マルチメディア通信と分散処理 68－5
グループウェア 9－5
（1995．1．26）

連続メディア転送を考慮した情報間同期機構

瀬田直也 勝本道哲 柴田義孝

東洋大学 工学部 情報工学科
{seta,katsu,shibata}@yosemite.sb.cs.toyo.ac.jp

筆者らがこれまでに提案してきたダイナミックハイパーメディアシステムにおいて、複数のメディアデータから構成されるマルチメディア情報を、意味的に統合された一つの情報としてネットワークを越えて提供する際には、時間的関係を定義するプレゼンテーションシナリオに従ってメディアデータの転送・表示を行わなければならない。本研究では、リップ同期を要する時間的に密接な関係をもった複数のメディアをメディアグループとし、その複雑な同期・転送メソッドと共にカプセル化することにより、容易に動作制御が可能なメディアオブジェクト (MO) を定義した。本稿ではこの MO を制御単位として、シナリオに基づいたメッセージパッシングにより、連続メディアと静的メディアとのシーン同期、またプレゼンテーション全体を制御単位としたコンテキストスイッチングを実現する機構の考案・実装を行ったので報告する。

Synchronization Mechanizm considering Continuous Media Transmission on Distributed Network Environment

Naoya Seta, Michiaki Katsumoto and Yoshitaka Shibata

Department of Information and Computer Sciences
Toyo University
{seta,katsu,shibata}@yosemite.sb.cs.toyo.ac.jp

So far, we have introduced Dynamic Hypermedia System(DHS) using Knowledge Agent(KA) that enables users to retrieve multimedia information flexibly and effectively. In this DHS, it is desirable to transmit and display various media data according to presentation scenario so that they can be provided as semantically one integrated multimedia information object. We define a Media Group(MG) which consists of media data that should be transmitted over network and require "Lip Sync." among them. The MG and its involved processing are encapsulated into a Media Object(MO) which can be easily controlled based on message passing. In this paper, the implementation models which perform "scene sync." according to the presentation scenario by message passing to MOs and context switching between scenarios are described.

1 はじめに

これまでに筆者らは、マルチメディア情報ネットワークシステムの構築のため、そのプロトタイプについての設計・実装及びその性能評価を行ってきた[1][2]。またこの中で自由で発想的な情報検索を可能とするハイパーテキスト構造に着目し、そのリンクを動的に行いながら、よりユーザの要求に適合した検索を可能とした。更にハイパーテキストの各ノードをマルチメディア構造に置き換えることにより時間概念をもたせ、より柔軟な情報表現を可能とする

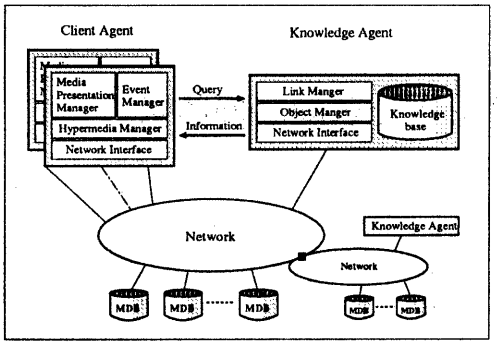

図 1: ダイナミックハイパーメディアシステム

ハイパーメディア構造の定義を行った [2]。またこれによる情報検索を実現するために図 1 に示すダイナミックハイパーメディアシステム (DHS: Dynamic Hypermedia System) の構造を提唱した [2]。このハイパーメディア構造の各ノードとなるマルチメディア構造は、複数のメディアデータとそれらを提供するための時空間情報であるプレゼンテーションシナリオから構成される。そこでこの複数のメディアデータを、意味的に統合された一つの情報としてネットワークを越えて提供するためには、そのシナリオに従ってメディアデータの転送・表示を行う必要がある。本研究ではこの中で特に、リップ同期を要する時間的に密接な関係をもった複数のメディアをメディアグループとして扱い、その転送・同期メソッドと共にカプセル化したメディアオブジェクトを定義した。またこのメディアオブジェクトを制御単位として、シナリオに基づいたメッセージパッシングにより、連続及び静的メディア間のシーン同期、またプレゼンテーション全体を制御単位としたコンテキストスイッチングを実現する機構の設計及び実装を行った。

2 同期モデル

本研究では、同期モデルをメディア間及びメディア内同期に大別する。

2.1 メディア間同期

DHS ではメディア間同期として、以下に述べるシーン及びリップ同期の二種類が定義されている。

シーン同期: メディア提供の開始点を時間的に制御する同期であり、シーン同期の同期点とは各メディアの表示開始時刻を意味する。具体的には図 2(a) に示すように、ある二つのメディア media 1,media 2 が parallel な関係にあるとき、ある基準時刻から t1, t2 秒後にそれぞれのメディアの提供を開始する。また media 1, 2 が図 2(b) のように serial な関係にあるとき、ある基準時刻から t1 秒後に media 1 の提供を開始し、media 1 の提供が終了してから t2 秒後に media 2 の提供を開始する。

リップ同期: 密接な時間関係があるメディア間で、相対的かつ連続的に行う厳密な同期である。ここでは一般に言われているリップ同期を拡張し、動画と音声間のみの同期でなく、更にその動画に対して字幕が表示されるといったように、3 つ以上のメディア間の同期をも含む。

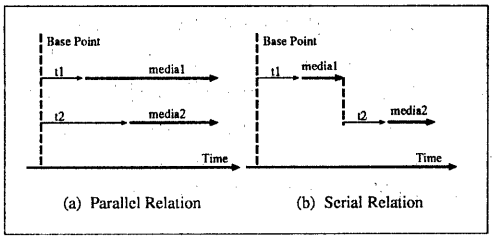

図 2: シーン同期

2.2 メディア内同期

連続メディアを提供する際には、ユーザに対して一定のフレームレートで再生しなければならないが、マシンやネットワークの不確定な負荷条件によって、実効フレームレートが低下または不安定になる可能性が十分にあり得る。これらの問題に対処するためには、以下に挙げる動画の再生制御が必要である。

時間優先制御: メディアの内容よりも、むしろ提供する時間や視覚的な速度感を重視する際に適用される。特に動画の再生においては、マシンやネットワークの負荷により設定されたフレームレートが満たせない場合は、動画の間引き表示を行うことによって実時間性を維持することにより、間延びした表示になることを防ぐ。

内容優先制御: 間引き処理により、内容上その意味が失われてしまう場合に適用される。例えば一瞬の重要な出来事を捉えた映像を再生する際、その出来事は動画中のわずか数フレームにしか記録されていないはずである。このような状況でフレームを間引くと、重要な出来事が提供されない恐れが生じる。この場合、提供時間を延長してでも全動画フレームを再生するべきである。

レート保持制御: 動画、音声などの連続メディアの提供の際に、それらが録画、録音されたフレームレート、またはアプリケーションやQoS制御によって設定されたフレームレートを正確に守るための基本的な制御である。この制御は、先に述べた二つの再生制御の中で常に行われる。つまりどのような負荷状況下においても、達成し得る実効レートが存在するので、システムはそのレートを保持して安定した連続メディアの提供を行わなければならない。

3 メディア統合の実現化

3.1 メディアグループ

時間的に密接な同期関係をもつメディアの扱いを容易にするためにメディアグループ(MG)を定義した[2]。例えば図3に示すシナリオにおいて、オーディオ1・ビデオ1、イメージ1間及びオーディオ2・ビデオ2間にはリップ同期が必要であり、またイメージ2はある時間においてオーディオ2・ビデオ2と同時に出力されなければならないが、これらの間には密接な時間関係は無いとする。ここで全てのメディアが独立であるとすると、これらのシーン同期を行なう際に6つの全メディアの同期制御を行わなければならない。また特定のメディア間のリップ同期をもそれらと同レベルで考慮しなければならず、高精度な同期を期待することは難しいと考えられる。そこでリップ同期が必要なメディア

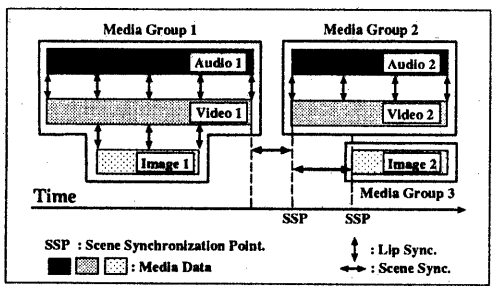

図3: メディアグループ

群をそれぞれグループ化し、それを一つのメディアとして扱う。その結果、図3のシーン同期のスケージュールでは高々3つのメディア(MG 1~3)を考慮すればよく、またリップ同期は同一MG内で実現されるのでこの同期レベルでは考慮する必要がなくなる。

3.2 メディアオブジェクト

シーン同期をメッセージパッシングにより容易に実現するため、ネットワークを考慮したデータモデルとしてメディアオブジェクト(MO: Media Object)を定義した[2]。図4に示すようにMOはその資源として、一つのMG及びそれを処理するためのメソッド群を有する。このメソッドにはデータ転送も含まれるためMOはネットワーク上に存在することが可能であり、効率的なパケット転送を行いながらMGを構成するメディア間のリップ同期の提供が可能である。この概念によりMGを構成するメディアタイプに関わらず、統一的な転送及び表示制御を行なうことが可能となった。

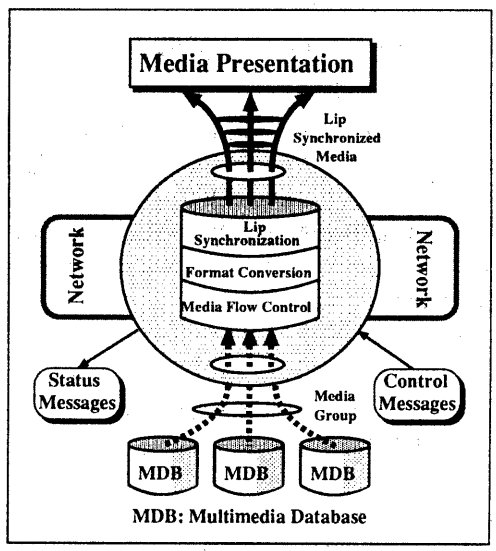

図4: メディアオブジェクト

3.3 MOの実装

MOのメソッド部分は図5に示すように、Message HandlerとMedia Handling Methodsに分離することができる。この利点としては、ビデオやイメージ転送表示等の既存システムにMessage Handlerを追加するだけでMOとしての制御が可能になることが挙げられる。

Message Handler: このモジュールはMOの外部インターフェースとして機能し、制御メッセージを非同期に受信して該当するメソッドを呼び出し、状態メッセージを返信する。

Media Handling Methods: これはメディアデータの操作を行うモジュール群であり、この操作に

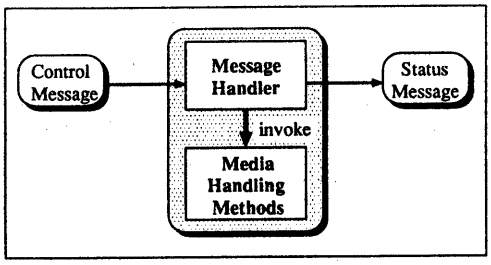

図 5: MO の実装モデル

は、データ転送、フォーマット変換、リップ同期処理などが含まれる。メディアタイプよって必要なメソッドが異なるので、扱うメディアタイプ毎に MO がもつ操作メソッドが異なる。

4 メディア同期の実現

4.1 リップ同期

MG を構成するメディア間のリップ同期は、MO のメソッドによって実現される。このリップ同期については、Strict, Relaxed 及び Silence Detected 同期の 3 種類の同期方式 [3] を適用することにより、ビデオ及びオーディオフレーム単位での高精度な同期が可能である。

4.2 シーン同期

マルチメディアプレゼンテーションにおけるシナリオに基づいたシーン同期は、MO 間の同期制御によって実現される。いま図 6 (a) に示すように、あ

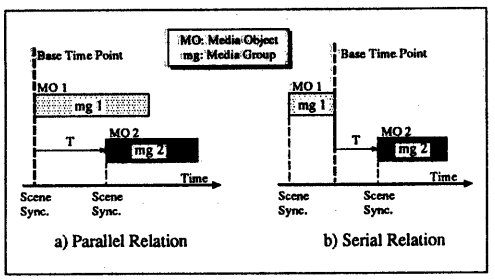

図 6: メディアオブジェクト間の同期

る二つのメディアグループ mg 1, 2 が parallel な関係にあり、それらを扱うメディアオブジェクトがそれぞれ MO 1, 2 であるとする。この場合 mg 1 の同期点において MO 1 に開始制御メッセージを送信し、その時刻から T 秒後に MO 2 に開始制御メッセージを送信する。これにより mg 1, 2 のシーン同期が実現される。また図 6 (b) のように mg 1, 2 が serial な関係にある場合は、同様に mg 1 の同期点において MO 1 に開始制御メッセージを送信する。その後 MO 1 からの終了状態メッセージを受信し、その T 秒後に MO 2 に開始制御メッセージを送信する。このように MO 1 の終了を確認してから MO 2 の開始を行うことにより MG 間の serial な関係を保存する。このメディアオブジェクト間のシーン同期制御は、次に述べる マルチメディアコントローラ によって実現される。

4.2.1 マルチメディアコントローラ

図 7 に示すようにマルチメディアコントローラ (MMC: Multimedia Controller) は一つのプレゼンテーションのために Client 及び Agent MMC が一対起動され、それらがシナリオに基づいた同期制御を行う。MMCs はプレゼンテーションを構成する

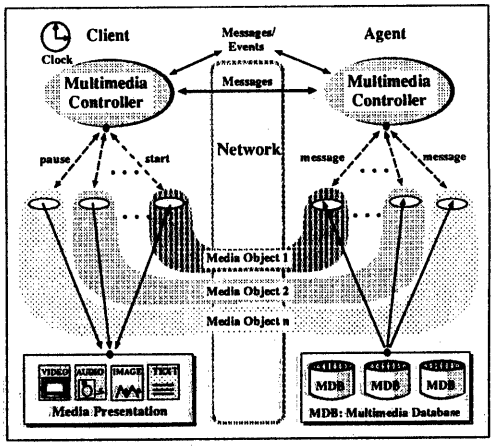

図 7: MMC のネットワークモデル

全ての MOs の管理/制御を行ない、Client MMC は各 MO へのシナリオに基づいたタイミングでのメッセージパッシングにより、シーン同期制御を実現する。このとき Client MMC は各 MO からの状態メッセージを受信し、その状態を把握しながら開始/停止/終了といった制御を行なう。また Client MMC は、プレゼンテーションの進行程度を示すシナリオ時間を管理し、ユーザからの時間的イベントを検出すると MOs に制御メッセージを送信し、現在参照時刻の移動を行なう。ここでプレゼンテーションの開始から終了までの Client MMC の処理の流れを示す。

1. シナリオの取得
2. プレゼンテーションに必要な MOs の起動

3. メッセージパッシングによるシーン同期制御
4. ユーザイベントに応じた関連 MOs の制御
5. プレゼンテーション終了に伴う MOs の終了

4.2.2 MMC の実装

図 8 に示すように、Client MMC は 3 つのモジュールから構成される。それぞれのモジュールの機能を以下に述べる。

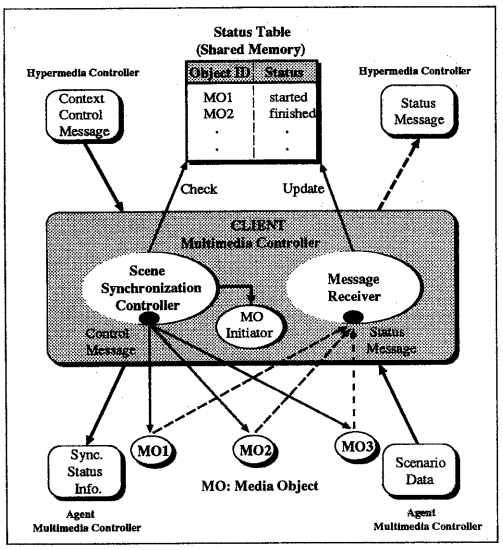

図 8: Client MMC の実装モデル

MO Initiator: このモジュールはマルチメディアプレゼンテーションの開始前に、必要となる MOs を起動する。

Scene Synchronization Controller: このモジュールは、シナリオに従って MO 群にメッセージを送信することによりシーン同期を実現する。この際共有メモリ上の状態テーブルを参照し、該当 MOs が準備完了状態であることを確認した後、それらに対して開始制御メッセージを送信する。またある MG の提供終了の確認も、この状態テーブルを監視することにより行う。

Message Receiver: このモジュールは、各メディアオブジェクトから非同期に送信される様々な状態メッセージを受信し、その状態を共有メモリ上に実装された状態テーブルに書き込む。これにより、Scene Sync. Controller は常に MOs の状態を把握することが可能となる。

5 コンテキストスイッチング

ハイパーメディア構造においては、複数のマルチメディアプレゼンテーションが同時に提供される場合があり、それらのコンテキストスイッチングが定義されている [2]。これを実現するためにはプレゼンテーション単位の動作制御、つまり先に述べた MMCs の制御が必要となる。そこで **ハイパーメディアコントローラ** を導入し、先に述べた MMC に対してメッセージパッシングを行うことにより、コンテキストスイッチングが実現される。

5.1 ハイパーメディアコントローラ

ハイパーメディアコントローラ (HMC: Hypermedia Controller) はシステムの起動時に、Client 及び Agent HMC が起動され、MMCs の管理や、各シナリオ内に記述されているオーサの指定に基づいたコンテキスト制御を、メッセージパッシングにより行う。また Client HMC は MMC から状態メッセージを受信し、その状態を把握しながら開始/停止/終了といったコンテキスト制御を行なう。図 9 に MMC を含めた HMC のネットワークモデルを示す。 またプレゼンテーションの開始から終了ま

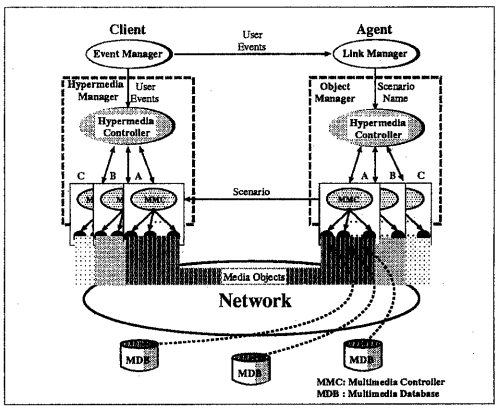

図 9: HMC のネットワークモデル

での Client HMC の処理の流れを示す。

1. Agent HMC からのシナリオ名の受信
2. MMC を起動してシナリオ名を通知
3. ユーザイベントに応じた MMCs の制御
4. プレゼンテーション終了に伴う MMC の終了

5.2 HMC の実装

図 10 に示すように、Client HMC は 3 つのモジュールから構成される。それぞれのモジュールの機能を以下に述べる。

MMC Initiator: このモジュールは、マルチメディアプレゼンテーションの開始毎に新たな MMC を起動し、それにシナリオ名を与える。

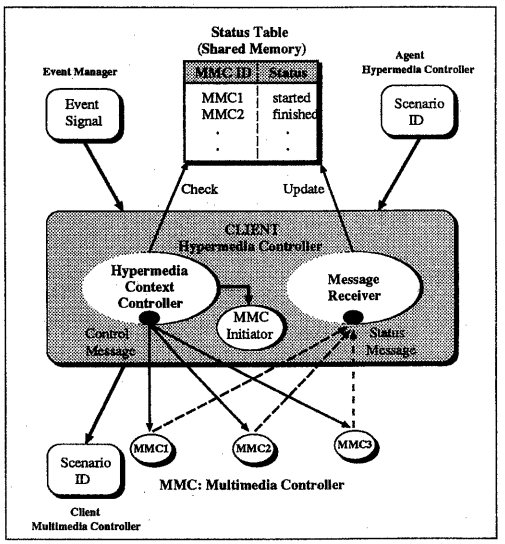

図 10: Client HMC の実装モデル

Hypermedia Context Controller: このモジュールは、シナリオの指定に従って MMCs にメッセージを送信することによりコンテキストスイッチングを実現する。この際共有メモリ上の状態テーブルを参照し、コンテキストスイッチ時における該当 MMC の状態を確認した後、制御メッセージを送信する。またあるプレゼンテーションの終了確認も、この状態テーブルを監視することにより行う。

Message Receiver: このモジュールは、各 MMC から非同期に送信される様々な状態メッセージを受信し、その状態を共有メモリ上に実装された状態テーブルに書き込む。これによって、Hypermedia Context Controller は常に MMCs の状態を把握することが可能となる。

6 プロトタイプの構築

これまでに述べた MMC 及び HMC の機能を評価するために、FDDI ネットワーク上に接続された 2 台の RISC ベースのワークステーションをそれぞれ Agent, Server として、X-Window 上に C 言語をベースとした開発を行いプロトタイプを構築している。アプリケーション例としては図 11 に示すような川越の歴史と文化を紹介する「電子博物館 "川越"」 [1] を仮定している。このアプリケーション上においてユーザは、ハイパーテキストの概念に基づいた自由で発想的な情報検索を行うことが可能であり、それにより選択されたマルチメディア情報が、シナリオに基づいたプレゼンテーションによって提供される。

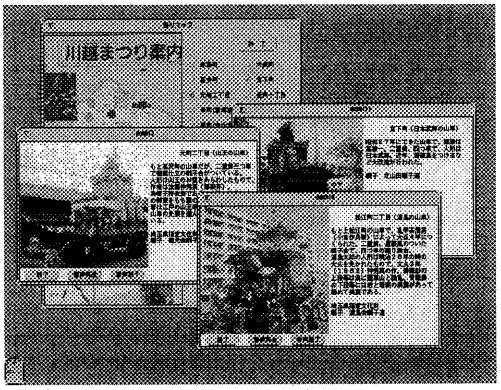

図 11: 「電子博物館 "川越"」

7 まとめ

本稿ではまず DHS における同期モデルについて、メディア間ではシーン及びリップ同期、メディア内では連続メディアの再生制御を定義した。またメディア統合の実現化方式として、MG, MO の概念について詳細に述べ、MO の実装方法を示した。更にシーン同期及びコンテキストスイッチングの実現方式として MMC による MOs の制御、HMC による MMCs の制御方法について述べ、各実装方法の概要をまとめた。最後にプロトタイプの構築について述べたが、今後はこの「電子博物館 "川越"」の実装を終え、その機能評価を中心に行うことにより、ここで述べたシステムの有効性を検証する予定である。

参考文献

[1] Yoshitaka Shibata, Michiaki Katsumoto: Dynamic Hypertext and Knowledge Agent Systems for Multimedia Informtion Networks, ACM Proc. of Hypertext'93, pp.82-93, Nov.1993.

[2] 勝本 道哲, 瀬田 直也, 柴田 義孝: ダイナミックハイパーメディアシステムへの時間的同期機構の導入, 情処ワークショップ論文集, Vol.94, No.1, pp.259-268, Jul.1994.

[3] 瀬田 直也, 柴田 義孝: パケットオーディオ・ビデオシステムの同期方法, 情処研報 DPS-64, Vol.94, No.19, 1994.