A Call Model for Multimedia Multiuser Communication Platform

Seungchul Park and Yanghee Choi

Department of Computer Engineering, Seoul National University
Shinlim-dong, Kwanak-ku, Seoul, Korea
e-mail: yhchoi@smart.snu.ac.kr, tel: 82-2-874-2353, fax: 82-2-872-2045

Abstract

Multimedia multiuser applications such as computer supported cooperative working, desktop multimedia conferencing, and distance education require more complex and structured communication services from the underlying communication platform than the conventional applications. The call model of the communication platform defines how the underlying basic data transport objects can be structured to support more complex communication services, and how the structured objects can be managed in a consistent manner by the applications. In this paper, we present a new call model for multimedia multiuser communication platform, based on the analysis of communication requirements for diverse applications. Our call model provides superior structuring capability than the existing call models, and the communication platform based on the call model will support present and future complex multimedia multiuser applications more efficiently. Detailed call control operations are also presented in this paper.

1. Introduction

The conventional communication platforms based on the OSI or TCP/IP provide monolithic point-to-point best-effort communication services. They do not generally satisfy the communication requirements of the multimedia multiuser applications. The new applications require more structured communication services for the integrated control of multiple media, and various types of multipoint communications. Since audio and video have different characteristics from the conventional data, wide range support of QoS(Quality of Service) is required for the multimedia multiuser applications[1,2,3]. Therefore, the underlying communication platform supporting multimedia multiuser applications should be developed in a new paradigm. In the context of the communication platform, the call model defines what kinds of data transport units(objects) are required for the communication platform, and how the underlying basic data transport objects can be structured to support more 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[1,4,5,6,7]. The call model is a foundation from which the communication protocols for the communication platform are developed.

This paper presents an abstract call model for the multimedia multiuser communication platform. It differs from the previous works in the following points. Most existing call models provide finite-level structuring capabilities of basic data transport objects. In the Touring Machine[4], the connectors abstracting communication bridges are grouped as a session to provide a logical communication topology for an application. In [5], two-level structuring capability, call and medium, is provided. In [6,7] a call groups one or more point-to-point simplex or duplex, or point-to-multipoint simplex bearer connections. In these call models, upper layer protocols are required to support more complex communication services. Otherwise, additional management functionality should be added to the individual application. Our call model provides infinite-level structuring capability. Grouping of the basic data transport objects can be nested so that more complex structuring is possible with ease.

Another important feature of our call model is the flexibility to define the basic data transport objects, the basic building blocks in the call model. Most existing call models rely on several pre-defined specific types of basic data transport objects. In our call model, however,

0-8186-7216-1/95 $04.00 © 1995 IEEE

various types of basic data transport objects can be defined according to the application requirements. They are realized at the communication platform with optimal resource allocation, independent of applications. This flexible definition of the basic data transport objects and the powerful structuring capability of our call model help to avoid unnecessary management overhead at the application level to tailor and coordinate the basic data transport objects.

This paper is organized as follows. In section 2, we present the basic communication requirements of the multimedia multiuser applications. The requirements are derived from the analysis of the structure and dynamic behaviors of the multimedia multiuser applications. Based on the communication requirements, a new call model is proposed in section 3. Brief analysis of existing call models is presented here too. The detailed call control operations are presented in the subsequent section.

2. Communication Requirements of Multimedia Multiuser Applications

2.1 Structure and Dynamics of Multimedia Multiuser Applications

The multimedia multiuser applications such as computer supported cooperative working, medical applications, desktop multimedia conferencing, and distance education, have more complex control structures than the conventional network applications. In general, multimedia multiuser applications are composed of multiple individually-controlled basic service elements. Examples of the basic service elements are voice conferencing, remote live video processing, working space sharing, stored multimedia retrieval, file transfer, electronic mail, and so on. Several basic service elements can be grouped as an integrated control unit. For instance, in a multimedia conference application composed of voice conferencing, remote live video processing, and working space sharing service elements, the multiple remote live video processing elements, each of which deals with the video originating from a communicating user, can be managed as a group(for activation or deactivation), or as separate elements. Moreover, the voice conference service element and the grouped remote live video service elements can be grouped as one control unit.

Among the service elements, relationships may exist. The lip-sync and audio-activated-video(remote video activated by the current speaker) are typical examples of the relationships[2,9]. The configuration of the service elements of an application can be dynamically changed during the lifetime. Additional service elements can be added to the application, and the existing service elements can be deleted dynamically. The service elements of an application may require different communication patterns and QoSs, which can be dynamically changed, too.

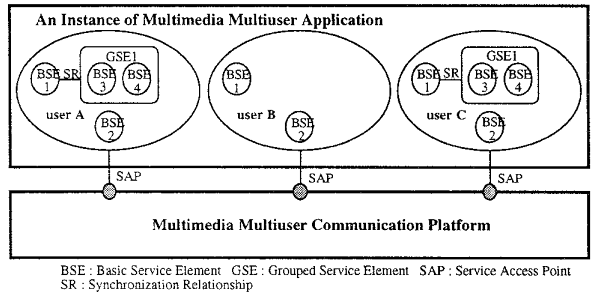


Figure 1 Structure of Multimedia Multiuser Application

Another important aspect of the applications is that multiple users may be involved. Here, users are the communicating entities participating in an instance of an application. They are identified by the addresses of the corresponding service access points. The users of an instance of a multimedia multiuser application may not have equal access capabilities for the service elements of the application. For instance, in a multimedia conference some users may participate with the voice conference and working space sharing service elements without using the remote live video processing service element. Moreover, the number of users for an application and / or its constituent service elements can be dynamically changed during the lifetime.

Figure 1 shows an example structure of a multimedia multiuser application. The application consists of two basic service elements(BSE1 and BSE2) and one grouped service element(GSE1) which groups two other basic service elements(BSE3 and BSE4). There exists synchronization relationship between the BSE1 and GSE1. In the application instance , three users(user A, user B, and user c) are involved. Since the user B does not access the GSE1, the users of the GSE1 are just user A and user C, subset of the users of the application instance.

2.2 Communication Requirements

In this section, we present the communication requirements for the underlying communication platform to support the control structure and dynamic behaviors of the multimedia multiuser applications. First, how to structure the communication platform is explained in terms of the multi-connection communication. The requirements to support multipoint communications are followed. After then, how to support the relationships of the applications will be presented. Finally, the access control requirements to control the dynamics of the communication platform will be described.

1) **multi-connection communication** : To provide communication services for the structured and dynamic multimedia multiuser applications, the communication platform needs to be structured in a modular and hierarchical way. This means that the association between the communicating users and the communication platform to provide communication services for an application should be able to consist of multiple distinguished and individually-controlled communication objects, so called logical connections. They are assigned to the constituent service elements of the application correspondingly. This is because the communication resource requirement for each service element may differ from each other, and the allocated resource should be able to be independently controlled according to the dynamics of the corresponding service element. The structure of the multi-connection association must correspond to the control structure of the application so that dynamic behaviors of the application can be reflected by the communication platform appropriately.

2) **multipoint communication** : The logical connections for the basic service elements of multimedia multiuser applications should be able to provide various types of multipoint communication as well as point-to-point communications. The types of multipoint communication may include point-to-multipoint simplex, point-to-multipoint duplex, and multipoint-to-multipoint nplex. Users attached to a multipoint connection may not have equal access capabilities. They may have different access modes(read, write, or readwrite), different data formats, and different QoS requirements. Therefore, the corresponding multipoint connections should be able to be configured to support the heterogeneity. In addition, new types of multipoint connections with added value should be considered. For a conference call, a multipoint connection handling voice mixing is preferred to be defined at the communication platform, reducing the application complexity.

3) **relationships** : In order to support the relationships among the service elements of an application, the communication platform should be able to control the communications over the corresponding connections. The service-level inter-media synchronization, for example, is easily achieved when the timing constraints among the data streams of the connections are kept at the communication platform[14,15,16]. Another example is the audio-activated-video relationship. The communication platform can support the relationship by activating only the video connection originating from the current speaker. The communication platform must provide means to specify and control various types of relationships among the related connections.

4) **access control** : Communicating users' capability to access the connections provided by the communication platform should be able to be controlled. Generally, the communicating users should be allowed to select the connections to communicate according to their capability. But the selection should be constrained in some cases. For example, in a multimedia conference, some observing users are constrained to operate in only receiving mode. The communicating users' right to send data over a connection may have to be controlled during communication to guarantee consistent execution of the corresponding service element. Usually this is called floor control[8]. The dynamic change of the configuration of the multi-connection associations should be able to be controlled so as to be performed by only permitted users. This change may include dynamic addition and deletion of connections to/from the association, and dynamic addition and deletion of users to/from the association and the accompanying connections.

3. Multimedia Multiuser Call Model

In the context of the communication platform, the call model defines what kinds of data transport units(objects) are required for the communication platform, and how the underlying basic data transport objects can be structured to support more 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 propose a new call model which meets the communication requirements for the multimedia multiuser applications. Before presenting our call model, the existing call models are briefly analyzed.

3.1 Existing Call Models

In this section, the call models of the Touring Machine[4], the G. J. Heijenk et al.'s model[5], and the L. Henckel's model[6] are analyzed. They are intended to provide well-defined, but network and application independent communication services for multimedia multiuser applications. Our analysis is focused on verifying the structuring capabilities of the call models.

The Touring Machine system is a software platform developed at Bellcore to provide an infrastructure on which applications requiring complex multimedia communications can be developed, independent of the actual network fabric used to provide transport[4]. To achieve this goal, the system was developed based on the two-level, session and connector, abstract call model. As in Figure 2, the multimedia logical communication topology for a multiuser application is represented as a session.

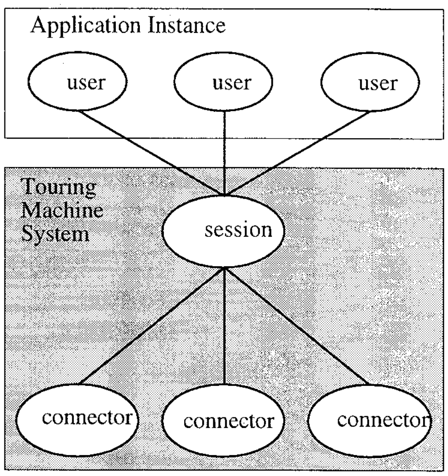


Figure 2 Call Structure of Touring Machine[4]

The logical topology of a session is specified as a set of typed connectors. A connector represents a multiway transport connection among endpoints(logical ports). A connector is an abstraction of a communication bridge, including point-to-point as well as multipoint connections. Since bridging is a medium-specific operation, connectors are typed by medium. Users can access the connectors in only three ways; IN, OUT, or IN-OUT. The problems of this call model can be summarized as follows.

1) Since this call model allows only two-level structuring of basic data transport objects (connectors), more complicated communication services cannot be supported by the underlying communication platform.
2) The relationships for the related connectors can not be specified in this model.
3) Since a user's capability to access the connectors can be represented by only IN, OUT, or IN-OUT, it is difficult to support service elements with heterogeneous QoS users. As the hierarchical coding techniques are widely emerging, the user-selected QoS support becomes more important[17, 18].

G. J. Heijenk et al. at University of Twente, proposed a service description for the communication systems supporting multimedia multiuser applications[5]. The description is based on a two-level, call and medium, abstract call model with the capability to specify media relations, as shown in Figure 3.

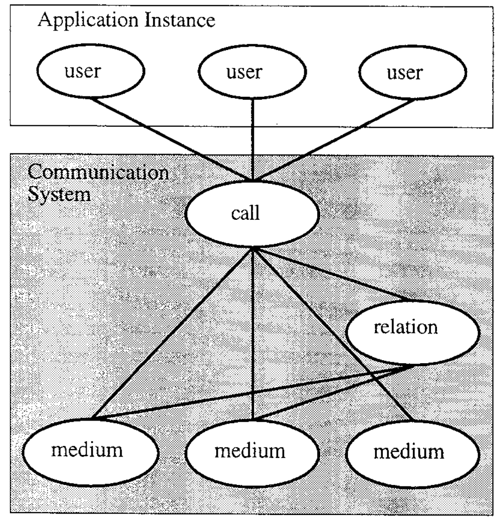


Figure 3. Call Structure of G. J. Heijenk et al.'s Model[5]

In this model, the dynamic association between multiple service users and the communication system is represented as a call, where all interactions between users and the service provider concerned with one sort of information are covered by the medium, the basic constituent of the call. To describe relationships between media, an object called "relation" is defined in the call model. The access mechanism to the medium is similar to that of the Touring Machine. Users can be attached to a medium in only three ways; IN, OUT, or IN-OUT. The same problems of the Touring Machine's call model, except the relation, exist in this model.

L. Henckel at GMD, presented a description of multipeer transport services for multimedia applications based on the separation of transport call(TL) and transport connection(TC)[6]. A transport call groups multiple simplex transport connections, point-to-point or point-to-multipoint, which can be individually identified and controlled. The quality of services of the transport connections of a transport call may differ from each other. This model can support more sophisticated communication topologies by combining the transport connections into the transport calls.

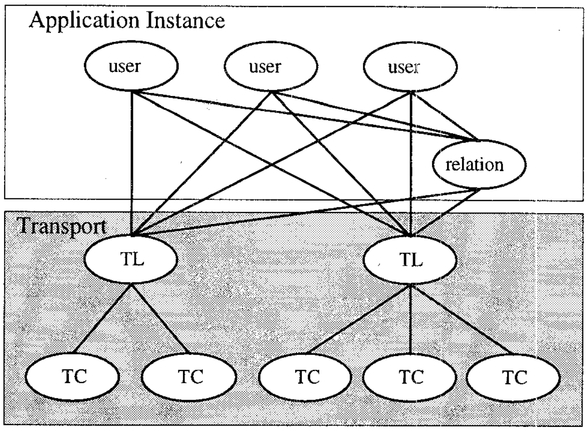


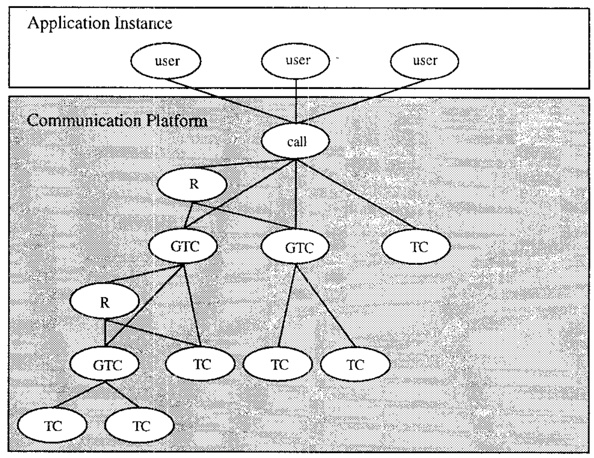
Figure 4 Call Structure of L. Henckel's Model[6]

In this model, an application composed of multiple related but individually-controlled service elements is required to manage multiple transport calls and their relations by itself, because the call model does not provide any mechanism to distinguish the communication services for the different service elements in the same application. Therefore the complexity of the multimedia multiuser applications is greatly increased. This call model also lacks the ability to describe the generic communication agents commonly required by the multimedia multiuser applications. Examples of the agents are voice mixer, video switch, and data bridge, which are transparent from applications. With this model, it is obliged to define additional session management protocol at the application level in order to model the agents.

3.2 A New Call Model

Figure 5 shows an example of the general call structure of our call model. In our 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 grouped TCs and/or GTCs integrally as well as controlling the constituent TCs and/or GTCs individually. The relation objects which specify the relationships among the subordinate 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 more than one subordinate objects to exist. And the relation objects should have more than two related objects to exist.

This nested grouping of basic data transport objects provides more powerful structuring capability than the existing finite-level call models. The structuring of the basic data transport objects can be also adjusted according to the control structures of applications. In our model, a call is a special GTC which represents the dynamic association between the communicating users and the communication platform to provide the whole communication capability for an instance of an application. It is the root of a grouping hierarchy of TCs, and it abstracts a communication platform instance. The global view of a call can be changed during the call lifetime. 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. Users can join and leave the call.



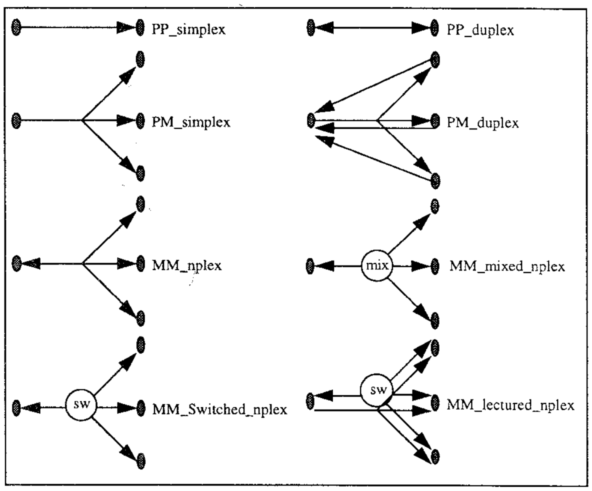
TC : Transport Connection GTC : Grouped TC R : Relation

Figure 5 Call Structure of the Proposed Call Model

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 TC subordinate to the GTC. A user's local view of a call can be changed during 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.

The TC is a typed object. Rich set of typed TCs can be defined. Figure 6 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.



PP : Point-to-Point PM : Point-to-Multipoint
MM : Multipoint-to-Multipoint Mix : Mixing
SW : Switching

Figure 6 Example Definition of TC Types

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

For each TC, applications can specify appropriate QoS(Quality of Service) requirement. The requirements can be represented in different ways according to media types. For continuous media TC, the end-to-end delay, jitters, and throughput can be defined[11]. The communicating users of a TC may have different QoS requirements from one another. And, the QoS requirement of a TC may be dynamically changed during lifetime by the permitted users.

3.3 Application : A Multimedia Conference Case

In this section, we discuss how the proposed call model can be applied to the development of a multimedia conference application which is composed of voice conferencing, remote live video processing, and shared working space with telepointer service elements. For remote live video processing, multiple PM_simplex video TCs can be used to transfer the live video of each user to others. All live video TCs can be grouped as a video GTC for the integrated control of activation, deactivation, extension to late comer, reduction from early leaving user, and so on.

For the voice conferencing service element, an MM_mixed_nplex voice TC which mixes the voice of all communicating users may be allocated. There exists a synchronization relationship between the voice TC and the video GTC. The voice TC, the video GTC, and the synchronization relation can be grouped as another AV conference GTC. For the telepointer service element, an MM_switched_nplex data TC may be allocated to support the floor-controlled access for the telepointer effectively. And, an MM_nplex data TC can be assigned to the whiteboard service element, for example. The telepointer TC and the whiteboard TC can be grouped as a shared work space GTC for integrated control. There may exist another synchronization relationship between the telepointer TC and the AV conference GTC. This means that the movement of the telepointer should be synchronized with the voice and the speaker's lip in the video.

4. Call Control Operations

In this section, we present in detail the control operations required to support the structuring capability and the dynamic behaviors of our call model. Control operations for call establishment/release, dynamic connection control, and dynamic participant are described here. Other operations such as change-owner, change-QoS, floor-control etc are defined in [12].

4.1 Call Establishment and Release

Table 1 shows the parameters in the operations for the call establishment and release. The call establishment operation(**establish_call**()) requires one parameter to specify the characteristics of the proposed call and another parameter for the user-list. Each call have to be uniquely identified within the communication platform by *call_ID*. The *call_type* can be defined as PRIVATE, where no external users may be added except at the invitation of the call owner(the initiator), CONTROLLED, where the owner has the option of refusing any attempt to add a user, and OPEN, where any user may be added without the owner having the option to refuse. The parameter *sub_list* specifies the objects(TC and/or GTC) handled in the requested call. It specifies the TC-list, or GTC-list, or both TC-list and GTC-list. The relation objects specifying the relationships among the related TCs and/or GTCs may be included in *sub_list*. Each TC is uniquely identified within the call by *TC_id*, and information on *TC_type* and *QoS* requirement are provided. For a GTC, a unique identifier *GTC_id* is attached.

For each user of the *call_user_list*, the constraints on the call modifiability and the access-rights for the TCs in the call are specified. If the call modifiability is ON, the user is permitted to modify the call configuration(i.e., addition and deletion of TCs and GTCs to/from the call). With *TC_access_right*, the call owner can limit each user's access capability to each TC. The call owner can destroy an existing call identified by *call_ID* by invoking the **release_call** () operation.

Table 1 Control Operations for Call Establishment and Release

establish_call(call : (call_id, call_type, sub_list :
((TC_list : {TC : (TC_id, TC_type, QoS)} |
GTC_list : {GTC: (GTC_id, sub_list)} | (TC_list,
GTC_list)), [relation_list : {relation_id,
relation_type, ((TC_id | GTC_id), {TC_id |
GTC_id})}]])**, call_user_list :** {user_SAP_addr,
modifiability, TC_access_right_list : {TC_id,
TC_access_right}});

release_call(call_id);

Legend :	( )	exactly one	[ ]	zero or one
	{ }	one or more	,	followed by
	|	or	:	consisting of

4.2 Dynamic Connection Control

The permitted users can modify the call configuration with the operations in Table 2. To add a TC to a call or a GTC of the call, the user must specify the target identifier, the specification of the TC to be added, and the proposed user-list for the TC. This is same for the operation to add a GTC(**add_GTC**()). However, in this case the access rights for all subordinate TCs should be listed for each user. The permitted user can delete a TC(GTC) by invoking the **delete_TC**()(**delete_GTC**()) operation.

Table 2 Control Operations for TC and GTC

add_TC(to: (call_id | GTC_id)**, added_TC :** TC,
TC_user_list : {user_SAP_addr,TC_access_right});

add_GTC(to : (call_id | GTC_id)**, added_GTC :**
GTC**, GTC_user_list :** {user_SAP_addr,
TC_access_right_list});

delete_TC(deleted_TC : TC_id);

delete_GTC(deleted_GTC : GTC_id);

4.3 Dynamic Participant Control

The communicating users of a call can invite other users to join the call by using the **add_user_to_call**() operation of Table 3.

Table 3 Operations for Participant Control

add_user_to_call(to : call**, added_call_user :**
(user_SAP_addr, modifiability,
TC_access_right_list));

add_user_to_GTC(to : GTC**, added_GTC_user :**
(user_SAP_addr, TC_access_right_list));

add_user_to_TC(to : TC**, added_TC_user :**
(user_SAP_addr, TC_access_right));

delete_user(from : (call_id | GTC_id | TC_id),
deleted_user : user_SAP_addr);

leave(from : (call_id | GTC_id | TC_id));

In this case, the calling user describes the characteristics of the on-going call in the *to* parameter. The call characteristics are specified as in the case of the call establishment. The call modifiability and the access-rights for the TCs are also specified for the invited user. The invited user doesn't have to accept the TCs as proposed, but he can use them according to his capability within the proposed constraint. Via the **add_user_to_TC**() and **add_user_to_GTC**(), the current users of a call can be invited to join a TC or GTC that he is not currently attached to. The call owner can delete users from the call, GTC, or TC by using the **delete_user**() operation. And, the communicating users can leave the call, GTC, or TC by invoking the **leave**() operation.

5. Conclusion

In this paper, we presented a new call model for multimedia multiuser communication platform, based on the analysis of communication requirements for diverse applications. Our call model provides superior powerful structuring capability of the basic data transport objects than the existing call models, and it can support complex multimedia multiuser applications in more efficient way. A call control protocol to handle the call control operations defined in this paper is designed and under implementation on the ATM-based multimedia testbed constructed in our laboratory[13, 19]. It is planned to develop at least two multimedia multiuser applications, desktop tele-presentation and tele-medicine on top of the common communication platform under the context of Korean Information Infrastructure.

References

[1] S. Minzer, "A Signaling Protocol for Complex Multimedia Services," IEEE Journal on Selected Areas in Communications, Vol. 9, No. 9, pp. 1383-1394, Dec. 1991.

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

[3] G. Coulson, et al., "Protocol Support for Distributed Multimedia Applications," The 2nd International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, Nov. 1991.

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

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

[6] L. Henckel, "Multipeer Transport Services for Multimedia Applications," The 5th IFIP Conference on High Performance Networking, Grenoble, June 1994.

[7] L. A. Crutcher and A. G. Waters, "Connection Management for an ATM Network," IEEE Network Magazine, Vol. 6, No. 6, pp. 42-55, Nov. 1992.

[8] K. Watebe, et. al., "Distributed Desktop Conferencing System with Multiuser Multimedia Interface," IEEE Journal on Selected Areas in Communications, Vol. 9, No. 4, pp. 531-539, May 1991.

[9] R. Steinmetz, "Multimedia Synchronization Techniques: Experiences based on Different System Structures", The 4th IEEE ComSoc International Workshop on Multimedia Communications, Monterey, California, April 1992.

[10] B. Wolfinger and M. Moran, "A Continuous Media Data Transport Service and Protocol for Realtime Communication in High Speed Networks," The 2nd International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, Nov. 1991.

[11] A. Camphell, G. Coulson, and D. Hutchison, "A Quality of Service Architecture," Computer Communication Review, Vol. 24, No. 2, pp. 6-27, April 1994.

[12] Seungchul park, et al., "Multimedia Communication Platform System Specification," Technical Report, Department of Computer Engineering, Seoul National University, Seoul, Korea, Feb. 1995.

[13] Seungchul Park and Yanghee Choi, "Multimedia Communication Architecture in SMART(Seoul Multimedia Advanced Research Testbed)," Multimedia Communications'93, Banff, Alberta, April 1993.

[14] J. Escobar, C. Patridge, and D. Deutsch, "Flow Synchronization Protocol," IEEE/ACM Transactions on Networking, Vol. 12, No.2, pp. 111-121, April 1994.

[15] K. Ravindran and V. Bansal, "Delay Compensation Protocols for Synchronization of Multimedia Data Streams," IEEE Transactions on Knowledge and Data Engineering, Vol. 5, No. 4, pp. 574-589, Aug. 1993.

[16] Seungchul Park and Yanghee Choi, "An Adaptive Multimedia Synchronization based on the Delay Offset and Playout Rate Adjustment," The 1st Joint Workshop on Multimedia Communications, Taejeon, Korea, Oct. 1994.

[17] ISO/IEC JTC1/SC21, Information Technology - Quality of Service Framework - Part 2 : Basic Framework, N9309, Jan. 1995

[18] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP : A New Resource ReSerVation Protocol," IEEE Network Megazine, Sept. 1993

[19] Seungchul park, et al., "Multimedia Communication Platform Design Specification," Technical Report, Department of Computer Engineering, Seoul National University, Seoul, Korea, May 1995.