

やわらかいマルチメディア会議システムの設計と実装

野村 尚央† 柴田 義孝† 白鳥 則郎‡

† 東洋大学工学部情報工学科

‡ 東北大学電気通信研究所

ネットワーク上に個々に分散して存在するマルチメディア情報を利用者の要求や目的に応じて統合して利用者に提供したり、処理を行ったりするような分散マルチメディア環境においては、動的/静的に変化する計算機資源やネットワーク資源環境においてもそのサービスの質を保証する必要がある。筆者らは、計算機やネットワーク等の資源環境/利用環境の変化に伴う場合でも、利用者に安定したマルチメディア通信機能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情報システムの研究を行っている。本稿では、そのシステムの応用例として、蓄積型及びリアルタイム型のメディアサービスを同時にかつ安定して同時に利用可能とする会議システムとして、やわらかいマルチメディア会議システムの開発を行ったので報告する。

Design and Implementation of a Flexible Multimedia Teleconferencing System

Takao Nomura † , Yoshitaka Shibata † and Norio Shiratori ‡

† Information and Computer Science Department, Toyo University

‡ Research Institute of Electrical Communication, Tohoku University

In order to provide and process multimedia information distributed over network to users by user's demands and purposes, multimedia system must guarantee QoS even though computing and network resources dynamically and statistically vary. We developed a flexible multimedia system using agent-based architecture for users to provide consistent multimedia communication function mechanism for user's demand and purposes in distributed environment. In this paper, we explain the design and implementation of a flexible multimedia teleconferencing system as one of the applications of the flexible multimedia system that users can take advantages of both stored and realtime media simultaneously.

1. はじめに

ネットワークを用いたマルチメディア情報サービスの利用は、マルチメディア会議システムを代表するように、データベースに格納されたマルチメディア情報を閲覧するような蓄積型メディアサービスや、遠隔地に離れた利用者同士が、互いの意志疎通やコミュニケーションを行うようなリアルタイム型メディアサービスとが混在することが一般的である。このようなマルチメディアサービスにおいては、プレゼンテーションメディアと実時間性メディアの両者を同時に利用するために、利用者の求める

情報を時間的、空間的に適切な形式で容易に提供できなくてはならない[1][2]。

しかしながら、これらマルチメディア情報の提供は、利用者のQoS(Quality of Service)要求や取り扱うメディアデータの特性、利用者ステーションの処理能力やネットワークの負荷変動等といった、動的/静的な状況変化によりサービスの質を保証できなくなる可能性があり、そのため、アプリケーションからネットワークまでの一貫したQoS保証と各層間における一連のQoS制御が必要となる[3]。

筆者らは、上述した問題点を考慮し、環境変化に

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システムとして、やわらかいマルチメディアシステムの研究／開発を行っている。本論文では、現在、そのシステム上にプロトタイプ例として設計／開発を行っているやわらかいマルチメディア会議システムについて述べる。

2. やわらかいマルチメディアシステム

2.1 定義

筆者らが研究／開発を行っている「やわらかいマルチメディアシステム」(以下FMS)とは、従来のマルチメディアシステムにやわらかさの概念[4]を取り入れ、システムのCPU負荷やネットワークの輻濺、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柔軟に対応し、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[5][6]。

そのために、筆者らはFMSを

1. 利用者にとって簡単で、使いやすいヒューマンインターフェイス機能
2. 環境の変化に対し動的に対応する機能
3. 状況に応じた資源の予約、交渉機能が行えるマルチメディアシステムと定義する。

本研究では、これらを実現する為の方法として、エージェントベースのアーキテクチャを導入する。本稿では、エージェント実現のための構成モデルとして、複数のオブジェクトにより構成され、かつ、自律性、社会性、反応性、協調性の基本的特性動作を行なえる処理体としてエージェントの実現を考える。このため、各々のオブジェクトをエージェントのコンポーネントとしてとらえ、特にマルチメディア処理を行うオブジェクトをメディアコンポーネントとして定義する。

2.2 FMS のシステムアーキテクチャ

FMSのシステムアーキテクチャはFig.1に示すように、大きく分けてClient Interaction Field(CIF)、Agent and Knowledge Repository Field(AKRF)、Component / Object Repository Field(CORF)の3つのコンポーネントにより構成される。

CIFは、利用者がマルチメディアサービスを受けるためのユーザステーション上において、エージェント及びコンポーネントが動作するCollaborative Client Interaction Space(CCIS)とActive Cooperative Agent Space(ACAS)の2つのコンポーネントにより構成される。

CCISは、ユーザパートナーエージェント(UPA)、インターフェイスエージェント(IA)、システム監視

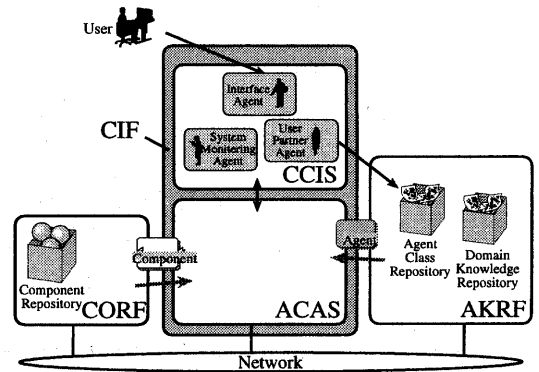


Fig.1 システム構成及びアーキテクチャ

エージェント(SMA)の、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。

ACASは、状況に応じたマルチメディア処理機能を行うため必要機能モジュールとエージェントの動的動作環境である。

AKRFは、計算機資源、ネットワーク資源および利用者要求に応じて、マルチメディアサービスを提供するエージェントが格納され、動的に起動することのできる場を想定している。本システムでは、エージェントに関する雛形やエージェントの用いる知識が格納されているAgent Template RepositoryとKnowledge Base Repositoryおよび、それらを用いることを可能とするAKRF Activity Engineから構成される。

CORFは、エージェントの構成に必要な様々なコンポーネントが格納され、動的に起動することのできるCORBAや分散オブジェクト環境のような場を想定している。本システムでは、コンポーネントおよびオブジェクトが格納されているComponent / Object Repositoryと、それらを動的に利用可能とするCORF Activity Engineから構成される。

3. やわらかいマルチメディア会議システム

一般的にマルチメディア情報の利用は、蓄積型のメディア利用と、リアルタイム型のメディア利用とに分けて考えられることが多いが、実際の利用者から見たマルチメディア情報を考えると、マルチメディア会議システムを代表するように、様々なメディアサービスを混在させながら同時に利用することが多い。このような複合型のマルチメディアサービスにおいては、例えば以下のような機能が必要である。リアルタイム・オーディオビデオサービスの場合、環境変化によりメディアデータの提供が途切れて、利用者間の会話性を壊さないために、厳格な同期よりは実時間性を優先したり、利用者の不快感

を与えないためにオーディオを優先しながらビデオを適切に提供する必要がある。蓄積型オーディオビデオサービスの場合、メディアの再現性を重視してサービスを提供したり、ある程度厳格なメディア間同期をとったり、メディアデータのキャッシュを行いながら、フレームレートの滑らかさを維持して提供する必要がある。これらの処理機能を実現するためには、Table.1,2にあげるようなシステム利用環境の属性と資源および環境変化と必要機能に柔軟に対応した制御メカニズムを必要とする。

しかしながら、従来のマルチメディアシステムにおいては、単独でこれらの問題の解決を行っていたり、これらの問題に柔軟に対応するメカニズムが無いといった点で、これらの問題を解決し、利用者に状況に応じたメディアの転送といった点からは、不十分で未熟である。

そこで、本稿では、サービスの形態やその目的等に応じて柔軟に対応し、さらに前述したやわらかさを実現を行なえるようなマルチメディア会議システムをやわらかいマルチメディア会議システムと定義する。

3.1 やわらかいマルチメディア会議システムの構成

やわらかいマルチメディア会議システムの構成を

利用者	計算機	ネットワーク
使用言語 年齢 専門分野	オペレーティングシステム メモリ量 計算機の演算能力 ディスプレイの出力分解能 入出力デバイス	ネットワーク構成 アクセス方式 ネットワーク速度

Table.2 システム利用環境の属性

Fig.2に示す。やわらかいマルチメディア会議システムは、上述したUPA、IA、SMAの他に、以下にエージェント種類とコンポーネントから構成される。

・Multimedia Service Management Agent (MSMA)

マルチメディアサービスのタイプにより、そのメディア間同期といった時間的關係や、マルチメディアシナリオに応じた空間的相對關係を調整、管理する。また必要な Media Agent を組織する。さらにこのエージェントでは、蓄積型のマルチメディアサービスの場合、Presentation Management Agent、リアルタイムオーディオビデオ通信の場合、Teleconference Management Agent等のエージェントがこれらの処理を行う。

・Media Agent (MA)

メディアデータを利用者に提供するにおいて、そのメディア毎の特性を知識を用いてメディア処理を行なうメディアコンポーネントを制御/操作する。このタイプに属するエージェントとしては、ビデオ通信を行う Video Agent、オーディオ通信を行う Audio Agent、テキスト通信を行う Text Agent 等が上げられる。

	環境変化	必要機能
計算機	処理量の変化 使用メモリ量	フレームレート制御
メディアデータ	データ量 フォーマット メディアの質	メディアフォーマット変換
ネットワーク	使用帯域の変化 信頼性の変化	メディアフロー制御

Table.3 資源/環境変化と必要機能

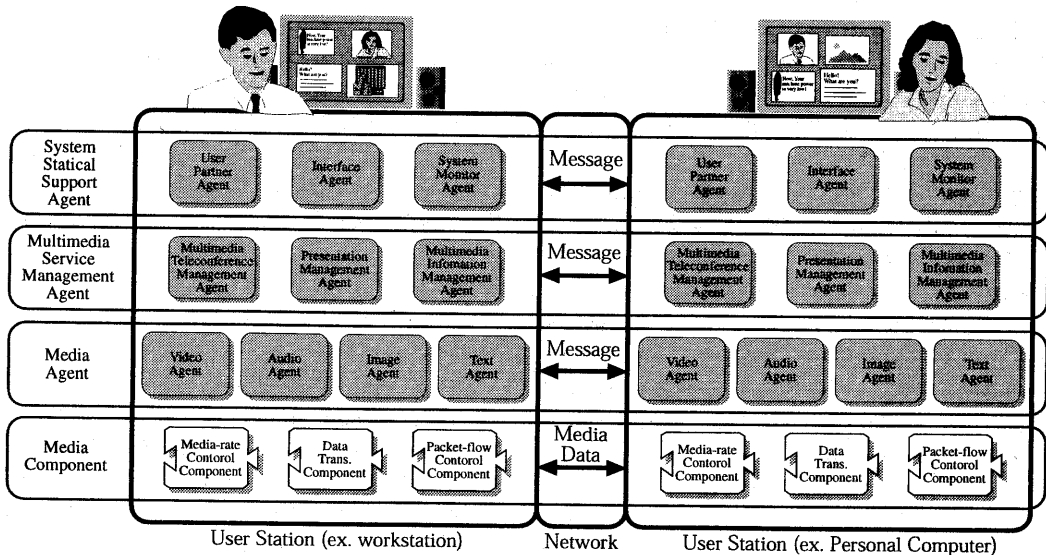
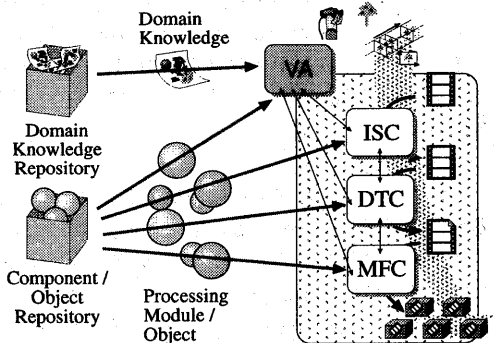


Fig.2 システム構成と階層

組織形成メッセージ		協調/交渉メッセージ		動作制御メッセージ	
メッセージタイプ	意味	メッセージタイプ	意味	メッセージタイプ	意味
Task	タスク通知	Proposal	提案通知	Direction	動作通知
Bid	入札通知	ContProposal	代案通知	Acception	許諾通知
Award	落札通知	Approval	賛成通知	Refusal	拒否通知
Request	情報要求	Reservation	保留通知	Report	状態報告通知
Report	情報通知	Refusal	反対通知	Result	動作結果通知
Termination	終了通知				

Table.3 メッセージのタイプ



VA : Video Agent
 ISC : Intra-media Sync Component
 DTC : Data Transform Component
 MFC : Media Flow Component

Fig.3 MA と MC の関係とプロトコル例

• Media Component(MC)

メディア処理を行うための完結した処理機能を有する独立した機能モジュールの集合体であり、MAにより受動的に制御/操作されることで、メディア通信を行う。本稿では、Table.2に示した必要機能をコンポーネント化して利用することによりメディア処理機能を提供する。

例として、ビデオを送信するためのMAおよびMCの構成をFig.3に示す。この例において、Video Agentは、知識リポジトリから、ビデオデータを配送するために必要なMCに関する知識と、それらをどのように制御/操作するかといった知識を取得し、その知識に応じて、MCを動的に起動することにより、環境の変化に応じたメディア処理を実現する。

3.2 プロトコルメッセージ

前述した、MSMA、MA、MCは、Fig.2に示した階層構造に従い、下位層のエージェントを組織化することにより、状況に応じたマルチメディアサービスの実現を行う。本研究では、エージェントおよびコンポーネント間のメッセージとしてTable.3に示すメッセージを用いる。

4. マルチメディア抽象化モデル

マルチメディア会議を代表されるように、蓄積型、リアルタイム型のメディアや、同一のメディア

サービスで、その通信方式の違うメディアを統一的に扱うために、メディア特性の相違を隠蔽し、簡単かつ容易な操作を実現するための統一的なメディア抽象化モデルとして、マルチメディア・サービス・ユニットMMSU、プレゼンテーション・ユニットPU、メディアオブジェクトMOを導入する。

各モデルについて以下のように定義する。

MMSU : 利用者に提供されるマルチメディアサービスの抽象モデル

PU : メディアの特性や表現形式が一致するメディアサービスを取りまとめた抽象モデル

MO : 個々のメディアの特性を考慮したメディア提供の抽象モデル

これらのモデルを用いることにより、利用者側からは、例えばMMSUに対しIAを通じて、「start」、「stop」等のサービス制御メッセージ送信することにより、PU,MOを構成するMAや各MCに対し、そのサービスの状況(コンテキスト)を制御する事ができ、また、そのサービスに対するQoS要求メッセージ、例えば、「フレームレート維持」、「画質優先」等の要求を送信する事により、PU、MOに対して現在のマルチメディアサービスの状況に応じた適切なQoS保証サービスを利用する事が可能となる。

以下にそれらを説明する。

4.1 メディア・オブジェクトMO

MOは、それぞれのメディアデータの特性、またそれらのメディア情報処理を実現するためのメソッドや制御機能を包括したモデルで、クライアント-サーバが一体となったモデルであり、

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

MC_i : メディア・コンポーネント
 MD : メディアデータ
 によって構成される(Fig.4)。

MCは、その取り扱うメディアの特性や、現在の計算機、ネットワーク資源の状態に応じてMAにより動的に起動される。本研究では、筆者らがこれらで研究/開発を行ってきた連続メディア転送に必要な機能をコンポーネント化し、メディア内同期コンポーネント、データ変換コンポーネント、パケット

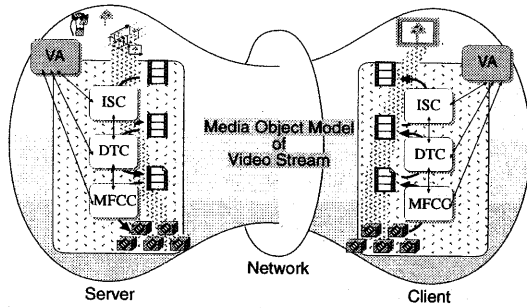


Fig.4 メディアオブジェクト

レート制御コンポーネント等を用いる。

4.2 プレゼンテーション・ユニット(PU)

PUは、メディア間の同期法、メディアの格納/発生方法、そして、メディアの特性や表現形式により、同様の属性を持ったMOを取りまとめたモデルであり、

$$PU = \{ \sum_{i=0}^n MO_i, PUI \}$$

PUI: PU インターフェイス

によって構成される (Fig.5)。

PUIは、PUに対する利用者が制御/操作を行うためのインターフェイスであり、MSMAによりメディアの特性、表現形式等の知識から、IS-A(Kind-Of)関係、HAS-A(Part-Of)関係により重み付けされ作成され、MSMAはIAに対してこれらPUIの生成要求を行う。ここで作成されるPUIはPUを構成する各MOに対する制御/操作方を統一したものである。

MSMAは、PUIの作成と、利用者要求に基づいたMOに対するメディア提供のスケジューリングや、メディア間の同期の方針等を決定する。利用者からのQoS要求は、MMSUからPUにマップされるために、IA、UPA、MSMAの順を経て適切にマップされ、これらは更にMOに渡される。

4.3 マルチメディア・サービス・ユニット(MMSU)

MMSUは、利用者側から見えるマルチメディアサービスの実体であり、利用者の望むサービスとその質(QoS)要求と、現在の計算機資源、ネットワーク

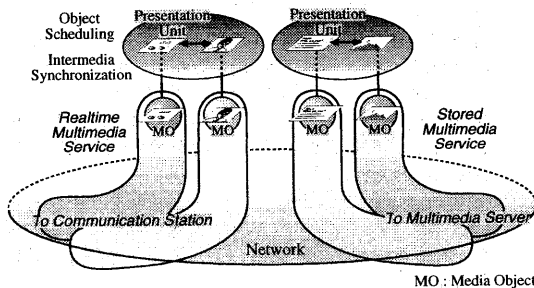
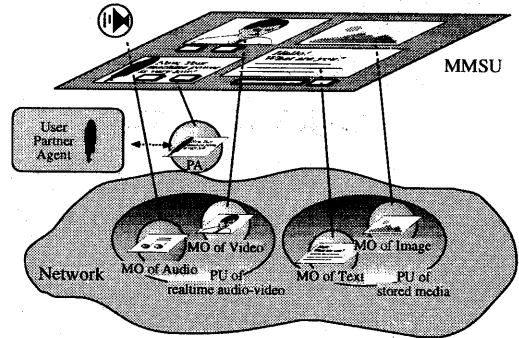


Fig.5 プレゼンテーションユニット



MMSU: Multimedia Service Unit
MO: Media Object

PA: Personalized Agent
PU: Presentation Unit

Fig.6 マルチメディアサービスユニット

資源の動的/静的な状況変化から決定される。MMSUは、そのサービスにおいて利用されるメディアサービスの属性や特性をまとめたPUの集合から構成され、

$$MMSU = \{ \sum_{i=0}^n PUI_i, UII, \sum_{i=0}^n SAI_i, PA \}$$

UI: ユーザインタラクションインターフェイス

SAi: システム内に存在するエージェント (IA, SMA, UPA)

PA: 擬人化されたエージェント

によって構成される (Fig.6)。

UIIは、利用者がマルチメディア情報サービスを利用するためのQoS要求やインタラクションを行うためのインターフェイスであり、IAにより提供される。利用者のサービスに対する要求は、このIAを通じてUPAに渡され、その利用者の要求の対象であるPUを構成するMSMAに対して、QoS制御メッセージを送信する。

5. やわらかいマルチメディア会議システムのプロトタイピング

現在、FMSプラットフォーム上において、やわらかいマルチメディア会議システムを開発中であり、第3章で示した構成を基に実装を行っている。実装にはSun WorkStation / Solaris 2.5.1上において、開発言語としてC++を用い、エージェント及びコンポーネントをC++のクラスオブジェクトとして実装を行っている。各エージェント及びコンポーネントはPOSIX互換のスレッドを用いることにより並行処理を実現する。さらに並行処理機能を簡単利用可能とするために、スレッド処理機能をオブジェクト化し、それを継承する形で実装を行っている。エージェントは、ポリモニックオブジェクトとして実現されるMCを動的にインスタンス化することにより、MCの動的起動を実現する。Fig.7に示すオブジェクトの階層図は特にMCの実装について示した実装例である。本プロトタイプは、ワークステー

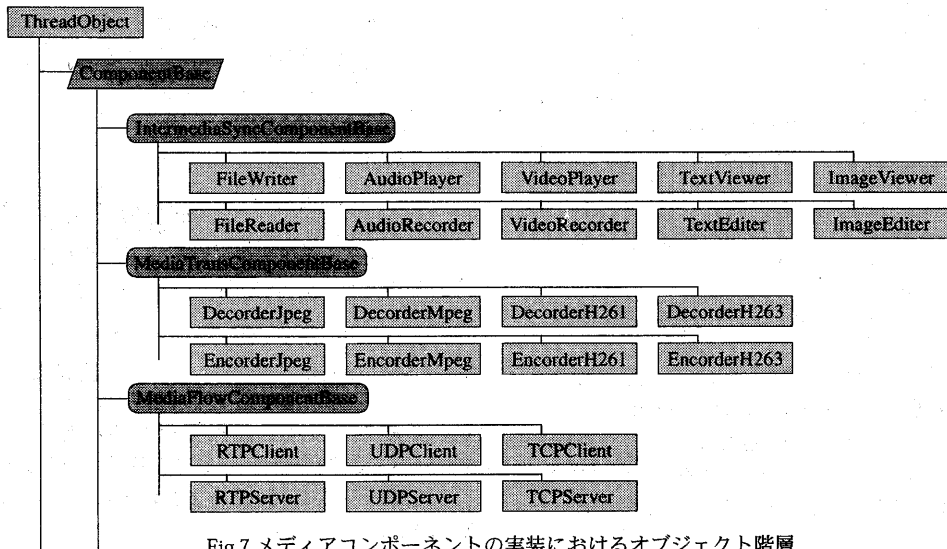


Fig.7 メディアコンポーネントの実装におけるオブジェクト階層



Fig.8 プロトタイプのインターフェイス例

シオン上に単一プロセスとして実行され、そのプロセス内においてスレッドが複数動作することにより並行処理を実現している。そのため、同一プロセス内のエージェント間の通信にはスレッド間でメッセージパッシングすることによりスレッド間通信を実現している。また、各端末間のエージェント間通信にはTCP/IPソケットを用いたIPC通信を用いている。エージェント及びコンポーネントの動的な起動には、メッセージを各エージェントが解析をし、ファイルに格納された知識を基に動的に対応するように実装を行っている。

Fig.8は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ーフェイスの例である。この例は、テレビ会議を行いながら同時に、分散するDBからマルチメディア情報を利用している例を示している。しかしながら現在のところ、同一レベルのエージェント同士の協調作業しかできてお

らず、総ての機能において、柔軟に対応はできておらず今後の課題となっており、現在開発を行っている。

6. まとめ

FMSの応用アプリケーションとして、やわらかいマルチメディア会議システムの実現方法と、現状について述べた。これにより、蓄積型、リアルタイム型のメディアサービスを安定して同時に利用者に提供できると考えられる。

しかしながら、現状においてはまだ一部のサービスしか提供されておらず開発途中の段階である。今後、初期プロトタイプの実装終了後、その性能評価を行う予定である。

参考文献

- [1] 勝本道哲:マルチメディア情報ネットワークのためのダイナミックハイパーメディアシステムの研究, 東洋大学大学院博士論文, 1996
- [2] ジョン・モース著他:分散マルチメディア技術, ソフト・リサーチ・センター
- [3] A. Vogel, B. Kerherve, G. Bochmann, J. Gecsei: Distributed Multimedia and QOS: A Survey, IEEE Multimedia Summer, Vol.2, No.2, pp. ,1995.
- [4] N. Shiratori, K. Sugawara, T. Kinoshita, G. Chakraborty, Flexible Network: Basic Concepts and Architecture, IEICE Trans. Commun., E77-B, pp.1287-1294, 1994.
- [5] 野村尚央他: エージェント指向に基づいたやわらかいマルチメディアシステムの研究, 情報処理学会, 第54回全国大会, 1997
- [6] 野村尚央他: 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情報提供システムの研究, 情報処理学会, DiCoMo ワークショップ, pp.395-400, 1997.