

# 分散環境におけるマルチメディア処理機構の実装と評価 — 情報提供システム Cosaic —

田中裕之 吉川耕平† 岡村耕二 荒木啓二郎  
奈良先端科学技術大学院大学 情報科学研究科

(†シャープ株式会社技術本部より派遣留学中)

E-mail {hiro-tn, kouhei-y, oka, araki}@is.aist-nara.ac.jp

我々は、音や映像など複数の知覚メディアを扱うマルチメディア処理において問題となる、表現メディア同士の関連性や時系列上での連続性の保証を考慮した計算機環境として PDE-II を提唱して、その実装および評価を進めている。本稿では、PDE-II においてマルチメディア処理を行うために考案したアプリケーションインタフェイスと、それを利用したアプリケーションの開発について述べる。また我々は、インタフェイスの評価のためのアプリケーションとして、近年注目を集めている情報提供システム WWW(World Wide Web) に複数メディアを同時に扱う機構を拡張したアプリケーション “Cosaic” の設計と実装を行っている。本稿では Cosaic の実装とその品質評価実験によってインタフェイスの有効性を示す。

## 1 はじめに

近年の計算機と通信回線の進歩によって従来の文字に加えて音や動画像の入出力が可能なハードウェアが安価で利用可能になりつつある。これとともに、複数の知覚/表現メディアを統合して扱うマルチメディア処理に対するユーザの期待が高まっている。特に情報提供システムと双方向対話型コミュニケーションシステムの分野への応用に対する期待は大きく、様々なアプリケーションが開発されている。

例えば、現在世界中に普及している広域ネットワークであるインターネットでは、複数メディアからなる情報とそれらの関連づけの情報によって構成されるハイパーテキスト形式の情報提供機構 World Wide Web(WWW)[1] を利用したサービスが急速に広がりつつある。また双方向対話型コミュニケーションの分野では、IP マルチキャスト [2] の技術を応用し

た双方向音声会議システム “VAT” や画像放送システム “nv” が開発され、会議中継などの分野で試験的に使用されている [3][4]。

しかしながら、インターネットでの通信や、現在学術研究機関で広く利用されている UNIX オペレーティングシステムに代表される処理時間に対する保証ができない計算機環境では、データの処理時間に揺らぎが生じる。従って音や動画像のような時間的制約のある表現メディアの処理を行うアプリケーションを実現するためには、メディア処理の際の処理時間の揺らぎの影響を解消して、連続した記録、再生を行う工夫が必要となる [5]。

そこで我々は、複数の知覚メディアを知覚/表現する際の連続性や同期を保証する機構を備えた計算機環境として PDE-II を提案している [6]。PDE-II は論文 [7] で我々が提示したマルチメディア処理モデルに基づいたハードウェアに依存しない計算機環境である。

本稿では PDE-II 上でマルチメディア処理を実現するために我々が用意したインタフェイスと各種表示メディアを制御する機構の概要を述べる。また本稿では、これらの機構を利用して実装を進めている情報提供システム “Cosaic” を紹介する。Cosaic は World Wide Web のハイパーテキストの参照方式を応用

---

“Implementation and Evaluation of Multimedia Processing on PDE-II — Multimedia Information Retrieval System “Cosaic” —  
H.Tanaka, K.Yoshikawa†, K.Okamura and K.Araki  
Graduate School of Information Science, Nara Institute of Science and Technology  
†also with SHARP Corporation

して、ディスク上に蓄積された情報のみならず、表示メディアから実時間で連続して取り込んだ情報をディスクに蓄積することなく即座に提供する、非蓄積型の情報提供を可能にしたツールである。

本稿では、Cosaic の実装とその評価を行うことによって、PDE-II における制御インタフェイスの有用性を示す。まず 2 章では、PDE-II の概念と実装の概要について述べる。次に 3 章では、PDE-II 上で複数メディアを扱うアプリケーションを作るために用意した入出力/通信機構について紹介する。4 章では Cosaic の概要と実装について説明し、5 章では Cosaic の評価実験とその結果について述べる。最後に 6 章でまとめと今後の課題について述べる。

## 2 PDE-II の概要

テキスト、音、映像など複数の知覚メディアを並行に処理するマルチメディア処理では、各表現メディアの時系列上での連続性や、メディア間の関連性を失うことなく処理を行う必要がある。我々が提唱している PDE-II は、分散環境でメディアの関連性や連続性を失うことなくマルチメディア処理を実現するための処理環境である。

### 2.1 マルチメディア処理モデル

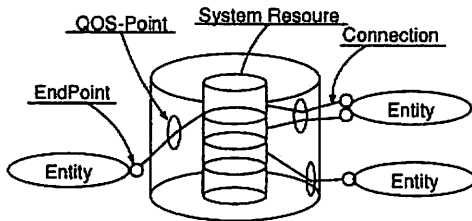


図 1: PDE-II のマルチメディア処理モデル

知覚/表現メディアの処理を行う場合、メディア間の同期のずれや時系列上での連続性の喪失の割合といった、ユーザに提供されるサービスの品質 (Quality of Service, QOS) の定義を行い、サービスの品質管理をその定義に基づいて行う必要がある。

我々は、サービスの品質を単一メディアの品質と複数メディア間の同期の品質の 2 つに分類してそれぞれ以下のように定義している。

- 単一メディアの品質

単一メディアの品質は、そのメディアが再生される時間の間隔と一時に再生される情報量で表す。あるメディアの再生間隔を短くして一時に再生される情報量を多くすると、そのメディアの品質は高くなる。例えば動画像の場合、単位時間当たりの再生コマ数を増やすことと、1 コマ当たりの画像の解像度を上げることが、それぞれ再生間隔を短くすることと一時に再生される情報量を増やすことに相当する。

- 複数メディア間の品質

複数メディア間の同期の品質は、同時に記録された複数のメディアが再生される時のメディア間の相対的な時間のずれで表す。同時に記録された複数メディアの再生時の時間のずれが小さければ小さいほどそれらのメディアの品質は高い。

我々は以上のようなサービスの品質について考慮した処理を行うアプリケーション製作の複雑さを軽減するために、PDE-II のシステムの機能として品質保証機構を用意した。図 1 に PDE-II のマルチメディア処理モデルを示す。

PDE-II のマルチメディア処理モデルではデータを処理する主体をエンティティ、エンティティ間の通信路をコネクションと定義して、全てのアプリケーションをエンティティ内でのデータ処理とコネクションを用いたエンティティ間通信で表現する。また、エンティティからコネクションにデータを入出力する部分をエンドポイントと定義している。エンティティは任意の数のエンドポイントを生成することによって任意のエンティティとコネクションを結ぶことが可能である。エンティティ間通信の QOS はコネクション両端に用意される QOS ポイントによって保証する。従って従来アプリケーション毎に実装されていたような QOS 管理機構をエンティティ内で実装する必要はない。

### 2.2 PDE-II のアーキテクチャ

前節で述べた PDE-II のモデルに基づいて、現在我々はマルチメディア処理環境 PDE-II の実装を行っている。PDE-II の実装時のアーキテクチャを図 2 に示す。PDE-II のアーキテクチャは、エンティティレイヤとコネクションレイヤの 2 つの階層で構成する。

- コネクション層

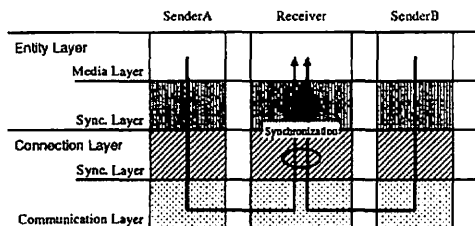


図 2: PDE-II のアーキテクチャ

コネクション層ではデータ転送レート制御及びデータ同期転送制御が考慮されたエンティティ間通信のための機能を提供する。我々は、コネクション層を機能に応じて同期層と下位レベルコミュニケーション層の2つの階層に分割している。

－ 同期層

コネクション層内の同期層では、データ送受信速度制御および同期転送制御などの PDE-II のコネクション管理を行う。この階層ではデータ送受信速度制御と同期制御を始めとする PDE-II のコネクションの機能を、実際にデータを送受するストリームの制御を行っている下位レベルコミュニケーション層の機能を利用して実現する。

－ 下位レベルコミュニケーション層

この階層は、OSI の通信モデルにおけるトランスポート層に相当する機能を提供する。各種トランスポートプロトコル間のストリーム制御方法の差異はこの階層で吸収され、同期層およびエンティティ層には影響しない。我々は、トランスポート層およびそれ以下の層のプロトコル自体に関する改良については検討を重ねている段階であるため、現在は TCP/IP を下位レベルコミュニケーションプロトコルとして利用している。

● エンティティ層

エンティティ層では、記録再生や圧縮符号化、復号化などメディア処理のための機能が提供される。エンティティ層もコネクション層と同様以下の2つの階層に分割している。

－ メディア層

メディア層は、メディアの記録再生のための、符号化および復号化の機能を提供する。この階層で記録、符号化された表現メディアはコミュニケーション層の機能を用いて伝達され、再びこの階層で復号化して再生される。またこの階層では、メディアを入出力するデバイス自体に関する情報など、表示メディアや表現メディアに依存する情報も管理する。

－ 同期層

エンティティ層内の同期層は、アプリケーションレベルでの同期制御を提供する。アプリケーションレベルでの同期は表現メディアの記録、再生をする際にデバイスへの入出力のタイミングを同期させる機能を持つ。

以上のような階層で構成されたアーキテクチャに基づいて、我々は現在 PDE-II の実装を各層ごとに進めている。次節ではこれらの各階層の中から特にエンティティ層を取り上げて、アプリケーションインタフェイスの仕様とその利用例について述べる。

### 3 PDE-II のアプリケーションインタフェイス

従来のアプリケーションでは、デバイスおよびネットワーク通信の入出力機構に品質制御を行う機能がないため、アプリケーション毎に品質制御機構を実装する必要がある。このような作業はマルチメディア処理を行うアプリケーションの実装の際に大きな負荷となる。また、アプリケーション間でデバイスを共有する場合やアプリケーション間でデバイスへの入出力を同期させるといったようなアプリケーション間の協調動作の実現が困難である。

そこで PDE-II では、品質制御をエンティティ間のコネクションの制御機構の一部として提供することにより、アプリケーション製作時の各種制御機構を実装する負担を軽減する。また、音および画像の入出力を行う仮想的なデバイスとして、オーディオ/ビデオエンティティを用意して、デバイスに依存した制御方法の差異やデバイス間の入出力の同期を実現している。

表 1: コネクション制御関数一覧

エンドポイント / コネクション 制御	
エンドポイントの生成	epID=Create_EndPoint("Name")
エンドポイント識別子の解決	epID=Resolve_EndPoint("Name")
コネクション接続要求	cID=Connect_Connection(epID,cID')
コネクション接続要求受理	cID=Accept_Connection(epID)
コネクションの終了	Stat=Close_Connection(cID)
エンドポイントの削除	Stat=Delete_EndPoint(epID)
データの送受	
データの送信	Stat=Send(cID,Data,E_Ctrl,QOSstat)
データの受信	Stat=Recv(cID,Data,E_Ctrl,QOSstat)
データの品質の制御	
コネクションに対する要求	QOSstat=QOS_Request(cID,QOS)

epID: エンドポイント識別子      cID: コネクション識別子  
 Data: 表示メディアのデータ      E\_Ctrl: エンティティ制御情報  
 QOSstat: 品質制御に関する情報      QOS: 品質制御パラメータ  
 Stat: 関数の処理結果の情報      Name: エンドポイントの識別名

### 3.1 エンティティ間の通信

PDE-II では、知覚メディアを扱うデバイスやアプリケーションはすべてエンティティとして扱い、表現メディアはエンティティ間を結ぶコネクションによってやりとりされる。PDE-II で用意しているコネクション制御関数の一覧を表1に示す。これらの関数は、パークレーソケットライブラリにおける TCP コネクションの処理手順とほぼ同様の操作手順でコネクションの生成、データの入出力、コネクションの終了などの機能をエンティティに対して提供する。

コネクションインタフェイスとソケットライブラリの最大の違いは、処理の品質の制御が指定可能である点と、データの送受を行う関数で、データ本体とは別にエンティティ間制御情報を同時に送受することを考慮している点である。

#### 3.1.1 コネクションの生成と削除

コネクションの生成は、Create\_EndPoint 関数でエンドポイントを生成した後、Accept\_Connection 関数でそのエンドポイント

に対する他からのコネクション接続を待っているエンティティが、Connect\_Connection 関数による接続要求を受けた時点で生成される。Connect\_Connection 関数で指定するエンドポイントは、接続待ちをしているエンティティが生成時に指定した名前から確定する。

エンドポイント識別子と、識別子の生成時に対応づけるエンドポイントの識別名の管理は、ホストごとに1つずつ起動するエンティティ管理プロセスによって行う。またコネクション間の同期に関する処理設定は、同期するコネクションの識別子を Connect\_Connection 関数の引数で指定することによって下位同期層に伝達する。

#### 3.1.2 データの送受信

データの送受信を行う関数 Send,Recv では、通常データの転送とエンティティ間の制御メッセージなど重要度の高い制御メッセージの転送を区別して送受信制御を行う。これは、エンドポイント間での通信の品質制御によって淘汰されてもよいデータと、必ず伝達すべき制御メッセージとを明確に分けるこ

とによって、コネクションでの品質管理を利用し易くするためである。

エンティティ間の制御メッセージの内容は、全エンティティで共通な部分と、エンティティの種類に依存する部分の2つから成る。共通部分には送信データの生成時間など、録音/再生時の時間情報が含まれている。これらの時間に関する情報はエンティティ層での同期における指標として利用する。またエンティティに依存する部分の制御メッセージは、オーディオエンティティにおける録音/再生音量の制御など、エンティティ毎に固有な制御情報の伝達に利用する。

### 3.1.3 データの品質保証

下位層でのデータの品質保証機構を制御するために、PDE-IIではいくつかのインタフェイスを用意している。QOS\_Request 関数はデータの送受信速度に関する特性の指定などの、コネクション層における品質保証の要求を行うためのインタフェイスである。また、Send\_Recv 関数はデータ送受信時のデータの品質制御に関する情報を返り値としてエンティティに返すので、アプリケーション側で現在のサービスの品質を判断して品質制御に関する指定を変更することができる。

エンティティ層における同期処理はエンティティ間制御メッセージ中の時間情報を元に行われる。時間情報に基づいたエンティティ層での同期制御は、仮想デバイスエンティティ内に、実デバイスからの入出力の時点で時間の記録/調整をする機構を実装することによって実現する。このため、エンティティ層における品質保証機能は仮想デバイスエンティティの特性に応じて実現することが可能である。例えば、Cosaicの実現のために実装したオーディオエンティティとMPEGエンティティでは、入出力の同期を取りたいコネクションの時間情報を元に入出力のタイミングを決定する拡張論理時間同期機構 [8] を採用している。

### 3.2 仮想デバイスエンティティ

計算機上で音や動画の入出力を行う場合、以下のような問題点を考慮してデバイスドライバを制御する必要がある。

- 品質管理を考慮した入出力 (時間情報の管理)

- アプリケーション間でのデバイス利用の競合調整
- デバイスの物理的な性質に依存したデバイスドライバのインタフェイス

そこでPDE-IIでは、知覚メディアを扱うデバイスに対する入出力を行うエンティティとして仮想デバイスエンティティを用意して、これらの問題を仮想デバイスエンティティ内の機構で解決する。従ってPDE-II上に実装されたアプリケーションでは、表示メディアの入出力を仮想デバイスエンティティ経由で行うことにより、これらの問題を解決することが可能となる。

また、必要に応じて自由に作成することができるエンティティを仮想デバイスとして扱うため、新たに高機能なデバイスドライバをPDE-IIに容易に追加することが可能である。例えば、本稿で紹介するCosaicの実装では音を扱うオーディオエンティティに加えて、MPEG-I形式の映像情報を扱うためのデバイスとしてMPEGエンティティを新たに用意している。

### 3.3 アプリケーション例

PDE-IIの機構を用いてアプリケーションを実装した場合の例として、図3にマイクから拾った音をそのまま再生するツールを示す。

このアプリケーションをPDE-IIの上で実現する場合、アプリケーションは、音の入出力を行うオーディオエンティティと、ユーザの指定に基づいてオーディオエンティティから録音したデータをもって再生するプレイヤエンティティの2つのエンティティで構成される。プレイヤエンティティとオーディオエンティティとの間のデータおよび制御情報のやり取りには、PDE-IIのコネクション機構を用いる。従って、従来のアプリケーション作成で問題となっていた品質制御機構の実装と、エンティティ間の制御情報伝達の実装はPDE-IIの下位層の制御に依存することにより解決する。

ここで示した例は、1つの表示メディアのデータを加工整形なしに録音再生する単純なアプリケーションであるが、PDE-IIの有効性評価にはこれよりも複雑かつ実用的なアプリケーションを実装して評価を行う必要がある。そこで現在我々はPDE-IIの評価のためのアプリケーションとして、情報提供システム

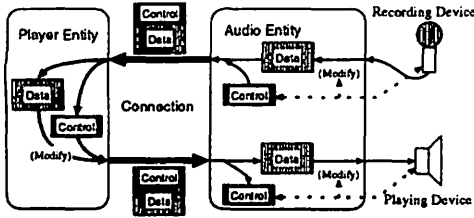


図 3: PDE-II のアプリケーション例

“Cosaic”の実装を、PDE-II アーキテクチャの実装と並行して進めている。

## 4 情報提供システム “Cosaic”

近年インターネットでは、複数メディアからなる情報とそれらの関連づけの情報によって構成されるハイパーテキストを利用した広域分散型情報提供サービス World Wide Web(WWW)[1]が、新たな情報発信/収集の手段として急速に普及しつつある。我々が実装を進めている情報提供システム Cosaic は、WWW サーバの情報提供サービスとして、複数の表示メディアを取り寄せながら表示する機能を新たに可能とするためのシステムである。本節では、Cosaic 開発の背景とその概要および実装について述べる。

### 4.1 背景

WWW 機構は、情報を提供する WWW サーバとサーバから情報を引き出して表示する WWW ブラウザから構成される。WWW サーバおよびブラウザの実装は、様々な計算機環境に応じて数多く行われているが、代表的な WWW サーバとして、CERN(Conseil European pour la Recherche Nucleaire)が実装したサーバと NCSA(National Center for Supercomputing Applications)が実装したサーバをの2つを挙げる事ができる [9]。また、WWW ブラウザとしては NCSA の開発した “Mosaic” が広く普及している [10]。

一般に WWW サーバでは、各種情報の種類とその所在および関連を示したポインタを含んだハイパーテキストを情報提供のためのインデックスとして用意している。ユーザは、WWW ブラウザを用いてサーバから引き出したインデックスから情報の関連を自ら辿っていくことによって目的のデータを入手する。

WWW で用いるハイパーテキストの書

式は HTML(HyperText Markup Language)として規定されている [11]。HTML 中で情報の所在を示すために用いるポインタはアンカー (anchor) と呼ばれ、URL(Uniform Resource Locators)[12]として規定された書式に従って記述する。URL は、HTML だけではなくインターネット上に存在するデータの場所とその入手方式を示す一般的な書式として利用されている。またサーバクライアント間のデータ通信プロトコルには、HTTP(HyperText Transfer Protocol)として規定されたプロトコルが用いられる [13]。

HTML および HTTP ではテキストだけではなく音や画像を扱えるので、WWW 機構では音や画像を利用した情報提供が可能である。しかしながら既存の WWW サーバ/ブラウザには以下のような問題点があると我々は考えている。

1. サーバが提供する情報はディスクに蓄積された情報に限定される。
2. ブラウザはサーバに要求した全ての情報を受信するまで表示処理を始めない。
3. 複数の HTTP コネクションを同時に作成して並行に通信を行うための指定方法が HTML では規定されていない。

問題 1 および 2 は、WWW 機構でやりとりするメディアに関する制限の問題である。既存の WWW 機構ではこれらの問題があるため、表示メディアから連続して取り込んだ表現メディアを即座に送受する非蓄積型の情報提供サービスが実現できない。

また問題 3 は、複数のメディア間の関連付けに関する制限の問題である。この制限により、既存の WWW 機構では複数のメディア間に関連をもたせて同時にブラウザ側で表示させるような処理ができない。

そこで我々はこれらの問題を解決するために、既存の WWW 機構の機能に加えて以下に示す 3 つの機能を拡張した情報提供システム “Cosaic” の設計および実装を行っている。

1. 情報を受信しながら処理する。
2. 複数のコネクションを同時に生成して通信を行う。
3. 複数のコネクションを関連づけながら処理を行う。

Cosaic では以上の機能を利用することによって、蓄積された情報を提供するだけでなく、相互に関連付けられた複数の非蓄積型の情報を提供することが可能となる。

## 4.2 Cosaic システムの概要

現在の WWW ブラウザでは、HTML 中の URL やユーザが入力した URL に基づいて目標のサーバから新たなデータを全て受信した後に表示を行っている。入手した情報の表示は通常ブラウザ自体で行われるが、知覚メディアを計算機で処理する際に使用される表示メディアには様々な形式があるため、WWW ブラウザだけではそれらすべてに対応する事が困難である。そこで既存の WWW ブラウザでは入手した表示メディアの種類に応じて外部の表示メディア処理プロセスに表示処理を依頼することが可能である。

Cosaic の基本的な概念は、本来 WWW ブラウザで行うべき接続処理を、表示メディアの処理を外部プロセスへ依頼する機構を用いて外部のアプリケーションに依頼させて、外部アプリケーション側で実装すれば、既存の WWW 機構にあまり手を加えることなく容易に機能の拡張が実現できる、という発想に基づいている。

図 4 に Cosaic と既存の WWW 機構との関係と処理の流れを示す。ブラウザ側で Cosaic に関する設定情報が含まれたスクリプトファイルである "Cosaic Script" (図中 foo.cosaic) へのアンカーが次のデータ入先に指定された場合、まずブラウザは HTTP を用いてサーバ中の Cosaic Script 取り寄せる (1)(2)。次にスクリプトの処理を行うプロセス (図中 Cosaic Client) を起動して Cosaic Script の処理を依頼する (3)。起動された Cosaic Client は、Cosaic Script 中の指定に基づいてサーバとの間に接続を生成した後 (4) データを入手して再生する (5)。

図中の Cosaic Client は WWW ブラウザとは完全に独立したプロセスなので、WWW サーバとの間で HTTP ないしは他のプロトコルに基づいた接続を複数本任意に張れるように実装してもブラウザには影響がない。また、Cosaic Client での接続の張りかたや同期の状態を決定し、Cosaic 起動のトリガでもある Cosaic スクリプトファイルは、WWW 機構で用いられる通常のハイパーテキストの中からアンカーで関連付けることができるので既存の WWW 機構との

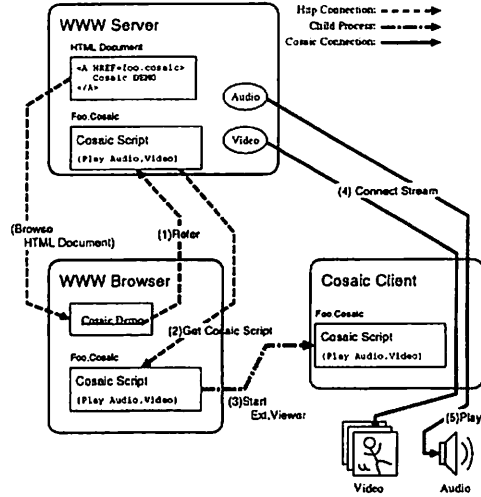


図 4: Cosaic システムの概要

親和性も高い。

## 4.3 Cosaic の実装

我々は、PDE-II および Cosaic の実装を図 4 の階層構造に応じて以下の 3 つの段階に大きく分けて進めている。

**第 1 段階** ブラウザ/サーバの外部への処理依頼に関する設定ファイルの修正を行い、接続に HTTP(TCP) を使用した Cosaic の実装を行う。

**第 2 段階** サーバと Cosaic の間の接続に HTTP だけではなく PDE-II の接続機構をも利用可能な Cosaic Client を実装する。この段階では Cosaic Client とオーディオデバイスおよびビデオデバイスはすべてエンティティとして実装する。またサーバ側でデータを送信するためのエンティティを WWW サーバとは別に用意する。

**第 3 段階** PDE-II の下位レベルコミュニケーション層の実装が終了した PDE-II/OS 上で第 2 段階の Cosaic を走らせて Cosaic で提供するサービスの品質向上を評価する。

第 2 段階以降では、WWW サーバ側にも Cosaic 独自の情報提供機構を用意するので、サーバ側のプロセスが提供するデータとして

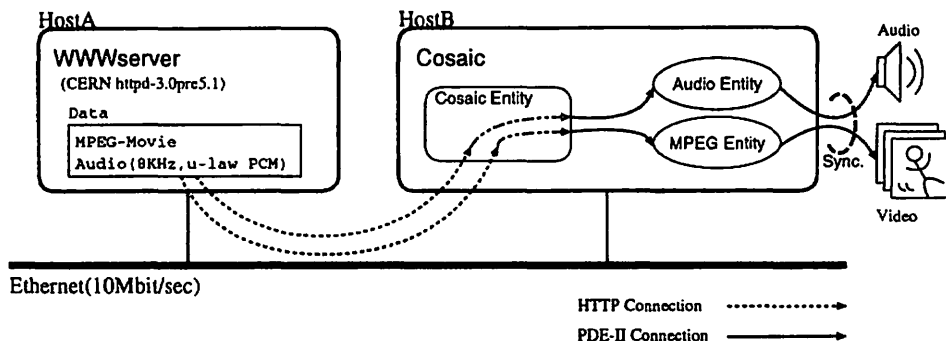


図 5: Cosaic の実験環境

音や動画といった非蓄積型の情報を想定すれば連続メディアの生中継が Cosaic によって実現できる。PDE-II のエンティティ層同期機構は、デバイスエンティティの実装と併せて実装を行う。またコネクション層の同期機構は、第 2 段階以降でエンティティインタフェースのから呼び出せるように組み込んで Cosaic で利用することを検討している。

## 5 Cosaic における同期実験

我々は PDE-II の各機構と Cosaic の実装を並行して進めている。前述した Cosaic 実装計画の第 1 段階は既に終了しており、現在は第 2 段階のデバイスエンティティとその内部でエンティティ層同期を実現するための機構の実装と、エンティティやコネクションを管理するエンティティ管理機構の実装を行っている。本節では第 1 段階の表現メディアの表示部分を、エンティティ制御メッセージ中の時間情報からエンティティ層同期を行う拡張論理時間同期機構を実装したデバイスエンティティで置き換えた Cosaic 第 1 段階改良版を使用して、MPEG-1 映像と音の同期出力実験を行った結果を示し、その評価および考察を行う。

### 5.1 同期制御実験

同期制御実験を行った環境を図 5 に示す。実験は Ethernet で接続された 2 台の計算機でそれぞれ WWW サーバと Cosaic とデバイスエンティティを起動して行った。実験にはオペレーティングシステム OSF/1 ver 2.0 を搭載した Digital Equipment 社の DEC 3000/300 (CPU Alpha21064 150MHz, Memory 64Mbyte) を使用した。ま

た、WWW サーバには CERN httpd version 3.0pre5.1 を使用した。実験で使用した音および映像の表現メディアには、音声はサンプリングレート 8KHz、量子化形式  $\mu$ -law 8bit の PCM データ、映像は縦横 160×128 ピクセルの MPEG-1 規格のデータをそれぞれ使用した。

実験では、Cosaic エンティティは音と映像の情報を HTTP を用いて WWW サーバから並行に取り寄せながら PDE-II のコネクション機構を用いてオーディオエンティティと MPEG エンティティに中継する。このとき、MPEG エンティティの画像出力にはオーディオエンティティの再生に従って同期させる指示を PDE-II インタフェースを用いて与えた。この同期機構では、同期のマスとなる音の再生に対して動画の再生が進みすぎた場合は動画の再生を一時停止して動画の再生時間を補正する。また動画の再生が遅れすぎた場合は、音の再生に追い付くまで画像表示処理をスキップして動画の再生を早送りする。実験では動画のずれの許容範囲を  $\pm 100msec$  として、許容範囲を越えた場合のみ同期処理を行うように設定した。

### 5.2 実験結果および考察

前述の実験環境で、2 種類の音/動画情報について動画の 1 秒当たりの表示フレーム数を変化させた場合の実験結果を表 2 に示す。

表 2 の FPS 計測値より、動画の 1 秒当たりの処理フレーム数の増加によって計算機の負荷が上昇した場合でも、フレームのスキップをすることによって同期保ちながら処理が行われていることが確認できる。また、同期のずれは指定した許容範囲  $\pm 100msec$  内には



表 2: 同期実験の結果

FPS (frame/sec)		9	10	11	12
音/動画像 A	FPS 計測値 (frame/sec)	8.996	9.984	11.091	11.977
	スキップ率 (%)	0.0	4.8	4.8	19.0
	同期のずれ (msec)	-82.29	-71.32	-35.83	53.13
音/動画像 B	FPS 計測値 (frame/sec)	8.995	9.989	11.070	11.957
	スキップ率 (%)	0.0	1.7	27.1	65.9
	同期のずれ (msec)	-81.37	-48.44	85.82	117.27

FPS: 1秒当たりのフレーム数の設定 (Frame Per Second)

FPS 計測値: システムクロックで計測した実際のフレーム処理間隔  
(スキップしたフレームもフレーム数に含む)

スキップ率: 同期制御によりスキップされた画像の割合

同期のずれ: マスタ (音) の再生時間とスレーブ (動画像) の再生時間のずれの平均

は収まっていることが確認できる。

ここで、音/動画像 B を 12 frame/sec で再生した時の同期のずれが  $\pm 100\text{msec}$  を越えている理由は、MPEG エンティティでのフレームスキップの実装方法にある。MPEG エンティティでは、フレームスキップによる動画像の早送りを高速に行うために、MPEG データから復号した 1 フレーム分の画像データを色変換処理して X ウィンドウ上に表示する直前で同期の判定を行わずに、色変換処理を実行する前で行っている。このような実装によって、フレームをスキップする時には表示処理だけではなく色変換処理をも省略することになるので、スキップ判定から次のフレームの復号処理までの余分な処理を完全に省略することが可能である。

しかしながら、通常の表示の際には、同期処理をしたあと出力が終るまでに色変換処理が実行されるため画像の表示に遅延が生じる。このため実際に画像を出力した時には同期のずれの許容範囲を越えてしまう場合がある。音/動画像 B の 12 frame/sec における同期のずれが許容範囲の 100msec を越えたのは頻繁にフレームがスキップするほど処理の負荷が高くなったために、色変換処理による誤差が顕著に現れた結果である。

また今回の実験では、動画像の処理が極端に遅れた場合にスキップ処理を繰り返しても音の再生に追い付かないため再生画像が停止してしまうという現象が生じることが確認された。この現象は、今回実装した同期制御機

構ではスレーブ側の画像の再生が遅れ続けても、マスタ側の音の再生は止まらずに再生を続けるようなアルゴリズムを採用しているために生じている。

以上の考察から、より実用的に利用できる Cosaic の実現には、エンティティ層同期機構におけるアルゴリズムの実装をユーザの幅広い要求に答えられるように改良して、PDE-II のアプリケーション側で利用できるように改善する必要があると我々は考えている。

## 6 まとめと今後の課題

本稿では、テキスト、音、映像など複数の知覚メディアを並列に処理するマルチメディア処理を行うための計算機環境として我々が提唱している計算機環境 PDE-II の概念と、マルチメディア処理アプリケーション作成のために PDE-II で用意したアプリケーションインタフェイスの概要とその利用法について述べた。また、PDE-II の評価用アプリケーションとして我々が同時に実装を進めている情報提供ツール Cosaic と、Cosaic における PDE-II インタフェイスを用いた同期出力実験の結果を紹介して、PDE-II のアプリケーションインタフェイスの有効性を示した。

今後は、今回の実験で問題点が判明したエンティティ同期機構に関する改良と現在第 2 段階途中の Cosaic の実装を進めて WWW 機構による相互に関連付けられた蓄積情報/非蓄積情報の提供実験と評価を行いたいと我々は

考えている。特に今回の実験で明らかになった、スレーブ側の再生が遅れ過ぎて表示をスキップし続けてもマスタに追い付けなくなる問題に関しては、下位レベルコミュニケーション層にデータの一部を選択して自動的に淘汰する機能を実装して解決することを検討している [14]。これは第3段階の Cosaic の実装に関する課題である。

また、本稿では Cosaic での処理手順を定める Cosaic スクリプトの仕様に関しては特に触れていない。我々は Cosaic スクリプトを並列処理記述言語として位置付けており、汎用性の高い記述が可能な言語仕様を検討中である。Cosaic スクリプトの仕様の決定とその実装および評価もまた本研究の課題である。

## 謝辞

本研究を行うに当たって、日頃から熱心に御指導、御助力を頂く奈良先端科学技術大学院大学 荒木研究室および福田研究室の諸先生方および学生の皆様に感謝致します。また同期実験のための動画像作成に当たり、画像データを快く提供して下さい九州大学工学部情報工学科牧之内研究室の吉川克彦氏に感謝致します。

## 参考文献

- [1] T.J. Berners-Lee, R. Cailliau, J-F Groff, B. Pollermann, CERN, "World-Wide Web The Information Universe", *Electronic Networking: Research, Applications and Policy*, Vol. 2 No 1, pp. 52-58 Spring 1992, Meckler Publishing, Westport, CT, USA.
- [2] Network Working Group, "Host Extensions for IP Multicasting", RFC 1054, May 1988.
- [3] Stephen Casner and Stephen Deering, "First IETF Internet Audiocast", *ACM SIGCOMM Computer Communication Review*, pp.92-97, Jul. 1992.
- [4] 海谷, 中山, 石田, "研究会会場でのインターネット利用の試み", 情報処理学会全国大会, 7C-2, pp.1-191-192, Mar. 1994.
- [5] H.Tanaka, K.Okamura, M.Hirabaru, K.Araki, "Voice Conference Tool across The Internet", 8th International Joint Workshop on Computer Communication, Dec. 1993.
- [6] 岡村, 吉川, 稲垣, 荒木, "PDE-II の概要 ~QoS に基づいたマルチメディア処理モデル~, 情報処理学会全国大会, 1H-5, Mar. 1994.
- [7] 岡村, 吉川, 稲垣, 荒木, "QoS 指定可能なマルチメディアモデルの提案", 情報処理学会マルチメディア通信と分散処理, Nov. 1993.
- [8] K.Okamura, K.Araki, "Distributed Multimedia Synchronization by Extended Logical Time", The 9th International Conference on Information Networking, Dec. 1994.
- [9] "World Wide Web Server Software", Available electronically at <http://info.cern.ch/hypertext/WWW/Daemon/Overview.html>
- [10] Marc Andreessen, "NCSA Mosaic Technical Summary", Available electronically as mosaic.ps.Z at <ftp://ftp.ncsa.uiuc.edu/Web/mosaic-papers>, Nov. 1993.
- [11] Tim Berners-Lee and Dan Connolly, "HyperText Markup Language", Available electronically at <http://info.cern.ch/hypertext/WWW/MarkUp/HTML.html>
- [12] Tim Berners-Lee, "Uniform Resource Locators", Available electronically as draft-ietf-uri-url-03.ps at <http://info.cern.ch/hypertext/WWW/Addressing/URL/url-spec.ps>, Mar. 1994.
- [13] Tim Berners-Lee, "Hypertext Transfer Protocol: A stateless Search, Retrieve and Manipulation Protocol", "Internet Draft", Available electronically as draft-ietf-iiir-http-00.ps, Nov. 1993.
- [14] 稲垣, 岡村, 荒木, "QoS に基づいた連続メディアデータの淘汰機構", 情報処理学会マルチメディア通信と分散処理ワークショップ, Oct. 1994.