

オブジェクト指向に基づいた マルチメディア・マルチユーザアプリケーション呼モデル

朴 美娘¹

三菱電機(株) 情報技術総合研究所 高速通信部

神奈川県鎌倉市大船5-1-1

mirang@kousoku.isl.melco.co.jp

現在、高速・広帯域通信網の構築が進むにつれて、多種多様なマルチメディア・マルチユーザアプリケーションの実用化が期待されている。ここでは、このようなアプリケーションを実現可能にするオブジェクト指向に基づいたマルチメディア呼モデルを提案する。このモデルは、アプリケーションを実現するために必要とされる多数の異なるサービス进行处理するオブジェクトより構成される。さらに、このモデルでは各オブジェクトごとにオブジェクトの属性や実行可能な制御オペレーションを定義することによって、ユーザの要求を実現する高度なサービス制御が可能になる。

1.はじめに

現在、ATM技術による通信網の高速・広帯域化が進むにつれて、複数のメディアを使って複数の相手と情報の授受を行なうことができるマルチメディア・マルチユーザアプリケーションの実用化が期待されている。しかし、これらは、従来のOSIまたはTCP/IPの上では実現が困難であるのが現状である。従って、これらのマルチメディア・マルチユーザアプリケーションを高速通信網の上に乗せるための、より柔軟で高度な新しいマルチメディア・マルチユーザ通信プラットフォームの構築が必要になる。現在、このようなマルチメディア通信プラットフォームの構築に関する研究・開発が盛んに行われている[1, 2]。

本稿では、マルチメディア通信プラットフォームを構築するための基盤になるマルチメディア呼モデルについて検討する。特に、呼モデルのコンテキストの中で、マルチメディア・マルチユーザアプリケーションを実現するためにどのようなデータ伝送ユニットが要求されるかを定義する呼モデルについて検討する。このような呼モデルは、新しいマルチメディア通信プラットフォームの上でマルチメディアプロトコルを開発するための基盤になる。

現在、このような呼モデルに関する研究は、大きく二つのアプローチで行われている。一つは、ユー

ザが要求するアプリケーションを実現するために、サービス提供者である通信網を意識せず、アプリケーション自体の論理的仕様だけに基づいて、それらを実現する通信の抽象的なモデルを構築する研究である[2, 6, 8]。これらの研究では、アプリケーションの論理的通信トポロジだけを提案しているので実現が難しいという問題点がある。

他のアプローチとしては、サービスを提供する通信網を意識して、その上でアプリケーションを実現するための呼モデルに関する研究が行われている[9, 14]。ここでは、サービスの提供者である通信網を想定して、アプリケーションの実現性に向けて帯域管理、柔軟なコネクションの接続方法などを解釈しなければならぬという問題点がある。さらに、アプリケーションを実際に利用するユーザの様々な要求に細かく対応できないという問題点もある。

本稿では、サービスを提供する通信網は広帯域通信網(B-ISDN)であるという想定で、ユーザの要求を満たす高度なサービスを実現するためのサービス制御方式について検討する。そこで、高度なサービス制御方式を実現する呼モデルを提案する。このモデルは、多くのアプリケーションを分散処理環境の上で効率的に実行させるために有効なオブジェクト指向の概念に基づいている。特に、通信プラットフォームを構築する際に重要な項目になる呼

¹ An Object-oriented Call Model for Multimedia-Multiuser Applications

Mi Rang Park (Information Technology R&D Center, Mitsubishi Electric Corp.)

コネクション/メディア制御に注目している。また、このモデルにおける呼制御方式は、ビデオ・オン・デマンドサービスや遠隔授業サービスを実現するために必要となる第三者呼制御を可能とする。さらにこのモデルでは、高度なサービス制御を行うためのサービス制御要素を各オブジェクトのオペレーションとして定義することによって、ユーザが呼の動的振る舞いを制御可能になる。

以下、2章ではまず、マルチメディア・マルチユーザアプリケーションを実現するための呼モデルの構築要求条件について検討する。3章ではマルチメディア・マルチユーザ呼を分類し、呼を管理するための呼状態モデルを定義する。最後に4章では、オブジェクト指向に基づいた呼モデルを提案し、呼制御オペレーションを定義する。

2. マルチメディア・マルチユーザアプリケーション呼モデル

マルチメディア・マルチユーザアプリケーションとは、CSCW(Computer Supported Cooperative Work)、医療アプリケーション、デスクトップ・マルチメディア会議、遠隔教育などのように、映像、音声、静止画やデータなど複数のメディアを用いて、複数の相手または複数のマシンと情報の授受を行なうサービスである。以下では、このようなアプリケーションを実現可能にする呼モデルの構築について検討する。

2.1 マルチメディア・マルチユーザアプリケーションアーキテクチャ

マルチメディア・マルチユーザアプリケーションは、ユーザが端末、STB(Set Top Boxes)、情報処理装置などを用いて様々なネットワークタイプ(例えば、ISDN/B-ISDN, PSDN, CATV, LAN...)にアクセスすることによって利用可能になる。ここでは、ユーザがアクセスしたアプリケーションがどのように実現されるかを定義するために、図1のようなマルチメディア・マルチユーザアプリケーションの構造を考える。

まず、ユーザが利用可能なマルチメディア・マルチユーザアプリケーションが存在するアプリケーションレベル(Application Level)が存在する。アプリケーションレベルにおける各アプリケーション(AP:Application)は、ユーザがインスタンス化することによって動作可能になる。

ユーザがインスタンス化したアプリケーシ

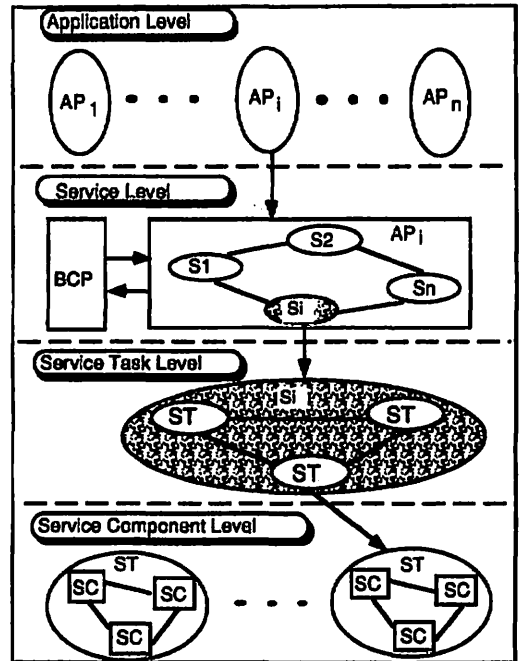


図1. マルチメディア・マルチユーザアプリケーションアーキテクチャ

ョンは、サービスレベル(Service Level)の呼処理装置(BCP: Basic Call Processing)によってそのアプリケーションを実現するためのいくつかのサービス(S: Service)が起動され、サービス間の組み合わせとして構成される。ここで定義されるサービスは、現在存在するアプリケーションや将来実現できるアプリケーションをサポートしなければならず、多数の可能なアプリケーションに共通な機能を含まなければならない。また、一般的概念に影響せずに拡張や変更することが容易でなければならない。ここで、アプリケーションを構成するサービスの組み合わせは、アプリケーションの実行時におけるユーザの要求によって他のサービスが追加されたり削除されたりして動的に変化可能である。

従って、サービスレベルにおける各サービスは、他のアプリケーションの定義に再利用可能なオブジェクトとして定義する。

サービスレベルで定義されたサービスをインプリメントするためには、サービスタスクレベル(Service Task Level)で実際にそのサービスに必要なとされるサービスタスク(ST: Service Task)を構成する。例えば、そのようなサービスタスクには、コネクションタイプを定義するサービスタスクとサービス

制御を行うタスクなどがある。

サービスタスクレベルで定義された各サービスタスクは、さらにサービスコンポーネントレベル(Service Component Level)で分散処理環境におけるサービスコンポーネント(SC: Service Component)を用いて実行される。

以下では、このようなマルチメディア・マルチユーザアプリケーションアーキテクチャに基づいてユーザがインスタンス化したアプリケーションを実現可能にする呼モデルの構築について検討する。

2.2 マルチメディア呼モデル構築要求条件

ここでは、マルチメディア・マルチユーザアプリケーションを実現可能にする通信プラットフォームを検討するに当り、特に呼モデルを構築する際に考慮すべき要求条件について述べる。

(1) マルチコネクション通信：

本稿で対象にしているマルチメディア・マルチユーザアプリケーションは、従来の通信サービスと違って複数のメディアによる基本サービスの組み合わせによって構成される。従って、呼モデルの構築においては、これらの複数のサービスに対応するマルチコネクション通信を考慮する必要がある。また、各アプリケーションはあるサービス実施中に他のサービス機能の追加/削除、メディアの変更/追加/削除等を自由に行うなど、柔軟なアプリケーションの動的振る舞いが反映されるようにコネクション間の通信を制御可能にしなければならない。

(2) マルチポイント通信：

オン・デマンド型サービス、会議型サービスなどのようなマルチメディア・マルチユーザアプリケーションでは、マルチパーティ通信を考慮する必要がある。従って、このようなアプリケーションを構成する各サービスに対する論理的コネクションは、ポイント・ツー・ポイントコネクションだけではなくマルチポイントコネクションなど様々なタイプも提供しなければならない。さらに、アプリケーション実行途中ユーザの要求によるメンバの追加/退席を自由に行うなど、マルチポイントコネクションを制御可能にしなければならない。

(3) 関係：

一般に、アプリケーションを構成する複数のサービス間には関係が存在するので、このような関係も

呼モデルの上で支援しなければならない。すなわち、関係に対応する各サービスのコネクションの上で通信を制御しなければならない。例えば、複数のメディアの組み合わせによって構成されるアプリケーションにおけるメディア間の同期は、各メディアに対応するコネクションのデータストリームの中の時間制約が呼モデルで守られることによって取られる。このように、関連するコネクション間における様々なタイプの関係を明確に表して制御可能な呼モデルを提供しなければならない。

(4) アクセス制御：

マルチメディア会議アプリケーションにおいて、あるユーザは音声会議サービスに参加しながら遠隔映像処理サービスを使わずに共同作業(working space sharing)サービスを利用するなどアプリケーションを構成する各サービスに対するユーザのアクセス能力(サービスに関する要求)はそれぞれ異なる。ユーザは、それぞれのアクセス能力によって通信するためのチャンネルを選ぶ。しかし、チャンネルの選択はいくつかのケースだけに限られている。

従って、ユーザが呼モデルによって提供されるコネクションにアクセスする能力を制御可能にしなければならない。例えば、アプリケーションに参加するユーザの数やそのアプリケーションを構成する基本サービスは、実行の間動的に変わることができる。また、アプリケーションに参加するユーザが観測ユーザ(observer)の場合は、アクセスモードが受信モードの時にだけ動作するように拘束されている。

すなわち、各コネクションの上にデータを送ろうとするユーザの権利は、そのコネクションに対応する基本サービスの実行を保証するために通信の間で制御されなければならない。ここでは、このようなマルチコネクション配列の動的な変化を限られたユーザによってだけ行われるように制御しなければならない。

3. マルチメディア・マルチユーザ呼

ここでは、前章で述べたマルチメディア呼モデルの構築要求条件を満たす呼モデルを構築するための準備段階として、マルチメディア・マルチユーザ呼における諸定義を行う。そして、様々な性質によって分類されるマルチメディア・マルチユーザ呼を定義する。

3.1 諸定義

通信サービスの呼処理機能には、サービスを実現するためのサービス成分の割当、解除だけではなく、呼の設定や切り離し手続きなども含む。以下では、このような機能を行うために必要とされる要素を定義する。

【定義1】参加者(Participants)

ある呼における参加者というのは、個人のユーザか、他のアプリケーションの実行途中参加する別の呼である。これらを以下のように分類して定義する。

- ・簡易参加者(simple participants)：個人のユーザ
- ・スーパー参加者(super participants)：個人のユーザではない別の呼全体 □

呼に参加者が存在すると、アクティブ状態になり、参加者がすべて離れると非アクティブ状態になる。このように呼は、サービスの実行中にアクティブになったり、非アクティブになる。

【定義2】呼属性(Call Attributes)

呼の属性というのは、呼識別子、呼タイプなどのように呼の性質を表すものである。

【定義3】呼オーナー(Call owner)

最初に呼を初期化したものとして、呼の制御権を持つ参加者。

【定義4】呼制御者(Call controller)

サービス実行中に呼をアクティブ状態にしたり、非アクティブ状態にすることができる参加者。

【定義5】アクセスモード(Access modes)

アクセスモードは、呼の参加者の特別な性質として、メディア(例えば、オーディオとビデオ)と呼に対するアクセス方向(in, out, inout)で構成される。アクセスモードには、以下のような2つのタイプがある。

- ・Permitted access modes (A_p):呼の初期者(オーナー)によって表されるアクセスモード。
 - ・Desired access modes (A_d):呼に参加する参加者によって要求されるアクセスモード。
- ここで、 A_p は A_d の部分集合である。 □

3.2 呼の分類

以上で定義したものは、ユーザの要求やサービスの実行によって変わる。本論文では、以下のような性質によって、マルチメディア・マルチユーザ呼をいくつかのタイプで分類する。

(1)オープン性：呼の動的な性質による分類

- ・Private call(Static call)：最初に初期化した後、参加者またはその呼の属性を変更しない呼。
- ・Open call(Dynamic call)：いつでも新しいユーザが参

加したり、呼や参加者の属性を変えられる呼。

(2)オーナーシップ：ユーザの追加・退席等の要求に対する決定権をオーナーが持つかによる分類

- ・dependent call：呼のオーナーが、新しいユーザの参加に対して拒否する権利を持つ呼。
- ・undepended call：新しいユーザの参加の要求に対して、呼のオーナー自身が拒否できない呼。

(3)相関性：複数の呼の間での関係による分類

- ・Unrelated call：複数の呼の間で属性を共有しない呼。
- ・Related call：以下のような文法的なコンテキスト(semantic context)を共有する呼。

(1)親呼(parent call)から子呼(child call)への呼の属性や参加者の属性の継承

(2)親呼の下にある兄弟呼の集成

(4)対称性：参加者のメディアの使用による分類

- ・Symmetric call：呼の参加者全員に通信するためのメディアについて同一メディアを使うように要求する呼。

・Asymmetric call：参加者各自が通信するためのメディアを選択できる呼。

(5)並行性：ユーザが複数の呼に参加できるかどうかによる分類。

・Sequential call：ユーザがある時点で正確に1つの呼に参加する呼。しかし、ユーザはある呼を保留して他の呼に参加するか、また、後で最初の呼を再び始めることができる。

・Concurrent call：複数のユーザが複数の呼に同時に参加することができる呼。

これは、CSCWなど共同作業(collaboration)のための実際的な環境を提供している。

3.3 呼状態モデル

以上で定義したマルチメディア・マルチユーザ呼の実行を設定し制御するために呼管理が必要になる。ここでは、呼参加者の進行を状態の系列によって表して呼管理を行う。以下では、このような呼状態モデルについて検討する。

呼管理を行うための呼の参加者に対する状態遷移図を図2のように表す。ここで、各参加者は6つの状態を持つことができる。すなわち、呼の初期者(caller)の状態としてIDLE, INITIATING, CONFIRM, ACTIVE, 呼参加者(callee)の状態としてIDLE, NOTIFIED, RESPONSE, ACTIVE状態を持つ。これらの各状態について簡単に説明すると以下のようになる。

(1)IDLE：各参加者が何にもやってない状態

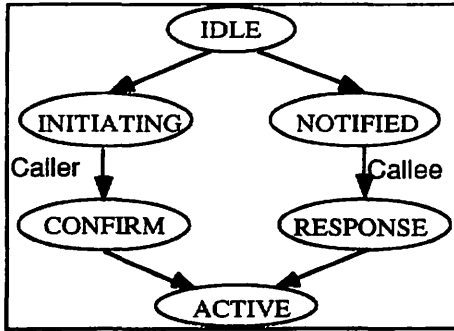


図2. 呼参加者の状態移行図

- (2)INITIATING : 呼の初期者が新しい呼設定要求を出す直前の状態
- (3)CONFIRM : 参加者に呼に参加するように要求を出した状態
- (4)NOTIFIED : 新しい呼に参加するよう要求を受けた状態
- (5)RESPONSE : 参加者が呼に参加しようとする状態
- (6)ACTIVE : 参加者が招待を受け入れた状態

このような呼の状態は、呼管理者によって作られた呼モデルで維持される。呼管理者は、呼参加者の名前とアドレスを管理し、呼の設定に必要なネゴシエーションを仲介する。そして、リソースの割当てを管理、アクセス条件およびすべての参加者の状態を管理し制御する。

従って、本論文では呼管理を行うために、呼における各参加者の情報を以下で示すような5項目によって表すことにする。

【定義6】 呼の参加者の表現 (D)

$$D = (u, i, s, Acc_p, Acc_d)$$

where, $Acc = \{(m, d) \mid m \in Media, d \in Direction\}$

ここで、u (user name) はそのサービスに参加するユーザの名前、i (indication)は端末のアドレスのようなユーザの位置、s (state)はユーザの状態、 Acc_p (permitted Access mode)、 Acc_d (desired Access mode) はそれぞれアクセスモードを表す。 □

このように定義された各参加者の情報を用いて、全体の呼の状態モデルを定義することができる。例えば、呼の参加者がnで、各参加者がk状態を持つならば、その呼には少なくともk個の可能な状態が存在する。このような呼の状態モデルをBNF(Backus-Naur Form)記法を用いて以下のように定

義する。

【定義7】 呼の状態モデル(Cs)

$$Cs := \{ D \}^* \mid \epsilon$$

$$D := (U, I, S, Acc_p, Acc_d)$$

$$U := u$$

$$I := i$$

$$S := IDLE \mid INITIATING \mid CONFIRM \mid NOTIFIED \mid RESPONSE \mid ACTIVE$$

$$Acc := \{ (M, Dr) \}^* \mid \epsilon$$

$$M := Voice \mid Video \mid Data$$

$$Dr := In \mid Out \mid InOut$$

ここで、 $\{ x \}^*$ はxの1回以上の出現を表す。 □

このように定義された呼の状態モデルでは、呼に対するユーザのローカルビューだけではなく、呼の明確なグローバルビューを表すことができる。

また、呼管理者はユーザからの入力に対応して状態遷移を初期化することもできる。ある呼における各参加者のすべての状態遷移は、呼管理者に報告される。呼管理者は、状態遷移の結果を他の参加者に再分配し、その結果は各参加者自身の状態遷移に影響される。呼のオーナーは、その呼に参加するようにユーザ(callee)とアクセスモードを指定することができる。

4. オブジェクト指向マルチメディア呼モデル

ここでは、以上で述べたようなマルチメディア・マルチユーザ呼を実現する呼モデルの設計を行う。特に、呼の設定・切り離しだけではなく、そのモデルの上でサービス制御を柔軟に行うことができる高度な呼モデルを提案する。

4.1 オブジェクト指向呼モデル

本稿では、2章で述べたような要求条件を満たすためにEXPANCE呼モデル[8]と同様に、オブジェクト指向アプローチに基づいて呼モデルの設計を行う(図3)。そこで、各要求条件に着目してそれらを処理可能にするためのオブジェクトを定義し、各オブジェクトごとに、高度で複雑なサービス制御を行うためのオペレーションを定義する。

このようなオブジェクト指向に基づいた呼モデルでは、各オブジェクトごとにオブジェクトの属性やオペレーションを定義することによって従来より迅速かつ高度なサービスの提供が可能になる。

以下、各オブジェクトとそれらの関係について簡

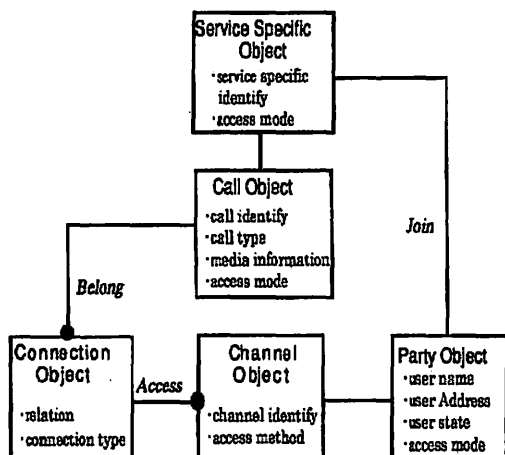


図3. オブジェクト指向呼モデル

単に説明する。

(1)パーティーオブジェクト

これは、ユニークに識別されるネットワークユーザや、サービスに対して情報を提供する情報提供者などを表すオブジェクトである。従って、ユーザを識別するためのユーザ名及びユーザアドレスとサービスにアクセスするために要求されるアクセスモード(A_p)などの属性を持つ。

このオブジェクトはチャネル(Channel)オブジェクトを介してコネクシオンにアクセスする。アクセス方法は、通常3つの方法(In, Out, InOut)で行われる。ここで、Inはパーティーに情報を伝送する能力、Outはパーティーから情報を受け取る能力、InOutは両方の能力を意味する。

一つのパーティーは、サービス参照識別子によって複数のサービスに参加(Join)することができる。また、各パーティーは複数のチャネルを持って、0個以上のチャネルが一つのコネクシオンにアクセスすることができる。さらに、このオブジェクトは、サービス実行中にユーザの追加や退席を要求することができるオペレーションを持つ。

(2)サービス固有オブジェクト

ユーザの要求によってサービスを起動し、複数の異なるサービスを制御するためのオブジェクトである。また、コネクシオンに依存しないセッションを制御するオブジェクトである。このオブジェクトは、サービスを区別するための識別子やそのサービス固有のアクセスモードを定義する属性を持つ。ま

た、ユーザの要求によって、他のサービスを起動するオペレーションを持っている。

例えば、ビデオ・オン・デマンドサービスを制御するビデオサーバなどがこのオブジェクトに相当する。

(3)呼オブジェクト

ユーザからの要求に応じたサービス提供者との間の情報の交換を行うために、ユーザ間やユーザ・サーバ間のネゴシエーションを調整したり、ユーザやサーバの状態および適合性などをチェックする制御オブジェクトである。

これは、複数の呼を区別するための呼識別子(call identity)、サービスの処理中他のユーザがその呼に付け加えられるかどうかを示す呼タイプ(call type)、呼の初期者によって表されるアクセスモード(A_p)などの属性を持つ。また、呼の設定や解除だけでなく、ユーザの要求によって呼制御を行うことができるオペレーションを持つ。これらの詳細については、次の節で述べる。

(4)コネクシオンオブジェクト

これは、パーティー間でユーザ情報をやりとりするための抽象的通信媒体として、コネクシオンの設定や解除、変更などコネクシオン制御を行うオブジェクトである。

これは、コネクシオンタイプや複数のコネクシオン間の関係を指定することができる属性を持つ。また、ユーザの要求によりコネクシオンの追加/削除などを行うことができるオペレーションを持つ。

(5)チャネルオブジェクト

パーティーオブジェクトがコネクシオンにアクセスするためのオブジェクトであり、一つのパーティーは複数のチャネルを持つ。各チャネルは、チャネル識別子によってユニークに識別される。また、サービスごとに違うチャネルを使うことによって、各サービスに対するサービス品質(QoS)を柔軟に要求することができる。

このオブジェクトには、コネクシオンにアクセスするためのVCI/VPIリソースを確保するオペレーションだけではなく、アクティブチャネルの変更などを行うことができるオペレーションを持つ。

4.2 呼制御オペレーション

以上で提案したオブジェクト指向に基づいた呼モデルでは、あるオブジェクトの具体化が、他のオ

プロジェクトの具体化に影響する、オブジェクト間の依存関係が存在する。

ここでは、ユーザが呼の動的振る舞いを制御可能にする高度なサービス制御を行うための各オブジェクトにおける制御オペレーションを定義する。このようなオペレーションを用いることによって、サービス実行途中新しいユーザの追加や退席、メディアの追加や削除などを実現することができる。従って、ユーザにおけるサービス要求情報の表現は一定ではない。ユーザが呼の動的振る舞いを制御可能にするオペレーションを表1に示す。

以下では、このようなオペレーションの中で、特に呼制御と関連するオペレーションについて具体的に定義する。

(1) 呼の設定・解除(Call Establishment/Release)

ユーザが新しいサービスを要求する時、以下のオペレーションを用いる。呼の初期者(オーナー)は、呼の参加者のリストとその参加者に対して許されるアクセスモード(A_i)を定義する。

- call establish [call: (call_id, call_type, sub-list), call_user_list: [user_addr, permitted_access_mode]] → callObject
- Join [call_id, desired_access_modes] (a)
- Leave [call_id, agentID] (b)
- Terminate [call_id] (c)

さらに、初期者によって呼ばれた参加者が呼の招待を受け入れる時は、以下のように呼に対して呼識別子と自分自身のアクセスモードを表すインタフェースを用いて参加する(a)。

また、サービス実行途中ユーザはいつでも呼から離れることができる。その時は、自分が離れる呼識別子と自分の場所を示すユーザアイデンティティを表す(b)。

一時的な呼(C_p)は、参加者がいなくなると終了する。さらに一時的呼および永久的呼は、呼の制御者によっていつでも終了される。このようなオペレーションに対するインタフェースは、上記(c)のようになる。

(2) 呼の合同(Merging Call)

以下のような Merge オペレーションを使うことによって、複数の呼を一つの呼に合同することができる。この時重要なのは、各要素の呼から合同する呼の状態を決めることである。特に、一人のユーザが異なるアクセス権を持って一つ以上の呼で現れる

表1. 呼制御オペレーション

Object	operation	内容
Service specific Object	Invoke implicit-services	ユーザの要求によって他のサービスを起動
	Invoke explicit-services	ユーザの要求によってサービスを起動
Call Object	Call Establish	ユーザからのサービスの開始要求による呼の確立
	Call Release	ユーザからのサービスの終了要求による呼の終了
	Merging Call	一つ以上の呼を一つの呼に合同
	Splitting Call	一つの呼を2つ以上に分離
	Copying Call	あるサービスの呼を他のサービスに複製
	Moving Call	あるサービスにおける呼を他のサービスに移動
	Change Controller	呼制御者を他のパーティのユーザに指名
Party Object	Add User	他ユーザの追加
	Remove User	サービスメンバの退席
Connection Object	Setup Connection	呼の確立のためコネクション設定
	Release Connection	コネクション解除
	Add Connection	コネクションの追加
	Delete Connection	コネクションの削除
	Reserve Channel	VCI/VPIリソースを確保
Channel Object	Commit Channel	チャネルの初期化
	Modify Channel	アクティブチャネルの変更
	Release Channel	アクティブチャネルの解除

ときは、注意が必要である。

このオペレーションは、すべての要素の呼(すなわち、C₁, C₂, … C_n)を終了させると実行可能になる。

- Merge [C₁, C₂, … C_n] → C_{mg}

(3) 呼の分離(Splitting Call)

複数の呼が一つの呼に合同するのと同様に、一つの呼はユーザの要求によって2つ以上の呼に分離される。ただし、この時分離された呼のそれぞれの参加者は、元の呼の参加者の部分集合になる。

- Split [call_id, list1Of [user, permittedAccessModes], list2Of [user, permittedAccessModes]] → [call1, call2]

(4) 呼の複製や移動(Copying and Moving Call)

呼も一般のファイルと同様に、複製(copy)や移動(move)が可能である。

ここで、移動するオペレーション Move はソース

呼を終了させるのに対して、複製オペレーション Copy はソース呼を終了させないのが特徴である。

- Copy [sourceCall, destCall]
- Move [sourceCall, destCall]

(5) その他(Micellaneous Operations)

以上で述べたオペレーション以外にも、(1)呼参加者のアクセス権の変更、(2)呼の制御者として他の参加者を指名することができるオペレーションなどを以下のように定義する。

すなわち、サービス実行途中アクセス権の変更を要求したいときは、いま参加している呼の識別子と自分の場所を表すアイデンティティと新しいアクセスモードを表してオペレーションを定義することができる。また、呼の制御権を他の参加者に譲りたいときは、その呼の識別子と新しい制御者になる参加者の場所を表すアイデンティティを用いて定義する。

- Change Access Attributes[call_id, userID, new Access Rights]
- Change Controller [call_id, new ControllerID]

5.まとめ

高速通信網の構築に伴い、その上で期待されているマルチメディア・マルチユーザアプリケーションを実現可能にするための高度な呼モデルの設計について考察した。特にビデオ・オン・デマンドや遠隔授業サービスなどに必要なユーザと情報提供者間におけるコネクションを制御する第三者呼制御を可能にするための呼モデルを提案した。

このモデルは、より多くのアプリケーションを分散処理環境の上で効率的に実行させるために有効なオブジェクト指向の概念に基づいている。すなわち、マルチメディア呼モデルに必要なとされるマルチコネクションおよびマルチパーティーなどの制御を行うものをそれぞれ一つのオブジェクトとして定義し、各オブジェクトごとにオブジェクトの属性や実行可能な制御オペレーションを定義することによって、より高度なサービス制御が可能になる。さらに、ユーザが呼の動的振る舞いを制御可能とし、呼の全体の状態を明確に示すことができるようにユーザのサービス要求記述法およびそれに基づいた呼の状態モデルを定義した。

今後は、このような呼モデルの上でより複雑で高度なサービス制御を行うためのサービス制御方式について検討する予定である。そこで、まず、高度

なサービス制御を実現可能にするために、より形式的なサービス制御記述法について検討する予定である。このようなサービス制御記述法は、ユーザの要求に基づいた動的なサービス制御要素をより正確に記述することができるし、さらにサービス競合などを検証しやすくなると期待される。

参考文献

- [1]M.Appeldorn, et al. : " TMN+IN=ITINA", IEEE Commun. Magazine'93, no.3, pp.78-85, March. 1993.
- [2]M.Arango, et al. : " Touring Machine System", Commun. ACM, vol.36, no.1, pp.68-77, Jan. 1993.
- [3]ITU-T SG XI : " Draft Recommendation Q.121x ", March. 1995.
- [4]ITU-T SG II TD : "Broadband Capability Set 2 Signalling requirements" (1993-12).
- [5]ITU-T L374 : "Framework Recommendation on Network Capabilities to support Multimedia Services" (1993-8).
- [6]G.J.Heijenck et al : "Communication Systems Supporting Multimedia Multi-user Applications", IEEE Network, pp.34-44, Jan./ Feb. 1994.
- [7]L.Henckel : " Multiper Transport Services for Multimedia Applications ", The 5th IFIP Conference on High Performance Networking, Grenoble, June. 1994.
- [8]S.Minzer : " A Signaling Protocol for Complex Multimedia Services ", IEEE J.Select.Areas in Communications, vol.9, no.9, pp 1383-94, Dec. 1991.
- [9]S.E.Minzer and D.R.Spears : " New directions in Signaling for Broadband ISDN ", IEEE Communication Magazine '89, vol.27, no.2, pp 6-14, Feb. 1989.
- [10]朴, 岡崎 : "高度なマルチメディア・マルチユーザアプリケーション実現のための呼モデルの設計" 信学技報 SSE 95-31, 1995.
- [11]S.C.Park et al : " A Call Model for Multimedia Multi-user Communication Platform", The 2nd International Conference on Network Protocols, pp.305-312, Tokyo,Oct. 1995.
- [12]P.V.Rangan, H.M. Vin : "Multimedia Conferencing as a Universal Paradigm for Collaboration", In Multimedia Systems, Applications and Interaction, Chapter 13, Las Kjelldsahl(editor), Springer-Verlag, Germany, April. 1994.
- [13]J.B.William, et al. : " The TINA Initiative", IEEE Commun. Magazine'93, no.3, pp.70-76, March. 1993.
- [14]山崎 : "マルチメディアネットワークのためのサービス制御方式" 信学技報 SSE 94-83, 1994.