An End System Software Architecture to Support Multimedia Multiuser Applications

T. Kwon, S. Park, Y. Choi
Department of Computer Engineering
Seoul National University
Seoul, Korea

S. Kang, H. Kim, S. Kwon
Software R&D Center
Hyundai Electronics Industries
Seoul, Korea

Abstract

The multimedia multiuser applications such as computer supported cooperative working and desktop multimedia conferencing are composed of multiple individually-controlled service elements which require complex communication services from underlying communication systems. In this paper, we present an end system software architecture to support multimedia multiuser applications in high-speed networks. The proposed end system architecture consists of three layers; the MMCP(Multimedia Multiuser Communication Platform) layer, the CASE(Common Application Service Elements) layer, and the target application layer. The MMCP provides structured and dynamically-configured communication services to the individual applications of CASE, and the CASE layer supports various types of basic application service elements which can be flexibly configured and integrated for the development of target multimedia multiuser applications. The proposed end system architecure will help to develop various and general multimedia multiuser applications flexibly.

1 Introduction

With rapid advances in the technology of multimedia workstations and high-speed networks, various types of multimedia multiuser applications in distributed environment are emerging widely and rapidly. This application area includes cooperative working, remote education, tele-medicine, entertainment, and so on[2, 4]. In general, the new multimedia multiuser applications are composed of multiple individually-controlled application service elements which require more complex communication services from the underlying communication systems than the conventional network applications. The general application service elements of the multimedia multiuser applications include voice conference, remote display of live video, shared manipulation of remote multimedia data, telepointing/marking, file transfer, message handling, and so on. The application service elements may be differently configured and integrated for the development of an application.

Most of the existing works[3, 8] for the development of multimedia multiuser applications are application-oriented, in the sense that each application includes the management functionalties of underlying data transport objects. And the service elements of the application are statically-configured to satisfy the requirements of the individual application. With this application-oriented approach, therefore, every application must handle independently the complexities of multimedia multiuser communications, and it is very difficult to re-use the constituent application service elements for the development of other applications.

In the alternative approach, platform-oriented, the services commonly used by the applications are provided by the underlying platform via well-defined application programming interfaces. With this approach, the management overhead of each application can be reduced, and the reusability of the component application service elements is enhanced. This approach has been mainly applied to develop the multimedia multiuser communication platforms which provide complex communication services in a flexible way[1, 5, 6]. The Touring Machine System[1] is a software platform developed at Bellcore to provide an infrastructure in which applications requiring complex multimedia communications can be developed, independent of the actual network fabric used to provide transport. Heijenk and et. al.[5] enhanced the call model of the Touring Machine System to support the relationships such as multimedia synchronization and audio- activated-video among the related data transport objects. Intensive research on the development of multimedia multiuser transport protocols[6] is also aimed to support common communication services for multimedia multiuser applications.

Our works on the design of an end system software architecture to support multimedia multiuser applications are devoted to the CASE(Common Application Service Elements) layer as well as the MMCP(Multimedia Multiuser Communication Platform) layer. The MMCP provides more enhanced communication services for the multimedia multiuser applications. For the MMCP, we have developed a new multimedia multiuser call model which provides more powerful structuring capability than the existing call models. Since the MMCP provides rich set of communication control services defined based on the call model, the individual application's burden to manage complex communication services can be greatly reduced. The CASE layer defines multiple common application service elements over the MMCP, which

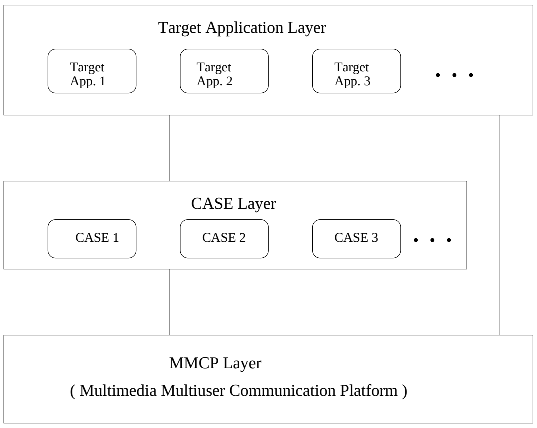

Figure 1: A Functional Model of Multimedia Multiuser End System

may be commonly used by the multimedia multiuser applications. In the proposed end system architecture, a target application can be developed by integrating the predefined common application service elements and additional application-specific service elements. This approach will greatly increase the reusability in the development of the multimedia multiuser applications.

In this paper, we review the functional model of the proposed multimedia multiuser end system . And then, we present the MMCP in detail. The proposed call model and the control services of the MMCP are presented. Each common service element of the CASE layer is briefly presented in the following section. Finally, we explain a prototype based on the proposed architcture; Multimedia Presentation System[7].

2 A Functional Model of Multimedia Multiuser End System

The functional model of the proposed end system supporting multimedia multiuser applications is shown in Figure 1. The MMCP layer provides various types of multi- connection multipoint communication services, independent of the actual communication resources. In the MMCP layer, a number of typed data transport objects, each of which provides a well-defined communication service, are defined. And, the MMCP supports control services to instantiate and structure the typed data transport objects. All data transport objects for an application are grouped as a call, top-level control unit, and further structuring within a call is allowed. Some data transport objects can be grouped as an integrated control unit. And the grouping can be nested within a call. Among the related data transport objects, some relationships can be also specified. In general, the multimedia multiuser applications are composed of multiple application service elements, and some of them can be commonly used for multiple applications with little adjustment. The CASE layer defines a number of common application service elements which may be commonly used by the target applications. Currently they include the Multi-Service Manager(MSM), the Audio-Video Conference(AVC), the Shared Working Space(SWS), and the Remote Processing of Stored Video(RPSV). Additional common service elements can be defined later. The MSM service element provides the functionalities to manage the communication services of the MMCP and other common application service elements according to the requirements of a target application. The AVC service element provides audio-video conference services among multiple users. Since the audio and video can be independently controlled within the service element, they can be individually integrated into a target application. The SWS service element allows multiple remote users to view and manipulate common data synchronously. The RPSV service element enables multiple users to retrieve, manipulate, and share the remote stored video.

Each common application service element allows flexible configuration so that it can be applicable to various applications. For example, the voice conference service element of the AVC is defined to support various dialogue modes such as free-talking with mixing, floor-controlled speaking, lecturing, and so on. It can be used for a target application with an appropriate dialogue mode. The target application layer supports end-user applications. Each end-user application can be developed by integrating the common application service elements and some application-specific service elements.

3 Call Model of Multimedia Multiuser Communication Platform

In the context of the communication platform, the call model defines data transport objects required for the communication platform, and how the underlying data transport objects can be structured to support complex communication services for the applications. And it defines how the structured objects, named as calls in this paper, can be managed in a consistent manner by the applications. In this section, we present a call model for multimedia multiuser applications.

Figure 2 shows an example of the general call structure of the proposed call model. In the call model, basic data transport objects, called Transport Connection(TC)s, can be grouped as an integrated control unit, called Grouped TC(GTC). A GTC can be a member of another GTC. That is, a GTC groups one or more GTCs and/or TCs so as to allow applications to be able to control the GTC integrally as well as to control the constituent TCs and/or GTCs individually.

The relation objects which specify the relationships among the related GTCs and/or TCs can be created as members of the GTC. The existence of a GTC and a relation is dependent on the existence of their subordinate objects or related objects. The GTCs should have one or more subordinate objects to exist. And the relation objects should have two or more related objects to exist.

This nested grouping of basic data transport objects provides more powerful structuring capability than the existing finite-level call models. In our model,

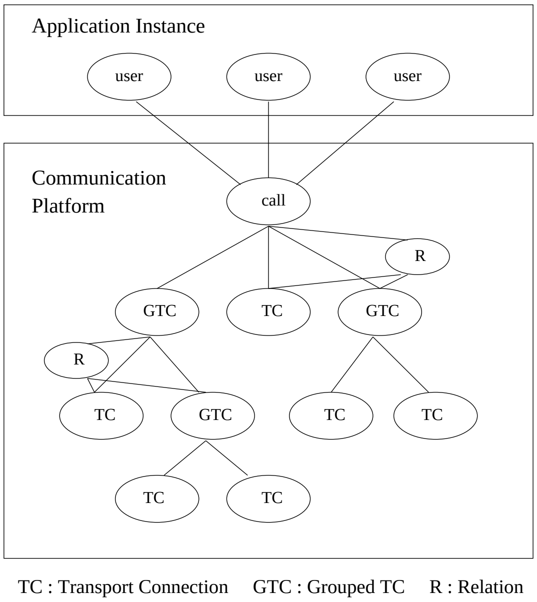

Figure 2: Call Structure of The Proposed Call Model

a call is a special GTC which represents the dynamic association between the communicating users and the communication platform, and it provides the communication capability for the application. It is the root of a grouping hierarchy of TCs, and it is an abstraction of the communication platform instance. The global view of a call can be changed during lifetime of the call. The subordinate TCs and/or GTCs can be dynamically added and/or deleted by the permitted users. And, the communicating users of the call can be also dynamically changed(i.e., users can join and leave the call).

The communicating users of a call don't have to access all subordinate TCs and/or GTCs but at least one TC or GTC at one moment. This means that a communicating user's local view of the call may be different from the global view, and local views of the communicating users differ from one another. A local view must be a subset of the global view of the call. A user's access to a GTC implies that he accesses at least one subordinate TC of the GTC. A user's local view of a call can be changed during the call lifetime. This means that a user can join and/or leave the TCs and/or GTCs of the call dynamically without changing his membership status of the call. The existence of a TC is dependent on the existence of the attached users. A TC exists only when more than two users are attached to the TC.

In our call model, the TC is the basic control unit which can not be divided into more fine-level control units. One TC is allocated for one basic service element of an application. Actual data interactions among the communicating users occur through the TCs. A communicating user can access a TC in READ, WRITE, or READ-WRITE access mode. However, a user's access QoS for the TC can be different from others. Therefore, a user's access right for a TC is represented by the combination of the access mode and access QoS. A user's access right for a TC can be dynamically changed during a call.

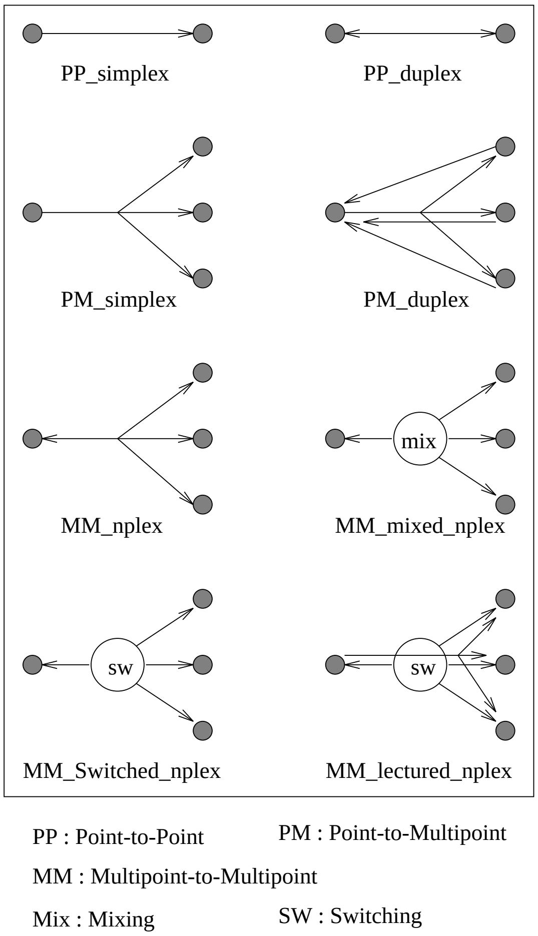

Figure 3: Example Definition of TC Types

The TC is a typed object. Rich set of typed TCs can be defined. Figure 3 shows an example definition of TC types. The MM_mixed_nplex TC mixes the data(for example, voice) of all communicating users. The MM_switched_nplex TC can be used for the voice conferencing service element where only one user may speak at any moment. The selection of speaker will be handled by the floor control mechanism. The MM_lectured_nplex TC allows the lecturer to access the TC always, but the other users' floor is switched. Explanation of other TC types is omitted.

Other TC types can be additionally defined. The call model does not constrain anything in the definition of TC types, except for the access right representation mechanism. Since the access right mechanism based on the specification of the access mode and access QoS is, however, sufficiently flexible to support various data interactions, we believe that it will not become an actual constraint in the definition of TC types. From the TC types, applications can select

```
call ::= ( call_id, call_type, { TC } | ( { TC }, { GTC } ),
        [ { relation } ], ( call_user, { call_user } ) );
call_type ::= ( PRIVATE | CONTROLLED | OPEN );
call_user ::= ( user_SAP_addr, modifiablilty, { TC_access_right } );
modifiability ::= ( ON | OFF );
TC_access_right ::= ( TC_id, access_mode, access_QoS );
access_mode ::= ( NONE | READ | WRITE | READ-WRITE );
TC ::= ( TC_id, TC_type, TC_QoS, [ ( TC_user, { TC_user } ) ] );
TC_type ::= (PP_simplex | PP_duplex | PM_simplex | MM_nplex
            | MM_switched_nplex | MM_Mixed_nplex | MM_Lectured_nplex );
TC_user ::= ( user_SAP_addr, TC_access_right );
GTC ::= ( GTC_id, { TC } | { GTC } | ( { TC }, { GTC } ),
         [ { Relation } ], [ ( GTC_user, { GTC_user } ) ] );
GTC_user ::= ( user_SAP_addr, { TC_access_right } );
relation ::= ( relation_id, relation_type, ( TC_id | GTC_id, { TC_id | GTC_id } );
relation_type ::= ( synchronization, audio_activated_video );

( ) exactly one,    [ ] zero or more,    { } one or more,    | or
```

Figure 4: Formal Specification of Call Structure

appropriate TCs according to the data characteristics and the data interaction patterns.

The call structure of the proposed call model can be formally specified as in Figure 4. A call is uniquely identified by the call_id. The type of a call, specified by the call_type attribute, can be PRIVATE, where no external users are allowed to join the call, CONTROLLED, where the call owner has the option of refusing a request to join, or OPEN, where any external user can join the call. Within a call, there exists one or more TCs and/or GTCs. And, there may exist relation objects to specify the relationships among the related TCs and/or GTCs. Two or more users are attached to a call. The characteristic of a call user is represented by the SAP(Service Access Point) address, the right to modify the configuration of the call, and the access rights for the subordinate TCs. If the modifiability is ON, the user is allowed to add/delete TCs and GTCs to/from the call. The access right for a TC specifies a user's access mode and access QoS for the TC. When a user does not access a TC of a call, the user's access mode for the TC is specified as NONE. A user's access QoS for a TC should be constrained by the maximum QoS provided by the TC. The characteristics of a TC are represented by the TC identifier, TC type, TC QoS, and attached TC users. The TC identifier should be unique within a call. The TC_type attribute specifies TC type as in Figure 4. The TC_QoS attribute specifies the maximum QoS provided by the TC. A TC user is characterized by the SAP address and the access right for the TC. Since the users of a TC always are a subset of the users of its parent object, call or GTC, it is not necessary to specify the TC user information when the TC is described as a subordinate object of a call or GTC. The characteristics of a GTC are represented by a unique GTC identifier, the specification of the subordinate TCs and/or GTCs, the optional specification of the relation objects, and the specification of the attached GTC users. A GTC user is represented by the SAP

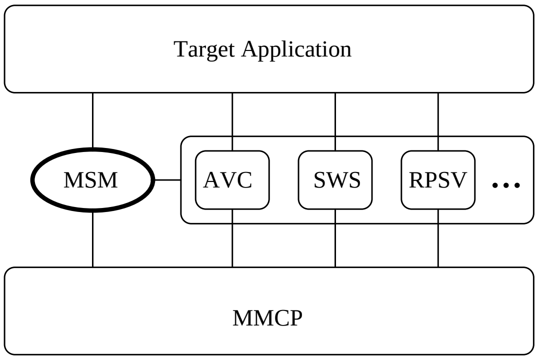

Figure 5: CASE Architecture

address and the access rights for the subordinate TCs. When a GTC is described as a subordinate object of a call or another GTC, the GTC user information is not needed. A relation object is represented by a unique relation identifier, the relation type (synchronization or audio_activated_video, for example), and the specification of the identifiers for the related TCs and/or GTCs.

4 Common Application Service Elements

4.1 Multi-Service Manager

The Multi-Service Manager(MSM) service element in Figure 5 provides the functionalities to manage the communication services of the MMCP and other common application service elements such as Audio-Video Conference(AVC), Shared Working Space(SWS), and Remote Processing of Stored Video(RPSV) according to the requirements of a target application. The configuration information for each common application service element is registered to the MSM so that it can be appropriately configured according to the application requirement. The MSM allocates TC(s) and/or GTC(s) to the common application service elements accordingly. The allocated TC(s) and GTC(s) belong to the call of the target application.

4.2 AVC(Audio-Video Conference)

AVC provides audio and video conference service to the target application. It consists of three modules as depicted in Figure 6, AVCSE(Audio Video Conference Service Element), ASE(Audio Service Element) and VSE(Video Service Element). AVCSE manages and controls ASE and VSE which handles audio and video data respectively. And it executes the requests from MSM in association with MMCP. The requests from MSM is activation of ASE/VSE, deactivation of ASE/VSE and so on. ASE transmits audio data between participants through the TC established by MMCP, and executes the audio control function. Video is handled by VSE in the same manner.

4.3 SWS(Shared Working Space)

SWS provides sharing of applications, and workspace sharing. The application which will be

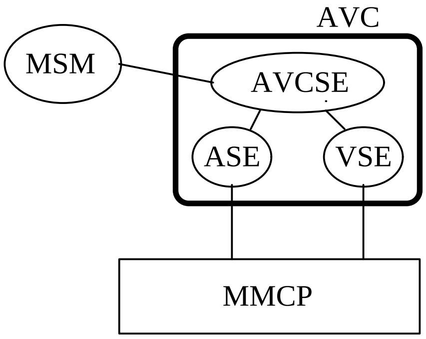

Figure 6: AVC Modules

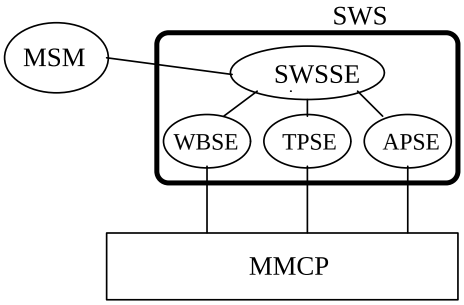

Figure 7: SWS Modules

shared includes whiteboard, telepointer and other cooperation-aware or cooperation-unaware applications. SWS in our architecture has four modules as in Figure 7 : SWSSE(Shared Working Space Service Element), WBSE(WhiteBoard Service Element), TPSE(TelePointing Service Element), ASSE(Application Sharing Service Element). SWSSE activates and deactivates WBSE, TPSE, ASSE when the request from MSM arrives. WBSE provides telepointing, telemarking, telewriting within the shared work space. TPSE enables the participants to share the telepointer. ASSE provides the participants with the cooperation-aware and cooperation-unaware application sharing service.

4.4 RPSV(Remote Processing of Stored Video)

RPSV is the application service element which provides the video multicasting service to the participants. It displays the video data stored in video server or local disk of some participant's workstation to the all participants. And authorized participants can control the display of video data by means of play, fast forward, rewind and so on. RPSV only handles the retrieval and multicasting of video and/or audio data; it doesn't have the function of compressing and storing the audio/video data.

5 Target Application Development

MPS(Multimedia Presentaion System) is under development as the first multimedia multiuser application based on our architecture. It is the application

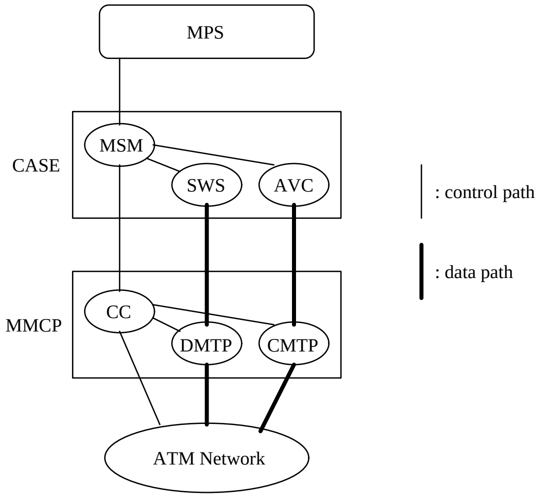

Figure 8: MPS Architecture

to distribute the multimedia data to participants for the purpose of sharing the presentation between participants.

MPS has the following ASEs of CASE layer as shown in Figure 8 : MSM, AVC, SWS, RPSV. The user requests are accepted by user interface of MPS and transported to MSM to be executed. And AVC is responsible for the management of audio/video data. SWS enables the participants to share the application and workspace. CC establishes and releases the TC(Transport Connection). And it controls the TC by means of adding user to TC, deleting user to TC, and so on. DMTP(Discrete Media Transport Protocol) and CMTP(Continuous Media Transport Protocol) transmit the non-real-time data and real-time audio/video data respectively.

We implement MPS over ATM network, which consists of adaptor board in each host and an ATM switch. Other software modules are implemeted in C using X Windows and Motif when applicable.

6 Conclusions

In this paper, we presented three-layered end system software architecture to support multimedia multiuser applications such as computer supported cooperative working, desktop multimedia conferencing, and tele-medicine. Our architecture is composed of the MMCP(Multimedia Multiuser Communication Platform) layer, the CASE(Common Application Service Elements) layer, and the target application layer. The MMCP provides structured and dynamically-configured communication services between participants to the individual applications of CASE, and the CASE layer supports various types of basic application service elements which can be flexibly configured and integrated for the development of target applications. The target application layer is the final application which has a direct interface with the user. The proposed end system architecure will help to develop various and general multimedia multiuser applications and flexibly by the layered architecture. The proposed three-layered architecture may have disadvantages such that the control path length through IPC(InterProcess Communication) is too long because of layering. As an ASE should transmit multimedia data indirectly through the MMCP layer, there may be a performance problem, to be solved by code optimization and process integration. We develop MPS(Multimedia Presentation System) as our first target application. Implementing MPS is the first step toward other multimedia multiuser applications such as tele-medicine and distance education.

References

[1] M. Arango and et. al., "The Touring Machine System," Communications of the ACM, Vol.36, No.1, pp. 68-77, Jan., 1993.

[2] H. Armbruster and K. Wimmer, "Broadband Multimedia Applications Using ATM Networks : High-Performance Computing, High-Capacity Storage, and High-Speed Communication," IEEE JSAC, Vol.10, No.9, pp. 1382-1396, Dec. 1992.

[3] M. Chen and et. al., "Software Architecture of DiCE : A Distributed Collaboration Environment," IEEE Multimedia '92, 1992.

[4] W. J. Clark, "Multipoint Multimedia Conferencing," IEEE Communications Magazine, Vol.30, No.5, pp. 44-50, May, 1992.

[5] G. J. Heijenk and et. al., "Communication Systems Supporting Multimedia Multi-user Application," IEEE Network Magazine, Vol.8, No.1, pp. 34-44, Jan./Feb., 1994.

[6] L. Henckel, "Multipeer Transport Services for Multimedia Applications," Proceedings of HPN '94, 1994.

[7] MPS(Multimedia Presentation System) Service Requirement Specification, Technical Report HEART-MPS100-2.0, Multimedia Lab., Dept. of Computer Engineering, Seoul National University, 1995.

[8] T. Ohmori and et. al., "Multiparty and Multimedia Desktop Conferencing System : Mermaid," IEEE Multimedia '92, 1992.