

# やわらかいマルチメディアシステムのための QoSを考慮したエージェントの組織化と実装について

橋本浩二<sup>†</sup>, 柴田義孝<sup>†</sup>, 白鳥則郎<sup>‡</sup>

<sup>†</sup>岩手県立大学 ソフトウェア情報学部  
{hashi, shibata}@iwate-pu.ac.jp

<sup>‡</sup>東北大学 電気通信研究所  
norio@shiratori.riec.tohoku.ac.jp

## Abstract

地理的に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ながら、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も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。これまで、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伴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してきた。本稿では、そのシステムにおけるエージェントの組織化と実装について述べる。

また、現在 RTP や RSVP といったメディアデータ転送時の QoS 保証を考慮したプロトコルが存在し、ATM ベースでの QoS 保証アーキテクチャに関する研究は盛んに行われているが、エンド間で利用者のサービスの質(Quality of Service: QoS)への要求を保証するマルチメディアシステムは存在しない。

そこで筆者らは、これらの問題を解決するためにやわらかさの概念 [2] に基づいたエージェント指向マルチメディアシステム (Flexible Multimedia System: FMS) の設計と実装を行ってきた [3]。本稿では、FMS の典型的な応用例としてやわらかいマルチメディア会議システムについて述べ、その構成要素であるエージェントの状態遷移と組織化プロトコルを明らかにし、QoS を考慮した組織化とその実装について述べる。

## 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、インターネットの普及などにより、マルチメディアを利用したテレビ電話や会議システムが利用可能となった。しかしながら、マルチメディアを利用した会議システムなどでは、地理的に離れた利用者同士が互いの意思疎通を行うためにリアルタイム型メディアサービスを利用しながら同時にネットワーク上に分散する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を利用することも想定される。したがって、このようなシステムは蓄積および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に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り、システムはメディア毎に異なる処理や制御方法とエンド間の QoS の保証 [1] が可能でなければならない。また、利用資源の状況変化の程度によっては、これらの制御機能を動的に組み替えて対応させることも必要となる。

## 2. マルチメディア会議システム

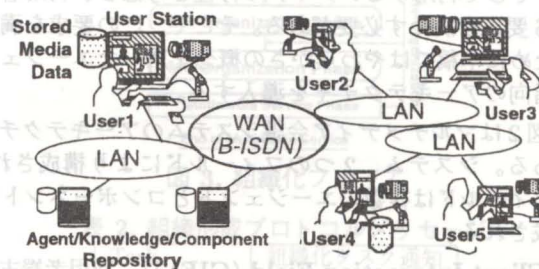


図 1. FMS の例: マルチメディア会議システム

本稿では、マルチメディアシステムの典型的な例としてマルチメディア会議システムをとりあげる。これは、図 1 に示すように地理的に離れた複数の利用者が LAN や WAN を通してマルチメディアを利用した会議を実現するものである。利用者は、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で互いに通信しながら、利用者ステーション内に既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やネットワーク上に分散するマル

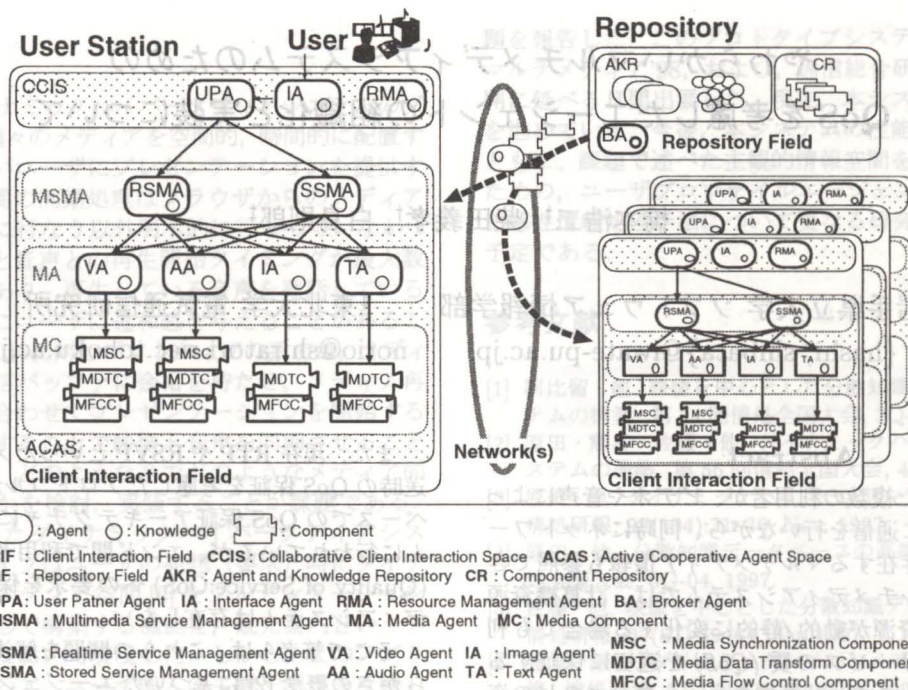


図 2. やわらかいマルチメディア会議システムの構成

チメディア情報を利用してプレゼンテーションを行うことが可能である。

このようなサービスは、複数のマルチメディアデータを同時に、かつリアルタイムに提供する必要があるため、利用者の環境や計算機とネットワークの資源利用状況、そして利用するメディアの特性を考慮し、利用者のQoS要求を満たす必要がある。そこで、この要求を満たすために本稿ではやわらかさの概念を用いたエージェント指向のアーキテクチャを導入する。

図2はマルチメディア会議システムのアーキテクチャである。システム、2つのフィールドにより構成され、各フィールドは複数のエージェントとコンポーネントで構成される。

**Client Interaction Field (CIF)** は、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、CCISとACASの2つの動作空間から構成される。

(1) **CCIS** には、システム常駐型の3つのエージェントが常駐する。UPAは次の(2) ACASで述べるMSMAを組織し、操作する。IAは、利用者からのサービス開始/終了要求やQoS更新要求などを処理する。RMAは、利用可能な資源利用状況を監視し、資源の割り当てや解放を行う。

(2) **ACAS** は、動的に組織されるエージェントとコンポーネントの動作空間である。MSMAは、リアルタイム、蓄積型といったマルチメディアサービスにおける各メ

ディア間の時間的および空間的関係の調整を行う。また、MAを組織して操作する。MAは、単一メディア毎に存在し、必要なMCを組織して操作する。MCは、メディア処理に必要な機能モジュールであり、大別して3つの機能が必要となる。MSCは、メディア内およびメディア間同期処理[4]を行う。MDTCは、JPEG、MPEG1/2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。例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。MFCCは、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[5][6][7]を行う。

**Repository Field (RF)** は、組織するエージェントや知識、コンポーネントを格納しておくフィールドであり、2つのリポジトリとエージェントにより構成される。AKRにはエージェントと知識が格納される。CRには必要となるメディアコンポーネントが格納される。BAは、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CIFへ提供するエージェントである。

そして、本システムではリアルタイムと蓄積メディアサービスを同時に処理することを想定し、RSMAとSSMAがそれらのサービスを管理する。

(1) **RSMA** は、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を組織し、その制御と管理を行う。例えば、会議中に会話が増え損なわれる場合、RSMA

表 1. FMS 構成要素

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	<b>UPA</b>	<b>MSMA, MA</b>	<b>RMA</b>	<b>RMA</b>

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、単位時間当りのビデオデータ転送量を抑えることにより、オーディオをデータの途切れを抑制する。

(2) SSMA は、蓄積メディアサービスを利用者に提供するための MA を組織し、MA の制御と管理を行う。例えば、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よう Video Agent に対して指示する。

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 により組織された MA は、そのメディア処理に必要となる MC を組織する。MA はメディア処理に必要となる 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。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、組織が進む方向を示している。また、RF から CIF への矢印は、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。

また、FMS における各エージェントは、様々なメディアサービスを統合して利用者へ提供するために表 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。

### 3. 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 3 に示す通り、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伴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。

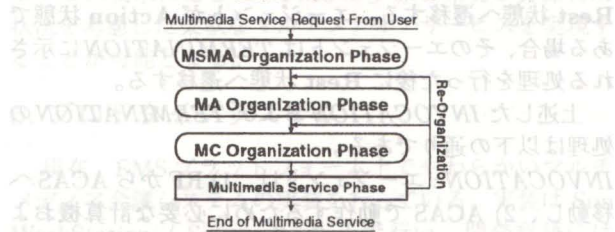


図 3. 組織化フェーズ

表 2. 組織形成プロトコルメッセージ

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

また、図 4 は、表 2 のメッセージによるエージェントの状態遷移を示している。各状態は下記の通りである。Rest は、エージェントの初期状態である。Task メッセージを受信することにより組織化を開始する。Wait Bid は、Task メッセージを送信したエージェントが、Task メッセージを受信したエージェントからの入札を待つ状態である。Wait Award は、Bid メッセージを送信することにより入札を行ったエージェントが、落札

を待つ状態である。Awardメッセージの受信に対し、そのエージェントがCIF内のACAS構成要素となる場合、エージェントはINVOCATIONで定義される処理を行い、Action状態に移移する。また、Awardメッセージの受信に対し、そのエージェントが他のエージェントを組織する場合、それらのエージェントに対してAwardメッセージを送信し、Action状態に移移する。Actionは、CIFにおいてマルチメディアサービスの実行が可能な状態である。

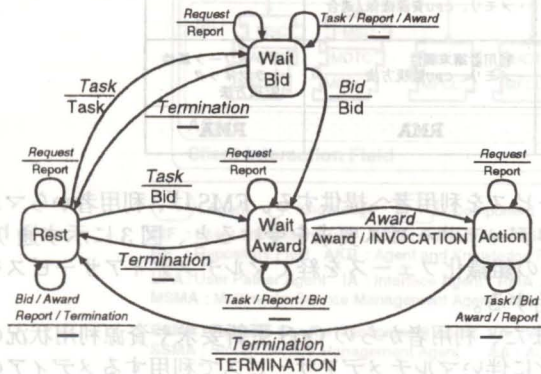


図 4. エージェントの状態遷移図

各状態において Request メッセージを受信すると、エージェントは必要な情報を Report メッセージにより返信する。また、Termination メッセージを受信すると、Rest 状態へ遷移する。エージェントが Action 状態である場合、そのエージェントは TERMINATION に示される処理を行った後に Rest 状態へ遷移する。

上述した INVOCATION および TERMINATION の処理は以下の通りである。

**INVOCATION:** エージェントは、1) RF から ACAS へ移動し、2) ACAS で動作するために必要な計算機およびネットワーク資源を確保する。

**TERMINATION:** エージェントは、1) INVOCATION 時に確保した資源を解放し、2) ACAS から RF へ知識を移動させる。

### 3.1. サービス開始時の組織化

FMS は、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。図 3 における各組織フェーズにおいて、エージェントやコンポーネントがどのように組織されるかをサービス開始時とサービス中に分けて述べる。

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、最初に MSMA を組織する。そのフローを図 5 に示す。図 5 は、テレビ会議のサービス開始要求に対し、リアルタイム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す。

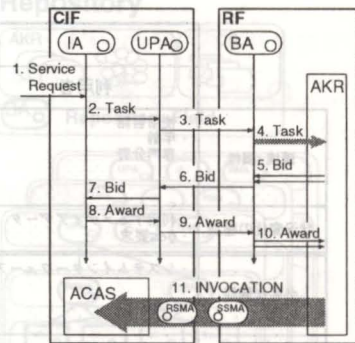
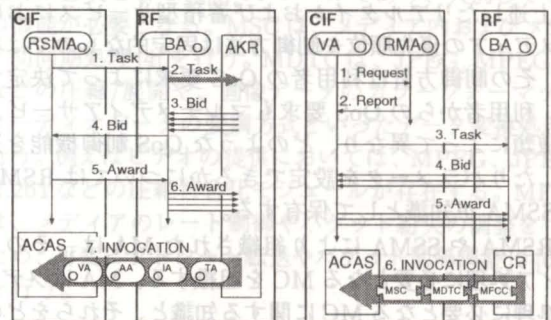


図 5. MSMA の組織化

まず、IA が利用者からマルチメディアサービス要求を受信すると、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 メッセージにより落札を行う。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 の組織を行う。



a) Organization of MA      b) Organization of MC

図 6. MA と MC の組織化

図 6 a) は、MSMA の 1 つである RSMA が、VA, AA, IA, TA を組織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 を組織する。

図 6 b) は、Video Agent (VA) が MC を組織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。

### 3.2. 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。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6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したエージェントが、削除すべきエージェントに *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である。

(3) エージェントやコンポーネントを交換する場合：まず、交換すべきエージェントやコンポーネントを削除し、図6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する。

## 4. 知識の生成と利用について

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保持し、利用する。

ACAS内に組織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もと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。蓄積された知識は、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、他の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：利用者のマルチメディア環境を特定するパラメータ。

**OS 種別**：実時間処理をするためのCPU資源確保が可能かどうかが決まる。確保不可能な場合、CPU資源を適合する必要がある。

**ネットワーク構成と利用可能なプロトコル**：帯域幅の確保や遅延、ジッタ制御が可能かどうかが決まる。不可能な場合、帯域幅の適合や遅延、ジッタの制御を行なう必要がある。

**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、音声デバイスなどで、利用可能なメディアが決まる。

(2) **資源利用状況パラメータ**：動的に変化する資源利用状況を把握するためのパラメータ。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の関係を各エージェントが知識として保有し、メディア処理を行うことにより、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。

## 5. プロトタイプシステム

現在、FMSプラットフォーム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 上でを行い、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、そのインターフェースの部分を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実現している。

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7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。全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。

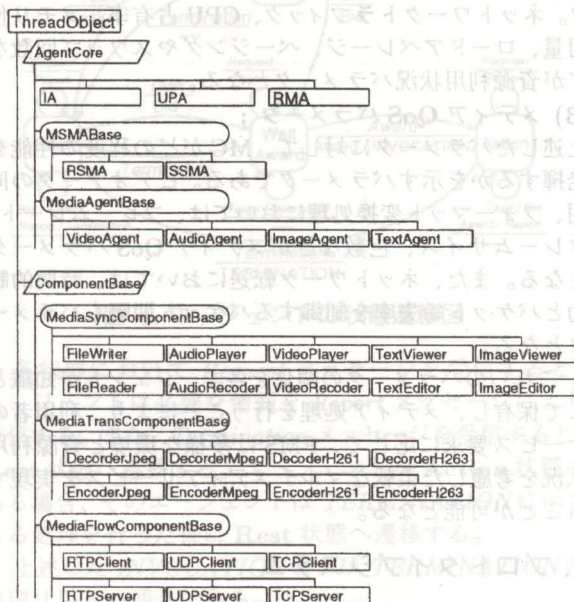


図7. オブジェクト階層

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を組織することが可能となった。しかしながら、FMSを組織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、さらにこれらのコンポーネントを分散化できるように実装を進め、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占有率は負荷変動の影響が大きく、逆にロードアベレージは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。

## 6. まとめ

本稿では、システム利用者の環境と資源利用状況の変化およびメディアデータの特性を考慮し、利用者のQoS要求を満たすマルチメディア情報を提供するFMSを提案し、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た。現在、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。

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確定しておらず、資源利用状況とメディア処理の関係は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やデバイス種別毎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、起動を実現する予定である。

## 参考文献

- [1] Koji Hashimoto and Yoshitaka Shibata.: *Performance Evaluation of End-to-End QoS Using Prototyped VOD System*, Proc. of ICOIN-12, pp.175-178, 1997.
- [2] 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.
- [3] 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, 情処研報 DPS-86-32, pp.183-188, 1998.
- [4] Yoshitaka Shibata, Naoya Seta and Syougo Shimizu.: *Media Synchronization Protocols for Packet Audio/Video System on Multimedia Information Networks*, Proc. of HICSS-28, pp.594-601, 1995.
- [5] 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, 情処ワークショップ論文集, Vol.96, No.1, pp.67-72, 1996.
- [6] Jun SATO, Koji Hashimoto, Yoshitaka Shibata.: *Dynamic Rate Control for Continuous Media Transmission*, IPSJ SIG-Notes, DPS-85, Vol.97, pp.85-90, 1997.
- [7] 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.