

QoS 保証を考慮したやわらかいマルチメディアシステム

橋本 浩 二† 野村 尚 央††
柴田 義 孝† 白鳥 則 郎†††

地理的に遠く離れた複数の利用者が、ビデオや音声によってリアルタイムに通信を行い、同時にネットワーク上に分散して存在するマルチメディア情報を参照できるような分散マルチメディアシステムでは、計算機資源やネットワーク資源が動的/静的に変化する場合でも、利用者が要求するサービスの質 (QoS) を適切に保証する必要がある。本稿では、計算機やネットワークなどの資源環境や利用状況の変化がともなう場合でも、利用者の要求する QoS を考慮し、利用者に安定したマルチメディア通信を提供するためのシステムとして、エージェント指向アーキテクチャを用いたやわらかいマルチメディアシステムを提案する。そして、その応用例として、蓄積型およびリアルタイム型のメディアサービスを同時に利用でき、状況の変化に対応でき、かつ安定したサービスを利用者に提供する「やわらかいマルチメディア会議システム」の設計と実装を行った。

Design and Implementation of a Flexible Multimedia System with QoS Guarantee Functions

KOJI HASHIMOTO,[†] TAKAO NOMURA,^{††} YOSHITAKA SHIBATA[†]
and NORIO SHIRATORI^{†††}

Distributed multimedia system which not only integrates various multimedia information distributed over computer networks but also provides it to users in accordance with the user's requirement during real time communication must guarantee the user requested quality of services (QoS) even though the computer and network resources change dynamically or statistically. We introduce a flexible multimedia system with QoS guarantee functions, which is based on agent-oriented architecture and provides both real time and stored multimedia information services to users simultaneously even though the multimedia service environment and the resource utilization of computers or networks have been dynamically changed. This paper describes system architecture and its design and implementation of the flexible multimedia teleconference system as one of the application of flexible multimedia system provides both of stored multimedia services and realtime multimedia services to users.

1. はじめに

コンピュータの高性能化やネットワークの高速化、そしてインターネットの普及により、コンピュータネットワーク上でマルチメディア通信を行うシステムが出現している。現在、インターネットでは会議システムやインターネット電話などのアプリケーションが利用可能となったが、これらのアプリケーションが提供するサービスの品質は必ずしも良いとはいえない。また、

コンピュータネットワーク上では様々なアプリケーションが実行されるが、利用者から要求されるサービスの質 (Quality of Service: QoS) はアプリケーション自身や利用者の計算機およびネットワーク環境により異なる。したがって、ネットワーク上でマルチメディアを扱うには、利用できる計算機やネットワーク資源と利用者の QoS への要求を考慮したシステムアーキテクチャ、プロトコルおよびメディア制御機能が必要となる。

現在、RTP や RSVP といったメディアデータ転送時の QoS 保証を考慮したデータ転送プロトコルが存在する。RTP¹⁾ はアプリケーション層フレーミングにより、MPEG, JPEG, H.261 などの様々な符合化のリアルタイムトラフィックに適用できる枠組みを提供している。また、RTCP は RTP の制御プロトコルで

† 岩手県立大学

Iwate Prefectural University

†† データーリンクス株式会社

Data Link Corporation Inc.

††† 東北大学電気通信研究所

Research Institute of Electrical Communication,
Tohoku University

あり、ネットワーク上での遅延や帯域幅の変動を検出し、上位プロトコルに通知する機能を有する。そして、RSVP²⁾はアプリケーションがネットワーク資源を予約するためのプロトコルであり、IPv6のフローラベルと優先度を利用し、送受信者間のトラフィックストリームに対する資源予約の手順を定義している。

一方、Campbell³⁾、Nahrstedt⁴⁾らはATMをベースとしたQoS保証を実現するためのアーキテクチャを示しており、Kadur⁵⁾、Georgiadis⁶⁾らは、遅延やジッタの制御について述べている。

このようにQoS機能を考慮したプロトコルやネットワークアーキテクチャなどの研究はさかに行われているが、利用者から見たエンド間でのQoSを保証できるシステムアーキテクチャや機能については、あまり議論されていない。これは、特にマルチメディア会議システムなどにおいて重要である。たとえば、遠隔地に離れた利用者同士が互いの意思疎通を行うためのリアルタイム型メディアサービスを利用する一方で、ネットワーク上に分散するデータベースに格納されたマルチメディア情報を参照するための蓄積型メディアサービスも利用する場合、システムは蓄積またはリアルタイムメディアを同時に処理しなければならない。また、蓄積されたマルチメディア情報を利用者へ提供するためにはオーディオ、ビデオのような連続メディアとイメージ、グラフィックス、テキストのような非連続メディアを統合して提供する必要がある。この場合もシステムは、メディアごとに異なる処理や制御方法によりエンド間のQoS保証⁷⁾が統合できなければならない。

そのようなシステムを実現するためには、利用者ステーションの処理能力、ネットワーク帯域幅とその利用状況、さらに扱うメディアデータの特性を考慮し、利用者のQoS (Quality of Service) 要求を保証する仕組みが必要である。そのために、アプリケーションからネットワークまで一貫したQoS制御機能と各メディア処理におけるQoS保証機能が必要となる。また、状況変化の程度によっては、これらの制御機能を動的に再構成して対応させることも必要となる。

そこで、本研究では、これらの問題を解決するためにやわらかさの概念⁸⁾に基づいたエージェント指向のマルチメディアシステム (Flexible Multimedia System: FMS) を提案する。

FMSはアプリケーションからネットワークまで一貫したQoS制御を行うために必要な機能をコンポーネント化し、また、計算機やネットワーク資源の状況変化や利用者の要求に従ってこれらのコンポーネントを

能動的に再構成したり管理を行ったりする各種のエージェントによって構成される。これら各種のコンポーネントとエージェントをモジュールとしてリポジトリ化しておくことにより、必要な機能モジュールを動的に構成することが可能となる。システムは利用者のQoS要求に応じて、必要な機能モジュールの組織化を行うことができる。また、マルチメディア通信を行っている最中に計算機やネットワーク資源の状況変化に応じて必要な機能モジュールをリポジトリから取得して組み替えを行ったり、不必要になった機能モジュールは削除することが可能となる。これにより、計算機資源の有効利用を図るとともに、動的/静的な状況変化に柔軟に対応し、安定したマルチメディア情報を提供することが可能となる。ここで、静的状況変化とは、利用者が要求するQoSパラメータの確定的な変化(たとえば、8bitカラービデオから24bitフルカラービデオへの変更)をいい、動的状況変化とは、ネットワークトラフィックやCPU占有率などランダムに変動する変化をいう。

以下、2章でエージェント指向に基づくやわらかいマルチメディアシステムを提案する。3章ではその応用例としてのやわらかいマルチメディア会議システム⁹⁾について述べる。また、4章ではマルチメディアを統一的に扱うための抽象化モデルについて述べ、5章ではエージェントおよびコンポーネントの組織化について述べる。そして、6章では現在の実装について述べ、7章で本稿をまとめる。

2. やわらかいマルチメディアシステム

2.1 システム概要

FMSは、従来のマルチメディアシステムにやわらかさの概念⁸⁾を取り入れたシステムであり、下記の機能を有するマルチメディアシステムと定義する。

- (1) 利用者にとって簡単で、使いやすいヒューマンインタフェース機能
- (2) ネットワーク構成や利用者ステーションなど利用環境に応じた資源の予約、交渉機能
- (3) 資源利用状況に応じたQoSパラメータ調整および機能モジュール再構成機能

FMSは、基本的には複数の利用者ステーションとネットワークで構成される。利用者は、利用者ステーションを通して、オーディオ、ビデオ、イメージ、グラフィックス、テキストによりリアルタイム通信を行ったり、利用者が所有する蓄積メディアをプレゼンテーションしたり、複数の利用者間で共有したりすることが可能である。

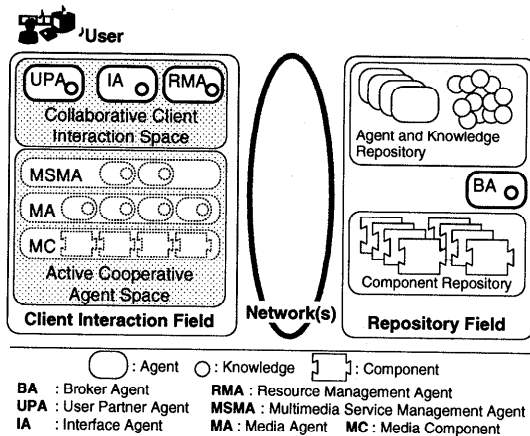


図1 やわらかいマルチメディアシステムアーキテクチャ
Fig. 1 Flexible Multimedia System architecture.

また上述した機能により、FMS は利用者端末における CPU パワーやメモリ量、ネットワークの帯域幅、そしてこれらの利用状況と利用者の QoS 要求といった、動的/静的に変化する計算機資源、ネットワーク資源、利用者要求に対して、利用者にとって満足度の高い安定したマルチメディアサービスを行えるシステムである⁹⁾。そのシステムアーキテクチャを次に述べる。

2.2 システムアーキテクチャ

FMS を実現するための方法として、エージェントベースのアーキテクチャを提案する。FMS は、複数のエージェントとコンポーネントで構成される。図1はそのシステムアーキテクチャである。システムは、Client Interaction Field (CIF) と Repository Field (RF) の2つのフィールドにより構成される。

2.3 Client Interaction Field (CIF)

利用者端末におけるエージェントやコンポーネントが動作するフィールドであり、以下に示す2つの動作空間で構成される。

(1) **Collaborative Client Interaction Space (CCIS)** は、システム常駐型のエージェントと利用者のインタラクションにより構成される動作空間であり、以下に示す3つのエージェントにより構成される。

- **User Partner Agent (UPA)** は、下記 (2) ACAS で述べる MSMA を組織化する。また MSMA を操作することによって、マルチメディアサービスを利用者に提供する。さらに、利用者に対するメッセージを生成する。これは、資源利用状況の変動にともなう QoS 更新が行われる場合、その旨を利用者に通知するためである。
- **Interface Agent (IA)** は、利用者に FMS のインタフェースを提供する。ビデオウィンドウや

GUI 部品を利用者に提供し、利用者からのサービス開始/終了要求や QoS 更新要求などをイベントとして処理する。また、UPA が生成する利用者へのメッセージを再生/表示する。

- **Resource Management Agent (RMA)** は、システムで利用可能な資源を把握し、その利用状況を監視する。また、必要な資源の割当てや使用を終了した資源の解放を行う。

(2) **Active Cooperative Agent Space (ACAS)** は、状況に応じたマルチメディア処理を行うために必要となるエージェントとコンポーネントの動作空間であり、以下に示す2種類のエージェントとコンポーネントの組合せにより構成される。

- **Multimedia Service Management Agent (MSMA)** は、利用者に提供するマルチメディアサービスを特定する複数のエージェントである。マルチメディアサービスごとに異なるメディア間の時間的および空間的關係の調整ならびに管理を行う。また、状況に応じて必要となる Media Agent (MA) を組織化し、操作する。
- **Media Agent (MA)** はメディアごとに存在するエージェントである。必要なメディアコンポーネント (MC) を組織化する。また、以下に示す MC を操作するために必要となる知識を保有する。
- **Media Component (MC)** はメディア処理に必要な機能モジュールである。たとえば、ビデオの提供においては、MPEG, JPEG, H.261 などの圧縮符合化モジュールが存在する。これらのメディアの処理は、MA によって組織化された複数の MC が行う。

2.4 Repository Field (RF)

CIF で使用するエージェント、知識、コンポーネントを格納しておくフィールドであり、下記に示す2つのリポジトリとエージェントにより構成される。

- **Agent and Knowledge Repository (AKR)** マルチメディアサービスを提供するエージェントと知識が格納されるリポジトリである。
- **Component Repository (CR)** マルチメディアサービスの提供に必要な様々なコンポーネントが格納されるリポジトリである。
- **Broker Agent (BA)** RF に格納されているエージェント、知識およびコンポーネントを計算機やネットワーク資源および利用者要求に応じて、CIF へ提供するエージェントである。

表 1 FMS 構成要素
Table 1 FMS constituents.

利用者を含むFMS構成要素				
	利用者	メディアデータ	利用者ステーション	ネットワーク
環境/属性	・使用言語 ・年齢 ・専門分野	・蓄積/リアルタイム ・連続/非連続 ・単一/複合 ・データフォーマット	・オペレーティングシステム ・メモリ量 ・CPUパワー ・ディスプレイの出力分解能力 ・入出力デバイス	・構成 ・アクセス方法 ・帯域幅 ・プロトコル
状況変化の種別	・利用するメディアデータ ・QoS要求	・データ量 ・メディアの質	・メモリ、CPUの使用状況 ・プロセス数	・トラフィック
必要な機能	・システムインターフェース	・メディア内同期 ・メディア間同期 ・フォーマット変換 ・メディアデータフロー	・メモリ、CPU使用状況監視 ・プロセス数監視 ・メモリ、CPU資源確保/適合	・トラフィック監視 ・帯域幅確保/適合
必要な知識	・利用者毎の属性 ・マルチメディアサービス種別と構成メディア	・メディアデータ属性 ・状況の変化に応じたメディアデータ処理法	・利用者端末属性 ・メモリ、CPU監視方法	・ネットワーク属性 ・トラフィック監視方法
知識利用エージェント	UPA	MSMA, MA	RMA	RMA

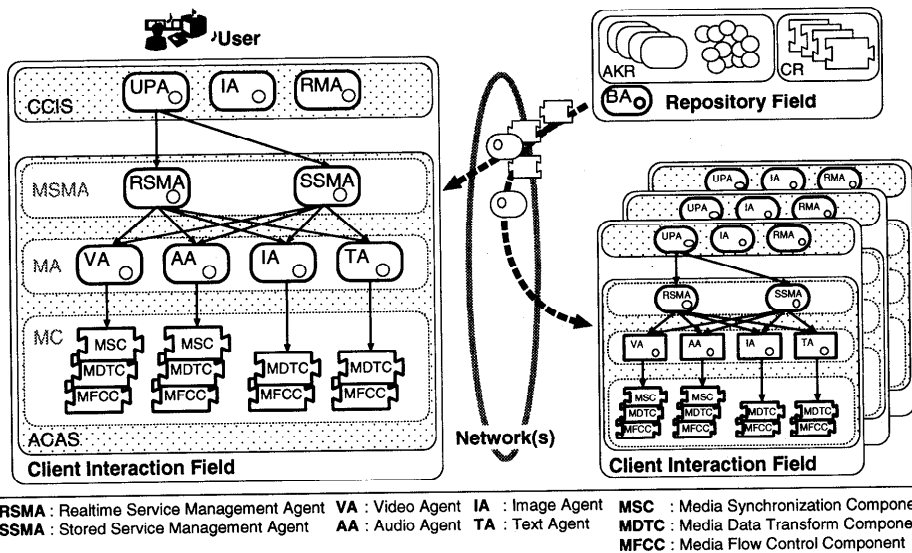


図 2 やわらかいマルチメディア会議システムの構成
Fig. 2 Configuration of Flexible Multimedia Teleconferencing System.

2.5 FMS 構成要素と機能および知識

マルチメディアシステムは、様々なメディアサービスを統合して利用者へ提供できなければならない。また、利用者の要求するマルチメディアサービスを必要に応じて実現するためには、表 1 に示すように、FMS 構成要素の属性や状況の変化に応じて必要となる機能と知識を用いてマルチメディア処理を行う必要がある。

3. やわらかいマルチメディア会議システム

FMS を基盤とした具体的なシステムとして、マルチメディア会議システムを図 2 に示す。

マルチメディア会議では、テレビ電話やホワイト

ボードなどのリアルタイムメディアの利用と、利用者ステーション内のデータベースにすでに用意されているオーディオ、ビデオ、テキスト、グラフィックスによって構成される蓄積型メディアの資料を同時に利用することが想定される。そこで、マルチメディア会議システムは、図 2 に示すとおり、リアルタイムメディアサービスと蓄積メディアサービスを管理する 2 つの MSMA により実現される。

(1) Realtime Service Management Agent

(RSMA)：リアルタイムメディアサービスを利用者に提供するための MA を組織化し、その制御と管理を行う。たとえば、リアルタイムオーディオ・ビデオサー

ビスを利用して、利用者同士が会話をする際は、本質的にビデオの画質やなめらかさよりオーディオデータの質が重要となる場合が多い。もし、環境属性や状況の変化によりオーディオデータに途切れが生じ、会話が損なわれる場合、RSMA は途切れをなくすために Video Agent に対してビデオデータのフレームレートを下げたりフレームサイズを小さくするよう指示する。これにより、単位時間あたりのビデオデータ転送量をおさえることにより、オーディオデータの途切れを抑制する。

(2) Stored Service Management Agent

(SSMA)：蓄積メディアサービスを利用者に提供するための MA を組織化し、MA の制御と管理を行う。たとえば、蓄積型オーディオ・ビデオサービスにおいて、メディアに対するインタラクティブな操作性よりメディアの再生を優先する場合、メディアデータをキャッシュしてビデオデータのなめらかさを維持する必要がある。また、VOD サービスのように厳格なメディア間同期が必要な場合もある。SSMA は、これらの制御メッセージを MA に対して送信する。

上述したリアルタイムおよび蓄積型サービスにおけるメディアの優先順位や制御方法は固定的なものではなく、その制御方針は利用者の QoS 要求によって決定する。利用者からの QoS 要求もマルチメディアサービスの種類によって異なり、どのような QoS 制御機能を選択したりパラメータを設定できるかについては RSMA や SSMA が知識として保有する。

RSMA や SSMA により組織化された MA は、そのメディア処理に必要な MC を組織化し、MC に関する知識と、それらをどのように操作するかといった知識を AKR から取得する。そして、MA は状況に応じて MC を動的に起動し、環境属性や状況の変化に応じたメディア処理を行う。図 2 において CIF 内の各エージェントとコンポーネント間の矢印の方向は、組織化が進む方向を示している。また、Repository Field から Client Interaction Field への矢印は、必要とされるエージェント、知識、コンポーネントが動的に配置されることを示している。

3.1 メディアコンポーネント

マルチメディアサービスを行うために必要となるメディア処理には、大別して以下に述べる 3 つの機能が必要であり、メディアコンポーネントは、これらの機能を実現する。

- **Media Synchronization Component**

(MSC)：メディア内およびメディア間同期処理¹⁰⁾を行う。

- **Media Data Transform Component**

(MDTC)：JPEG, MPEG1/2¹¹⁾などの圧縮/展開や、画像データのカラーフォーマット、オーディオデータの変調方式といったデータ変換処理を行う。

- **Media Flow Control Component**

(MFCC)：メディアのレート制御やパケット紛失の調整を行うため、可変ビットレート転送やパケット間隔調整^{12)~15)}を行う。

メディアコンポーネントは、メディアデータ生成側と利用側でそれぞれ必要に応じて動的に組織化される。本稿では、メディア処理に必要な複数のメディアコンポーネントの集合をメディアオブジェクトと定義する。次の章では、そのモデルについて述べる。

4. マルチメディア抽象化モデル

FMS では、環境と状況の変化に応じて動的にエージェントとコンポーネントを組織すると述べた。しかしながら、エージェントやコンポーネントは、各々別の機能を有しており、操作方法も異なる。操作方法が異なる場合、エージェントがどのようにエージェントやコンポーネントを操作するかという問題が生じる。そこで、メディア処理の特性を隠蔽し、エージェントおよびコンポーネント操作の統一化を実現するために下記のモデルを導入する。

- **Media Object (MO)**

メディアごとに異なる特性を隠蔽した抽象化モデル。

- **Presentation Object (PO)**

リアルタイムメディア、蓄積メディアなどマルチメディアサービスを構成するメディアの特性が同じ MO をまとめた抽象化モデル。

- **Multimedia Service Object (MSO)**

利用者に提供されるマルチメディアサービス全体の抽象化モデル。

これらのモデルを考慮してシステムを実現することにより、サービスに対する操作方法が、利用者からメディアエージェントまで統一的に決定される。すなわち、どのようなメディアエージェントやコンポーネントが組織化されていても「開始」、「一時停止」などのインタラクティブな操作や、「なめらかさ優先」、「画像サイズ優先」など利用者からの QoS 要求⁷⁾に対し、適切な制御を行うことが可能となる。また、システム動作中でも各メディアエージェントやコンポーネントの追加や削除が容易なものとなる。

以下、マルチメディア抽象化モデルである MO, PO,

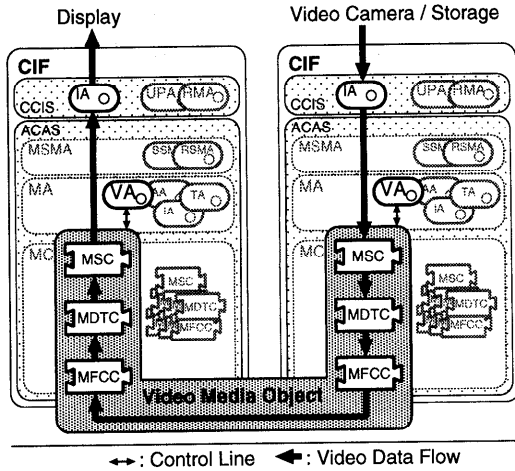


図3 メディアオブジェクトの例
Fig. 3 Example of Media Object.

MSO とエージェントおよびコンポーネントである MSMA, MA, MC の関係についてその詳細を述べる。

4.1 メディア・オブジェクト

メディア・オブジェクト (MO) は、オーディオ、ビデオデータなど特性の異なるメディアデータとその処理を隠蔽し、MA に対して統一した操作方法を提供するものである。

図3で示すように、MO は、1つのメディアデータ (MD) に対し、MD の生成側と利用側各々で処理を行う複数のメディアコンポーネント (MC) で構成される。その定義を以下に示す。

$$MO = \left\{ \sum_{i=0}^n MC_i, MD \right\}$$

- n : メディアコンポーネントの数
- MC_i : i 番目のメディアコンポーネント
- MD : Media Data

利用者へメディアデータを提供するために、MD の生成側と利用側両方で、MA は必要となる複数の MC を組織化する。それらの MC を MO として1つの処理単位と見なし、MO が統一した操作方法を MA に提供することにより、MA は組織化した MC に依存せずにメディア処理の制御と監視を行うことが可能となる。

また、たとえば、同じリアルタイムビデオサービスを行う場合でも利用者ステーションの環境や資源利用状況により MA が組織化する MC は異なり、MD の生成側と利用側における MC の構成によっては実効フレームレートやフレームサイズ、許容エンド間遅延な

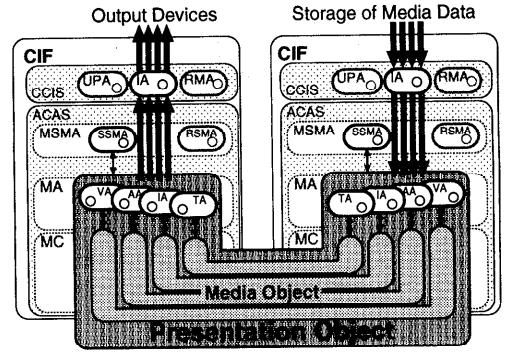


図4 プレゼンテーション・オブジェクトの例
Fig. 4 Example of Presentation Object.

どの QoS がどの程度保証できるかが異なる。MO を導入することにより、MA はメディアデータを処理する MC の組合せによって異なる QoS を把握することが容易になり、MA は環境や資源利用状況に応じて、利用者の QoS 要求を満たすためにより適切な MC の組織化が可能となる。

4.2 プレゼンテーション・オブジェクト

プレゼンテーション・オブジェクト (PO) は、リアルタイムメディア、蓄積メディアなどマルチメディアサービスを構成する複数のメディアの特性が同じ MO を統合し、MSMA に対して統一した操作方法を提供するものである。

図4に示すように、PO では MSMA によってそのインタフェースが生成される。インタフェースは統合されるメディアの特性、表現形式などの知識から、IS-A (Kind-Of) 関係、HAS-A (Part-Of) 関係により生成される。これを Presentation Object Interface (POI) と呼ぶ。また、MSMA は IA に対して POI の生成要求を行う。生成される POI は PO を構成する各 MO に対する操作方法を統合したものである。PO を以下のように定義する。

$$PO = \left\{ \sum_{i=0}^n MO_i, POI \right\}$$

- n : メディアオブジェクトの数
- MO_i : i 番目のメディアオブジェクト
- POI : Presentation Object Interface

PO を定義することにより、MSMA は PO 内部の MO の構成によらず、PO におけるメディア処理の監視や制御を統一的に行うことが容易になる。また、リアルタイムメディアや蓄積メディアを処理する複数の MO を PO として各々統合し、POI を生成することにより、サービス利用者に対して PO 内部の MO が

処理するメディアの特性を考慮したインタフェースを提供することが可能となる。たとえば、蓄積型の複数のメディアを統合して利用者へ提供する場合、「開始」「一時停止」などのインタラクションが可能であるが、リアルタイム型のメディアを利用者へ提供する場合、「一時停止」や「巻き戻し」「早送り」といったインタラクションは無効にできる。

4.3 マルチメディア・サービス・オブジェクト

マルチメディア・サービス・オブジェクト (MSO) は、利用者へ提供されるマルチメディアサービス全体の抽象化モデルである。これを図 5 に示す。

システム常駐型のエージェントは、MSO と連携して環境や状況の変化に応じたマルチメディアサービスを実現する。ここで、MSO を以下のように定義する。

$$MSO = \left\{ \sum_{i=0}^n PO_i \right\}$$

n : プレゼンテーションオブジェクトの数

PO_i : i 番目のプレゼンテーションオブジェクト

MSO を定義することにより、UPA は PO の構成に依存せずにマルチメディアサービスの監視や制御を行うことが可能となる。たとえば、資源利用状況の変化にともなう QoS 更新が行われる場合、UPA は利用者に対してその旨を通知するメッセージを生成するが、UPA がメッセージ生成のために必要な情報を取得する際、MSO 内部で PO がどのように構成されているかに依存しないインタフェースを利用することが可能となる。

FMS は、マルチメディアサービスを利用者へ提供するために、エージェントやコンポーネントをメディアデータの生成側と利用側で各々動的に組織化する。マルチメディア抽象化モデルは、動的に組織化された複数のエージェントやコンポーネントを 3 つのレベル

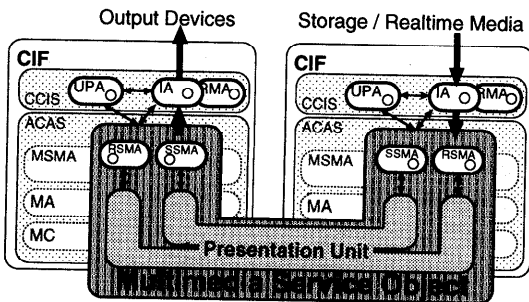


図 5 マルチメディア・サービス・オブジェクトの例
Fig. 5 Example of Multimedia Service Object.

でモデル化することにより、環境や資源利用状況の変化によって構成が変わるエージェントやコンポーネントに対し、メディア処理の監視や制御の統一化を実現する。

5. 組織化

FMS では、マルチメディアサービスに必要なエージェントとコンポーネントを動的に組織化してマルチメディアサービスを利用者へ提供する。FMS は、利用者からマルチメディアサービス要求を受けると、図 6 に示すとおり、3 つの組織化フェーズを経てマルチメディアサービスを開始する。また、利用者からの QoS 更新要求や資源利用状況の変化にともないマルチメディアサービスで利用するメディアの追加や削除が必要な場合、FMS は、MA や MC の再組織化を行う。

組織化および再組織化時に、エージェントは組織形成プロトコルを利用する。表 2 は組織形成プロトコルで利用するメッセージの種類を示している。

図 6 における各組織化フェーズにおいて、エージェントやコンポーネントがをどのように組織化されるかをサービス開始時とサービス中に分け、以下に述べる。

5.1 サービス開始時の組織化

利用者ステーションの性能やネットワーク構成などマルチメディアサービス利用環境と、利用者のサービス要求に応じた組織化を行う。

FMS は、利用者からマルチメディアサービス要求を

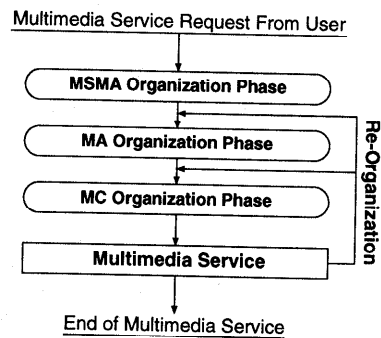


図 6 組織化フェーズ

Fig. 6 Organization phase.

表 2 組織形成プロトコルメッセージ

Table 2 Message type of organization protocol.

Task	組織化タスク通知
Bid	入札通知
Award	落札通知
Termination	終了通知
Request	情報要求
Report	情報通知

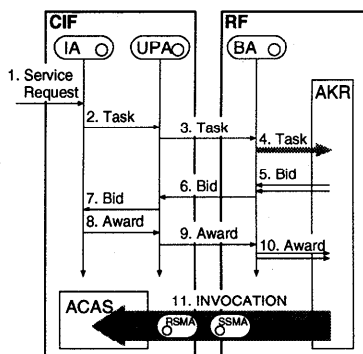


図7 MSMAの組織化

Fig. 7 Organization of MSMA.

受けると、最初に MSMA を組織化する。そのフローを図 7 に示す。図 7 は、テレビ会議のサービス開始要求に対し、リアルタイム型と蓄積型のサービスを統合する RSMA と SSMA が起動されるまでを示している。

まず、利用者からマルチメディアサービス要求を IA が受信すると、IA は UPA に対して *Task* メッセージを送信する。UPA は BA に対して *Task* メッセージを送信し、BA は AKR 内の各エージェントに *Task* メッセージをブロードキャストする。AKR 内の各エージェントは、自分自身が ACAS 構成エージェントになるべきかを判断し、*Bid* メッセージを送信することで入札を行う。BA, UPA, IA は複数のエージェントからの入札に対し、どのエージェントを ACAS 内に組織化するかを判断し、適切なエージェントに対して *Award* メッセージを送信して落札を行う。*Award* メッセージを受信した AKR 内のエージェントは、AKR から ACAS へ移動し、MA の組織化を行う。

図 8 a) は、MSMA の 1 つである RSMA が、VA (Video Agent), AA (Audio Agent), IA (Image Agent), TA (Text Agent) を組織化する例を示している。まず、RSMA は BA に対して *Task* メッセージを送信する。BA は *Task* メッセージを AKR にブロードキャストし、各 MA からの入札を待つ。そして、各 MA からの入札に対し、BA と RSMA は必要な MA を落札する。落札された MA は AKR から ACAS に移動し、各 MA はメディアを処理するために必要となる MC を組織化する。

図 8 b) は、VA が MC を組織化する例を示している。VA は RMA から資源利用状況を取得し、BA に対し *Task* を送信する。BA は Component Repository (CR) 内に存在する複数のコンポーネントから、入札すべきコンポーネントを選択し、VA に対して入札す

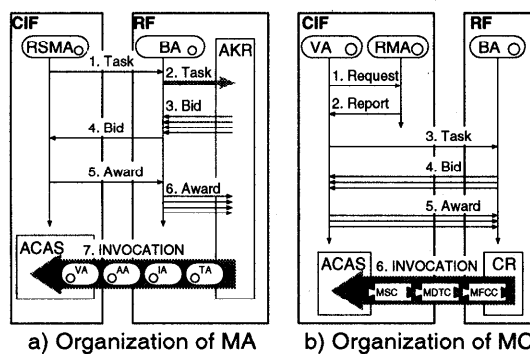


図8 MAとMCの組織化

Fig. 8 Organizations of MA and MC.

る。VA は必要なコンポーネントに対する落札を BA に対して行う。最後に CR から ACAS へコンポーネントが配置され、メディアの処理が可能となる。

5.2 サービス中の再組織化

動的に変化する資源利用状況や、利用者からの QoS 更新要求に応じて、マルチメディアサービスで利用するメディアの追加や削除が必要な場合、FMS はエージェントとコンポーネントの再組織化を行う。

メディアコンポーネントにより構成される各メディアオブジェクトは、動的な資源利用状況の変化に対応する。たとえば、図 3 に示したビデオメディアオブジェクトは、動的な負荷変動に応じてフレーム間引きやパケット間隔調整を行い、ビデオの時間的制約を保証する¹⁵⁾。しかしながら、メディアオブジェクトが対応できるのは、オブジェクト内部で発生する事象に対するものであり、サービス全体の質やメディアの優先順位を考慮すると、動的な再組織化が必要となる。

再組織化には、3つの場合が存在する。

- (1) 利用者に提供するメディアを追加する場合：図 8 に示す組織化プロトコルを利用し、エージェントやコンポーネントを新たに組織化する。
- (2) 利用者に提供するメディアを削除する場合：削除すべきエージェントを組織化したエージェントが、削除するエージェントに *Termination* メッセージを送信する。*Termination* メッセージを受信したエージェントは、利用した知識を AKR へ戻し、使用していた計算機またはネットワーク資源を解放する。コンポーネントを削除する場合も同様に行われる。
- (3) エージェントやコンポーネントを交換する場合：図 8 に示す組織化プロトコルを利用してエージェントやコンポーネントを新たに組織化し、その後、交換すべきエージェントやコンポーネントを削除する。

5.3 知識の生成と利用

FMS は、利用環境の属性や資源利用状況に応じてどのようなマルチメディアサービスを利用者に提供可能かを知識を基に決定する。

マルチメディアサービスを利用者に提供するためには、特に、各 MC がメディア処理を行う際に必要となる計算機およびネットワーク資源を明確にする必要がある。そして、MA は MC がメディア処理をするために必要な資源とそのサービスの質の関係を知識として蓄積し、利用する。

ACAS 内に組織化されマルチメディアサービスを行うエージェントは、組織化される時に AKR 内の最新の知識を取得する。そして、サービス中は MC のメディア処理状況と RMA から得られる資源利用状況を基にメディア処理と資源利用状況の関係を知識として蓄積する。蓄積した知識は、マルチメディアサービス終了時またはエージェントや MC の再組織化時に RF 内の AKR へ格納される。格納された知識は、別の FMS において利用可能となる。

MA が MC を操作するために利用する知識の基本的なパラメータを以下に示す。

(1) 環境パラメータ：

環境パラメータは、利用者のマルチメディア環境を特定するパラメータである。

- OS 種別：実時間処理をするために CPO 資源を確保できるか、適合するかが決定する。
- ネットワーク構成と利用可能なプロトコル：帯域幅を確保できるか、適合するかが決定する。
- CPU 種別と処理速度：MC の基本的な性能を決定する。
- デバイスの特性：ディスプレイの解像度、音声デバイスなどを示し利用可能なメディアが決定される。

(2) 資源利用状況パラメータ：

資源利用状況パラメータは、動的に変化する資源利用状況を把握するパラメータである。ネットワークトラフィック、CPU 占有率、メモリ使用量、ロードアベレージ、ページングやスワップ回数などが資源利用状況パラメータとなる。

(3) メディア QoS パラメータ：

上述した環境および資源利用状況パラメータに対して、MC がどの程度の性能を発揮するかを示すパラメータである。ビデオデータの同期、フォーマット変換処理においては、フレームレート、フレームサイズ、色数などがメディア QoS パラメータとなる。また、ネットワーク転送においては、時間的制約とパケット紛失

率を制御するパケット間隔もパラメータとなる。

これらのパラメータを知識として扱うことにより、利用者のサービス要求に応じて、FMS は多様な環境と資源利用状況を考慮した柔軟なマルチメディアサービスを実現することが可能となる。

6. プロトタイプシステム

現在、FMS の有効性を評価するために、FMS プラットフォームを構築し、その上でやわらかいマルチメディア会議システムの実装を行っている。実装は Sun WorkStation/Solaris 2.5.2 上で行い、開発言語には C++ を用いている。

CIF, RF はそれぞれ UNIX における 1 つのプロセスで実現し、エージェントやコンポーネントは複数の POSIX 互換スレッドで実現した。

また、プロセス内のスレッド間通信には共通のメモリを使用し、ワークステーション間のスレッド間通信には TCP/IP を用いた。これにより、エージェント間通信を実現している。

RMA が行うネットワークトラフィックの測定には snmp を利用し、利用者ステーションの資源利用状況監視には rstat ライブラリを利用している。rstat ライブラリを利用することにより、CPU 占有率、ロードアベレージ、ページングやスワップなどの情報を取得することができる。

メディアオブジェクトなどの抽象化モデルは、そのインタフェースの部分を C++ の仮想関数を利用して実現した。また、コンポーネントはポリモニックオブジェクトとして実装することによりエージェントからの動的な起動を可能としている。

エージェントとコンポーネントに関するオブジェクトの階層図を図 9 に示す。ThreadObject クラスはスレッドとしての処理機能をオブジェクト化したクラスである。そして、ThreadObject クラスを継承している AgentCore クラスと ComponentBase クラスがエージェントとコンポーネントのベースとなるクラスである。すべてのエージェントとコンポーネントのクラスは ThreadObject クラスを継承しており、スレッド単位の動作を统一的に扱うことが可能となっている。

現在までに実装できているやわらかいマルチメディア会議システムのインタフェースの例を図 10 に示す。これは、複数の利用者がテレビ会議を行いながら、同時に、分散する DB のマルチメディア情報をインタラクティブに利用している例である。

プロトタイプでは、エージェントやコンポーネントを必要に応じて起動することが可能となり、利用者の

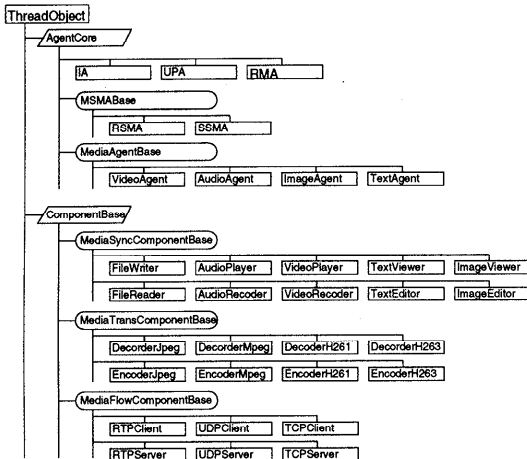


図9 メディアコンポーネントの実装におけるオブジェクト階層
Fig. 9 Object hierarchy for realized media component.



図10 プロトタイプインタフェース例
Fig. 10 Example of prototyped interface.

環境や資源の利用状況と利用者のサービス要求に応じてFMSを組織化することが可能となった。しかしながら、FMSを組織化するエージェントやコンポーネントのリポジトリは、現在のところCIFと同一の計算機上に実現されているため、多様な環境における組織化と知識の利用に関しては、さらに実装を進め、評価を行う必要がある。

また、RMAは資源利用状況を監視するためにrstatライブラリを利用してCPU占有率とロードアベレージを監視している。しかし、CPU占有率は負荷変動に敏感でその値の変動が激しく、ロードアベレージ負荷変動に対してその値の変動が緩やかである。現在、これらの値をいかに組み合わせて利用するかを検討している。また、その他有益なパラメータとしてページングやスワップ数があるが、これらのパラメータ利用する方法も検討している。

7. まとめ

本稿では、動的/静的な状況変化に柔軟に対応し、利用者に安定したマルチメディア情報を提供するFMSの提案をし、マルチメディアを統一的に処理するための抽象化モデルについて述べた。また、FMSにおけるエージェントとコンポーネントの組織化について述べ、現在の実装について述べた。

FMSは利用者の環境や資源の利用状況と利用者のサービス要求に応じてエージェントやコンポーネントを組織化することが可能であるが、組織化を行うために必要となる知識は十分に確定しておらず、資源利用状況とメディア処理の関係は必ずしも明確になっていない。これらについては、プロトタイプ機能および性能評価を行うことにより明確にしていく予定である。

プロトタイプは、今後JAVAとC++を利用して行う予定である。エージェントはJAVAで実装する。これにより、エージェントは、多様な環境下で同じクラスインスタンスを利用することが可能となる。しかしながら、JAVAはOSやデバイスに依存した処理に向いておらず、実際にメディア処理を行うコンポーネントの実装には引き続きC++を利用する。OSやデバイス種別ごとにコンポーネントを実装し、ダイナミックリンク機能などを利用して動的な組織化、起動を可能とする予定である。

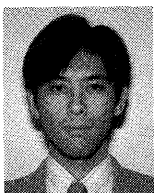
参考文献

- 1) Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications, RFC 1889 (Jan. 1996).
- 2) Zhang, L., Berson, S., Herzog, S. and Jamin, S.: Resource ReSerVation Protocol (RSVP) - Version 1, Functional Specification, Braden, R., (Ed.), RFC 2205 (Sep. 1997).
- 3) Campbell, A., Coulson, G. and Hutchison, D.: A QUALITY OF SERVICE ARCHITECTURE, *ACM Sigcomm Computer Communication Review*, Vol.24, No.2, pp.1-27 (1994).
- 4) Nahrstedt, K. and Smith, J.M.: The QOS Broker, *IEEE MultiMedia*, Vol.2, No.1, pp.53-67 (1995).
- 5) Kadur, S., Golshani, F. and Millard, B.: Delay-jitter control in multimedia applications, *ACM Multimedia Systems*, Vol.4, pp.30-39 (1996).
- 6) Georgiadis, L., Guerin, R., Peris, V. and Rajan, R.: Efficient Support of Delay and Rate Guarantees, *ACM Sigcomm*, Vol.26, No.4

- (1996).
- 7) Hashimoto, K. and Shibata, Y.: Performance Evaluation of End-to-End QoS Using Prototyped VOD System, *Proc. ICOIN-12*, pp.175-178 (1997).
 - 8) Shiratori, N., Sugawara, K., Kinoshita, T. and Chakraborty, G.: Flexible Network: Basic Concepts and Architecture, *IEICE Trans. Communication*, Vol.E77-B, No.11, pp.1287-1294 (1994).
 - 9) 野村尚央, 柴田義孝, 白鳥則郎: やわらかいマルチメディア会議システムの設計と実装, 情報処理学会研究報告 DPS-86-32, pp.183-188 (1998).
 - 10) Shibata, Y., Seta, N. and Shimizu, S.: Media Synchronization Protocols for Packet Audio/Video System on Multimedia Information Networks, *Proc. HICSS-28*, pp.594-601 (1995).
 - 11) Gall, D.L.: MPEG: A Video Compression Standard for Multimedia Applications, *Comm. ACM*, Vol.34, No.4, pp.46-58 (1991).
 - 12) 赤間孝司, 渡辺光輝, 橋本浩二, 柴田義孝: パケットオーディオ・ビデオシステムのための動的なパケット間隔制御, 情報処理学会研究報告 DPS-67-7, pp.37-42 (1994).
 - 13) 知念 正, 柴田義孝: パケット紛失を考慮した連続メディア転送プロトコルの研究, 情報処理ワークショップ論文集, Vol.96, No.1, pp.67-72 (1996).
 - 14) Sato, J., Hashimoto, K. and Shibata, Y.: Dynamic Rate Control for Continuous Media Transmission, *IPSJ SIG-Notes*, DPS-85, Vol.97, pp.85-90 (1997).
 - 15) 橋本浩二, 知念 正, 佐藤 純, 柴田義孝: 圧縮ビデオデータ転送のためのパケットおよびフレームレート制御法, 情報処理学会論文誌, Vol.39, No.2, pp.337-346 (1998).

(平成 10 年 5 月 8 日受付)

(平成 10 年 9 月 7 日採録)



橋本 浩二 (正会員)

1970 年生。1996 年東洋大学大学院工学研究科電気工学専攻博士前期課程を修了。同年 (株) CSK 総合研究所に入社。1998 年より岩手県立大学ソフトウェア情報学部助手。マルチメディアシステムにおけるやわらかさとエンド間 QoS 保証の研究に従事。



野村 尚央 (正会員)

1972 年生。1996 年東洋大学工学部情報工学科を卒業。1998 年同大学大学院工学研究科電気工学専攻博士前期課程を修了。同年, データリンクス (株) 入社。開発研究部勤務。現在, データベースおよびインターネットサービスシステムの研究・開発に従事。



柴田 義孝 (正会員)

1950 年生。1985 年 UCLA コンピュータサイエンス学科修了。Ph.D. in Computer Science. 1985 年から, 1988 年まで Bellcore (旧 AT&T ベル研究所) にて専任研究員としてマルチメディア情報ネットワークの研究に従事。1989 年より東洋大学工学部情報工学科助教授。1997 年同大学教授。1998 年より岩手県立大学ソフトウェア情報学部教授。同大学メディアセンター長。高速パケットビデオ, マルチメディアプロトコル, ハイパーメディアシステム, 感性情報処理等の研究に従事。IEEE, ACM, 電子情報通信学会各会員。



白鳥 則郎 (正会員)

1946 年生。1977 年東北大学大学院博士課程修了。1984 年同大学助教授 (電気通信研究所)。1990 年同大学教授 (工学部情報工学科)。1993 年同大学教授 (電気通信研究所)。情報通信システム, ソフトウェア開発環境, ヒューマンインタフェースの研究に従事。1993 年本会マルチメディアと分散処理研究会主査。IEEE Fellow.