalse ) {
 cerr << "Open Failed\n";
 exit(1);
}
// Resource reservation and wating for
stream.BlockTillOpen();
// Start scheduling to Devices
```

8. Realisation and evaluation

The architecture we proposed in this paper is implemented on an ANSAware environment[13]. The service components were registered in a trader with service types and interface references. We now explain how to pass a message between building blocks. Refer to Fig. 8. ❶A GC BB and a CC BB export their own interfaces to the trader. ❷A GCA BB imports from the trader an interface reference (InRef) to request service to the GC BB using the InRef. ❸A GCA BB sends a message to a service access point to the GC BB using InRef that imported from trader. ❹If a GC BB needs to establish a connection using the operation message of ConReq, it imports an InRef using the service type CC. ❺Then, the GC BB accesses

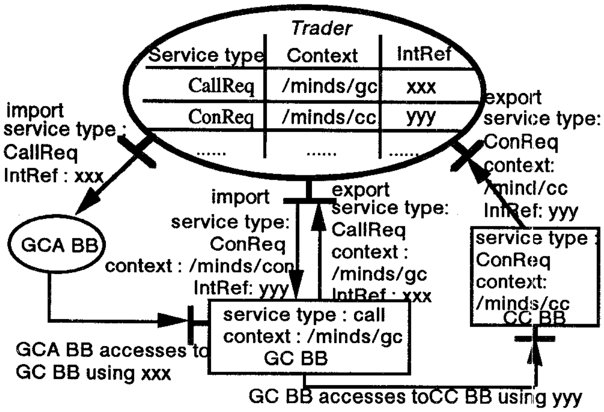


Figure 8: Procedure using trader to establish connection

the CC BB using the InRef.

The directory system used for managing user/session profiles by the GC BB made use of ISODE/QUIPU[14]. Especially, in order to incorporate DUA function into our system, we implemented DUA functionality using a procedure call provided by ISODE/QUIPU. A session profile in the DIT (described in section 4.2) can be flexibly

Table 2: Result for measurement operation on entries

Functionality	Average Access Time
read_entry	64.233 ms
add_entry	414.638 ms
modify_entry	423.368 ms
delete_entry	399.121 ms

changed during operation. In Table 2, we show the results of time measurement taken to read and modify entry information and to add or delete an entry. This measurement was performed on a SUN SPARCstation. The read_entry function affects a user's service time. On the other hand, the other functions are not related to a user's service time. To enhance the these processing times, we could use the LDAP(Lightweight Directory Access Protocol)[15] proposed by the ISODE consortium.

Because the *ForeRunner*™ ATM Fabric possessed by our lab. doesn't provide a media mixing function, we used a TCP/IP transport network for flexible control of the audio mixing function and the media multicasting function that we implemented by software.

To prove the usefulness of Ad hoc BB, we constructed CSCW BB. The configuration describing the role to be played by the CSCW BB is shown in Fig. 9. In this figure, the CSCW BB broadcasts to all participants (i.e., user , agent, carriers) the data sent by the user, agent or carriers.

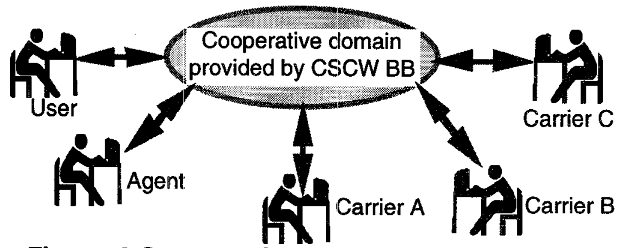


Figure: 9 Concept of domain provided by CSCW for cooperative work

So, the CSCW BB provides a cooperative work domain for the user, agent and carriers. The details of our CSCW application are beyond the scope of this paper. The demonstration views for the video conferencing with the special purpose network management that are realised using CSCW BB as Ad hoc service BB are shown in Fig. 10.

The demonstration view illustrates group work for the effective management of a permanent virtual path (VP) as a leased line. In these figures, an agent manages the network information and deals with users' requirements that modify the bandwidth and path of the VP connection. It also has a negotiating function with carriers as the network provider.

9. Conclusions

This paper proposes a networking architecture for the effective multimedia service and connection management and describes its implementation on a DPE. This architecture is supposed to provide flexible multimedia services in high speed network environments by using Ad hoc service building blocks.

The directory system making use of X.500 was constructed for the easy and flexible management of service information such as *QoS* level, bandwidth and the subject and owner of a session, and for network resource management, in an open system environment.

The architecture realised by a multimedia control building block in a multimedia terminal hides the complexity of physical elements and provides flexible connection to the each device object. For the purpose of proving the usefulness of proposed architecture, we realized a CSCW BB as one of the Ad hoc building blocks playing a role of service provider.

Acknowledgements

The authors would like to thank Dr. Yuji Inoue, Dr. Kenichi Kitami, Mr Osamu Miyagishi, Mr Fumito Sato. Mr Sakae Chikara and Mr. Wataru Takita at NTT

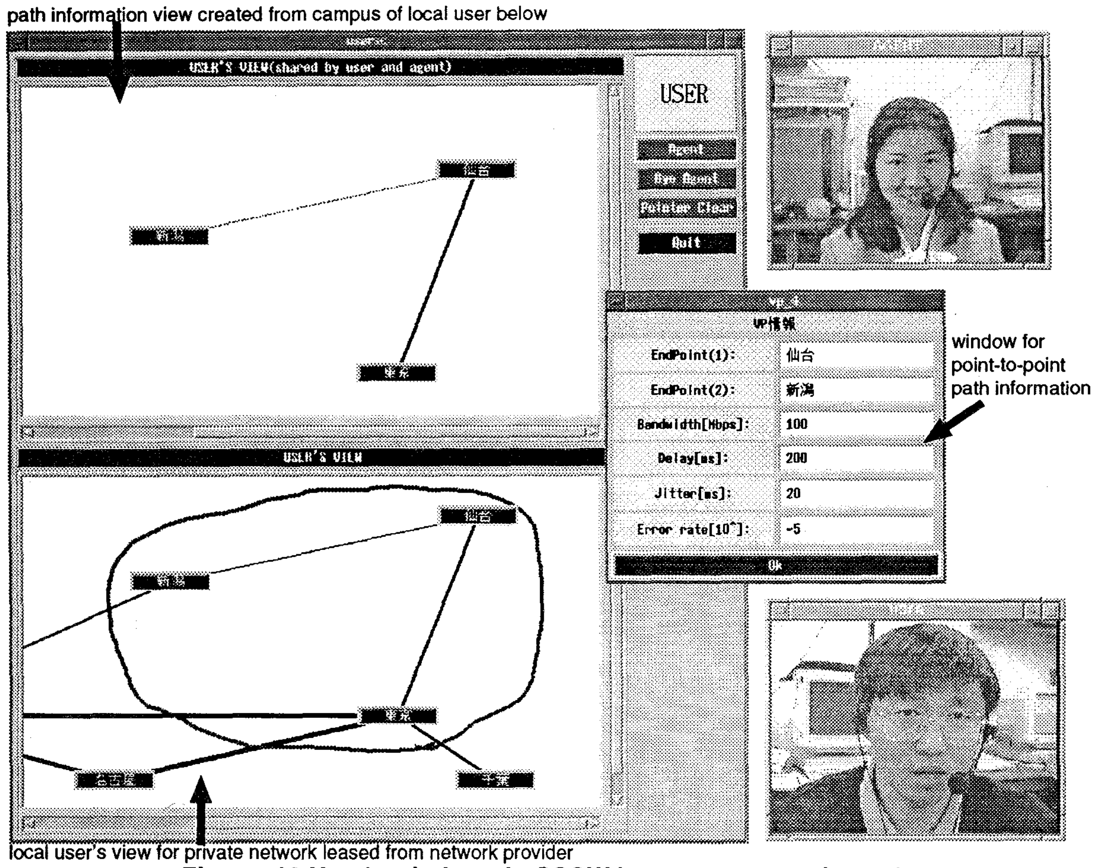


Figure: 10 User's windows in CSCW between user and agent

Telecommunications Network Laboratories, Professor Jonathan Grudin U. of California at Irvine, for their useful discussions and comments. The authors particulary want to thank out collegues who implemented our system : Mr. Shinkuro Honda, Mr. Kiyoto Kawauchi.

References

[1] Barr, W. J., Boyd, T. and Inoue, Y., "The TINA Initiative, IEEE Communications Magazine.", March 1993, pp. 70-81.

[2] Natarajan, N. and Ferro, M., "The Distributed Processing Framework in the INA Architecture", Proc. TINA '93 Vol. 2, Sept. 1993, pp. 255-267.

[3] Natarajan, N. and Slawsky, G. , "A Framework Architecture for Information Networks",IEEE Communications Magazine , April 1992, pp. 102-109.

[4] ITU-T, "Principles for a Telecommunications Network Management", ITU-T Draft Rec. M. 3010 , 1991.

[5] ITU-T," Generic Network Information Model", ITU-T Draft Rec. M.3100 , 1992.

[6] Appeldorn,M., Kung R., and Saracco, R., "TMN+IN=TINA", IEEE Communications Magazine, March 1993), pp. 78-85.

[7] Rubin, H., Caram, B., Flinchbaugh, G., Fratini, S., Hall, M., "The Information Networking Architecture(INA) Advantage", Proc. of TINA 93 , Vol. 1, Sept. 1993), pp. 3-21.

[8] TINA-C," Service Component Specification", TINA Draft Doc. , Nov. 1994.

[9] ITU-T, "Basic Reference Model of Open Distributed Processing - Part 3: Prescriptive Model", ITU-T Rec. X. 903 ,1994, pp. 63-68.

[10] TINA-C, "Connections Management Architecture", TINA Doc. , Dec. 1993.

[11] ITU-T, "Data Communication Networks Directory", Rec. X.500-X.521, Nov. 1988.

[12] Abe, K., "Portable Thread Library", Ver. beta1.3, http://sunfish.ics.es.osaka-u.ac.jp/PTL/ , 1993.

[13] APM Ltd, "ANSAware4.1 Application Programming in ANSAware", Document RM.102.02, Feb. 1993.

[14] Robbins, C. J., Kille, S. E., "The ISO Develope Environment: User's Manual Vol. 5 QUIPU", ISO Consortium , July 1991.

[15] Yeong, W., Howes, T. and Kille, S., "X.500 Lightweight Directory Access Protocol", RFC1487, ISODE Consortium ,July 1993.

[16] Rumbaugh, J. , at el, "Object-Oriented Modeling and Design", Printice Hall, 1991