

QoS 指定可能なマルチメディアモデルの提案

岡村耕二 吉川耕平 稲垣英太郎 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

我々は、QOS(Quality of Service)に基づいたマルチメディア処理機能を OS に取り込む研究を進めている。本稿では、マルチメディア処理において保証すべき QOS、QOS に基づいたマルチメディアの処理モデル、モデルに基づくマルチメディア処理モジュールについて述べる。本稿では時間的制約に加えて、複数メディアの同期とその優先度を QOS として採り入れたことを特徴としている。

1 はじめに

マルチメディア処理および通信は、その計算量や通信量が膨大であるため、従来は高価で専用のハードウェアが不可欠であった。しかし、近年のハードウェアの設計・製造技術の進歩による計算機の処理能力の飛躍的な向上により、マルチメディア処理および通信を安価なハードウェア上で行なうことが可能となってきている。さらに今後のハードウェアの処理能力の向上により、より高品質なマルチメディア - 滑らかに躍動する美しい動画や臨場感溢れる音声など - を机上の計算機で利用できることが期待されている。

しかしながら、UNIX に代表される現在の多くのオペレーティングシステム(以下 OS と表す)による時間的な保証のないスケジューリング方式や資源管理方式を用いた処理では、マルチメディアに特有である時間的な制約をシステムが保証することができない。また、ハードウェアの高速化により単位時間当たりのメディアの処理量が増加しても、それだけでは時間的な制約の保証はできない。

そこで我々は、OS のスケジューリング方式や資源管理方式に、時間的な制約のある処理に

必要な機能を付加することで、処理するマルチメディアの品質を保証することが可能になると考え、現在、OS にその機能を付加する研究を行なっている。マルチメディア処理やその同期に関しては文献 [1][2] 等の報告があるが、我々は第一段階として、時間的な制約のあるマルチメディア処理を、Quality of Service (以下 QOS と表す) の概念に基づいてモデル化した。従来のマルチメディア処理における QOS は、メディア毎の解像度が主であるが [3]、我々はメディア毎の解像度の保証だけではマルチメディア処理における複数メディア間の処理の品質を保証することができないことに着目し、解像度に加えて、メディア間同期やメディア間優先度を QOS に導入した。本稿では、新たに導入したメディア間同期やメディア間優先度の品質を定義することによって保証すべき QOS を明確にし、その QOS に基づいたマルチメディア処理のモデル化を行なっている。

本稿の構成は次の通りである。第2章でまず QOS の定義を行なう。次に第3章でその QOS に基づいたマルチメディア処理モデルを提示し、その説明を行なう。第4章ではモデルの実装のための具体的な QOS の指定方法、同期処理方法およびマルチメディアモジュールの概説と、実装に関する考察を行なう。最後に第5章で全体のまとめを行なう。

"A Model for Multimedia Processing and Application based on QoS",
Koji OKAMURA, Kouhei YOSHIKAWA, Eitaro INAGAKI and Keijiro ARAKI
Graduate School of Information Science,
Nara Institute of Science and Technology

2 マルチメディアの品質

音声や映像といったメディアは、データが時間的制約を持ち、かつ時間軸上で途切れることなく連続して定義されているという特徴を持つ。さらにこれらのメディアが相互に関連づけられている複数メディアをマルチメディアと呼ぶ。そのマルチメディアを扱うプログラムをマルチメディアアプリケーションと呼ぶ。本稿で想定している典型的なマルチメディアアプリケーションを図1に示す。例えば、この図においてユーザは Mixer に「CD の品質で提供されるマイクの音声」や「カメラの映像とマイクの音声の同期」といった、品質を付加したマルチメディア処理要求を出す。このようなマルチメディア処理に付加された品質は、システムが保証できる QOS の範囲で保証することができる。本章では、システムが保証する QOS の品質の定義を行なう。

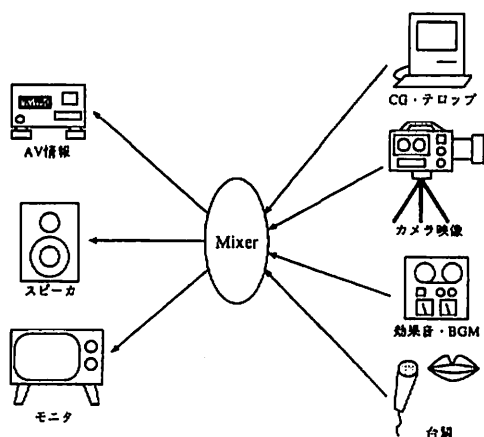


図 1: アフレコシステム

映像や音声などのメディアデータの伝送経路をコネクションと呼ぶ。我々は、各コネクションを流れるデータの量とその周期、コネクション間の同期、コネクションの優先度の品質を、システムが QOS として保証する品質として定義した。単一コネクションの品質は従来から定義されている QOS の代表的なものであるが、同期と優先度

は我々が新たに導入したものである。単一コネクションの品質はハードウェアの QOS に直接的に依存しているのに対して、同期や優先度は、ハードウェアではなく、アプリケーションに依存している。各々の品質の定義を以下に示す。

(1) 単一コネクションの品質

単一コネクションの品質は、時間的解像度と空間的解像度で表現することができる。例えば、時間的解像度とは動画の単位時間当りの駒数、音声のサンプリング周波数であり、空間的解像度とは画像の色数やピクセル数、音声の量子化ビット数やチャンネル数である。どのような種類のコネクションの品質も、時間的解像度と空間的解像度で表現することができるため、我々は、単一コネクションの品質を、周期時間と周期時間内に処理すべきデータ量の組で定義した。また、システムの動的な負荷の変動を考慮して、単一コネクションの品質は最高品質と最低品質を指定し、その幅の品質が提供されていれば、そのコネクションの品質が保証されていると定義する。

例えば、ある単一コネクションと、そのコネクションに対して周期および周期時間内で処理すべきデータ量が共に 10 倍である単一コネクションは、いずれも、単位当りに処理するデータ量は等しいが、品質は異なる。前者は短い周期でデッドラインを守りながらデータを処理しているコネクションであり、後者は長い周期で大量のデータを処理しているコネクションである。

(2) コネクション間同期の品質

図2は、カメラとマイクのコネクション間で同期をとっている例である。このように異なるコネクションを通るデータに時間的制約を付加することによって同期をとることをコネクション間同期をとるといふ。

同期を取る単位は、単一メディアの品質における時間的解像度の単位、すなわち、一周期である。同期機構を定量的に表現するため

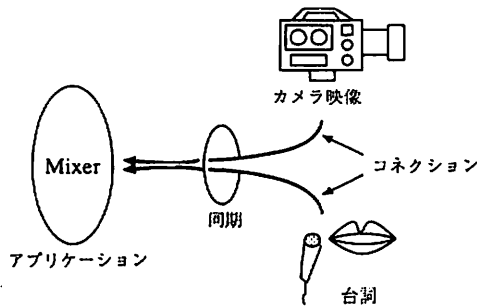


図 2: 音声とカメラ映像の同期

に、一周期の処理が終る度に一周期分の時間だけ進む論理時間を用いる [4]。図 3 のグラフの横軸は実時間軸であり、縦軸は論理時間軸である。図の中で、正方形のブロックは一周期当りに行なうデータの処理単位を表しており、ブロックの一边はその処理時間の長さを表している。図 3 を用いて、同期の品質の定義を述べる。

完全同期状態

図 3(a) は、実時間と論理時間の間に遅延が全く存在しない完全同期状態を示している。

同期状態

単一の接続であっても、システムの動的な負荷の変動によって論理時間と実時間の間にずれが生じる可能性がある。その結果、例えば図 3(b) のように同期の乱れが生じる。そこで、我々は、この論理時間に対する実時間のずれを同期の品質と定義し、指定された値以内のずれであれば、同期の品質が保証されていると定義する。

(3) 接続の優先度の品質

我々は、1 つのエンティティ内の複数の接続の間に優先度を設けている。各接続の品質を保証するために行なわれるリソースの調停や、同期の補正方法の選択は、接続の優先度に基づいて決められる。現在考えている同期の補正方式に

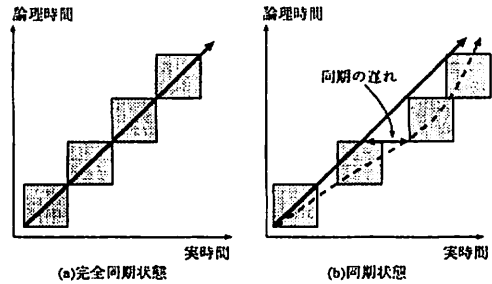


図 3: 同期状態

については第 4 章で詳しく述べる。なお、異なるエンティティにわたる接続間の優先度は考えない。

さて、アプリケーション間のリソースに対する排他制御メカニズムにより生じる優先度逆転現象のように、必ずしも高い優先度の接続が優先的に処理されているとは限らない。そこで我々は、全処理時間に占める優先度逆転の時間の割合を優先度の品質と定義し、優先度逆転現象がその割合内で発生しているならば、優先度の品質が保証されていると定義する。

以上、定義した品質の具体的な使用例について述べる。

(1) 単一接続の品質

単一接続の品質は映像、音声など、個々のメディアの品質そのものを規定するものである。マルチメディアデータベースの検索、映像や音声を含むマルチメディアドキュメントの編集、Tele-Conference などの応用目的ごとに、必要とする品質をきめ細かく設定できる。

(2) 接続間同期の品質

メディア間同期の品質は、リップシンクと呼ばれる口の動きと声の同期に代表される比較的厳密な同期を規定するもので、Tele-Conference 等の応用には必須である。

(3) コネクションの優先度の品質

例えば、Tele-Conference においては、たとえ映像が停止しても音声を途切れずに再生したい、というような要求に答えるものである。

3 マルチメディア処理モデル

我々が提案するマルチメディア処理モデルを図4に示す。図4は、エンティティ間に張られる仮想伝送路(コネクション)を通して多数のエンティティがマルチメディアデータの交換を行なっている様子を表している。

図中、円筒の芯の部分は、CPU タイムやメモリ、ネットワークバンド幅などエンティティが使用できるシステムリソースを、また円筒を貫くエンティティ間のリンクはコネクションを示しており、コネクションの設定がリソースの消費を伴うことを表している。一方、円筒の表面部分とコネクションの交わりは、同期をとるためのポイントを示している。コネクションはエンティティからの出口と入口の2つの同期ポイントを持つ。

マルチメディア処理モデルでは、任意のエンティティは任意のエンティティに対してコネクションを張ることが可能であり、任意の方向にデータ通信を行なうことができる。そのため、図4では、システムリソースを円筒で表現し、その周囲の任意の場所にエンティティを配置する。

(1) エンティティ

プロセスやデバイスなどマルチメディアデータの通信および処理を行なう主体である。データの入出力口であるエンドポイントを任意個持つことができる。

(2) システムリソース

エンティティ間で通信を行なうために必要なシステムのリソースを抽象化したもので、それを共通に使用するエンティティの集合に対して一つ存在するものとする。システムリソースは有限で、個々のコネクションはその設定に必要な品質から計算できる量を消費する。図4において、システムリソースの高さはシステムの全処理能

力を表し、斜線部分は既に確保されている部分を表す。

(3) エンドポイント

エンティティのデータの出入口であり、そこを流れるデータの品質(時間的解像度と空間的解像度)を設定できる。

(4) コネクション

エンティティ間でデータの交換を行なうための仮想伝送路であり、送信/受信両方のエンドポイント及び同期ポイント、後に述べる優先度を指定して生成する。生成の際、両ポイントに設定されたデータの品質を保証するために必要なリソース量をシステムリソース内に確保する。確保できなければ生成に失敗する。

(5) 同期ポイント

単一コネクション上でのデータ到着の時間的なずれの補正と、コネクション間の同期を行なう。保証する同期の品質を設定して生成する。個々のコネクションについて入力/出力の両方で独立に同期を設定できるものとし、また各々の同期ポイントでは、そこを通る複数のコネクションの優先度に従って同期の品質の保証を行なう。

システムリソースの定義

システムリソースは、モデル上、任意のエンティティの集合に対して一つずつ存在するものとする(図5)。ただし、一台のホスト中のエンティティすべてによってその集合を構成する必要はない。例えば一台のホストの中に複数のシステムリソースを定義してもよく、またエンティティがLAN等で接続された複数ホストに分散していてもよい。具体的なリソース定量化の手法については現在検討中である。

コネクション間の優先度

コネクション間の優先度は、エンティティ内でローカルに設定され、複数のエンティティに渡るものではない。同期ポイントでは、各コネクションに設定された優先度をもとに優先順位を決定し、同期の品質の保証を行なう。

システムリソースの共有

本モデルでは、図4の中で、エンティティBとC、エンティティDとCの間の二つのコネクショ

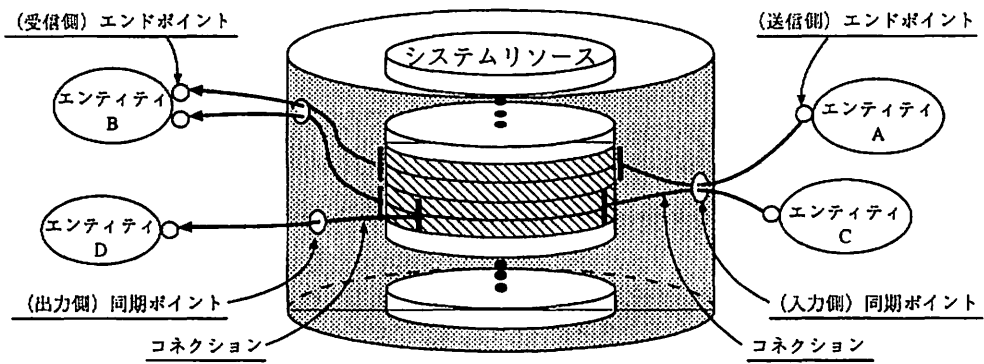


図 4: マルチメディア処理モデル

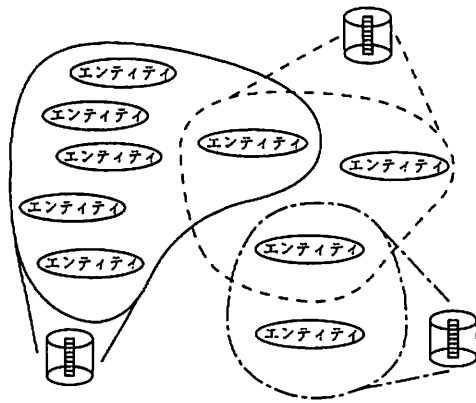


図 5: システムリソースの定義

ンに示すように、複数コネクションによるシステムリソースの共有を許している。これによりブロードキャストの概念を自然に表現している。

遠隔エンティティ間の通信

共通のシステムリソースがないエンティティ間では、中継機能を持つ他のエンティティを介して通信を行なう。図6では、この通信の様子を中継のために使用しているエンティティを破線で囲んで示している。

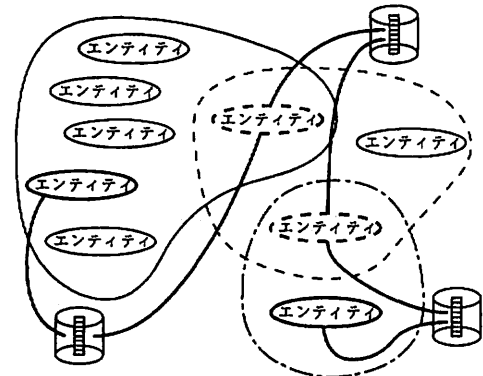


図 6: 遠隔エンティティ間の通信

4 マルチメディアモジュール

本章では、QOS の具体的な指定方法、同期の補正処理、現在設計中のマルチメディアモジュールのモジュール構成と各構成要素の機能、実装において問題となる点について考察を行なう。

4.1 QOS の指定方法

我々は、QOS として保証すべき品質として、(1) 単一コネクシオンの品質、(2) コネクシオン間同期の品質、(3) コネクシオンの優先度の品質を定義した。各々の品質の具体的な指定方法について述べる。

(1) 単一コネクシオンの品質

現在計算機上では、音声や動画などのメディアが利用可能であるが、今後その他様々なメディアが利用可能になることも考えられる。しかしながら、どのようなメディアのコネクシオンの品質も時間的解像度と空間的解像度で表現できるため、各コネクシオンの品質は一周期の時間と、一周期の時間内に処理すべきデータ量で指定することができる。

一周期の時間

このパラメータは時間的解像度の値であり、時間の単位で指定する。

一周期の時間内に処理すべきデータ量

このパラメータは空間的解像度の値であり、最上品質および保証すべき最低品質ともにデータ量はバイト単位で指定する。実行時の変更も可能とする。

(2) コネクシオン間同期の品質

コネクシオン間同期の品質は、同期のずれの許容範囲を時間の単位で指定する。この指定は、同期ポイントを生成する時に行なう。実行時の変更も可能とする。

(3) コネクシオンの優先度の品質

コネクシオン間の優先度の品質は、全処理時間あたりで許容される優先度逆転状態の割合で指定する。

優先度の品質は、エンドポイントの設定時に指定する。実行時の変更も可能とする。

現在の品質は、連続的な実数で表されているため、実際の指定が困難であるという問題に対して、プログラマやユーザに対するインタフェースを検討中である。

4.2 同期の補正

ここでは、文献 [4] を参考にした、複数のコネクシオン間の同期の補正処理方法について述べる。同期の品質は、コネクシオンの優先度によって「遅延挿入」または「データスキップ」のいずれかの方法を用いた同期の補正により保証する。それぞれの方法を 図 7 を用いて説明する。

図 7 は、横軸が実時間で、縦軸が論理時間を表している。今、周期の短いコネクシオン (色の濃い正方形) が周期の長いコネクシオンよりも優先度が高いとする。

これらの周期の異なるコネクシオン間で周期が取れているということは、2 章で述べたように、それぞれのコネクシオンの実時間と論理時間の進み具合のずれが、与えられた品質の値をこえていないことであり、それぞれのコネクシオンの対応する番号の正方形同士が同期するわけではない。以下に、同期のずれが生じた場合の補正方法の説明を行なう。

遅延挿入

遅延挿入は、最も優先度の高いコネクシオンの同期の補正を行なう方法である。図 7 において、優先度の高いコネクシオンの 4 番目の処理が遅れた場合、優先度の低いコネクシオンは 3 番の処理を止めて優先度の高いコネクシオンに論理時間を合わせる。遅延挿入によって、優先度の低いコネクシオンの論理時間が一時的に止められる。

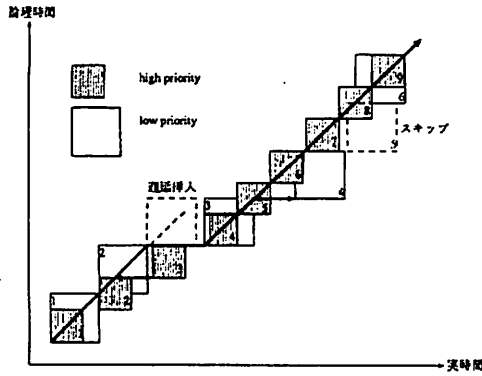


図 7: 同期の補正

データスキップ

データスキップは、優先度の低いコネクションの同期の補正を行なう方法である。図7において、優先度の低いコネクションの4番目の処理が遅れた場合、優先度の低いコネクションは、5番目の処理は放棄し、次の6番目の処理を行なうことで、優先度の高いコネクションと論理時間を合わせる。データスキップによって、優先度の低いコネクションの論理時間が一時的に進められる。

4.3 モジュール構成

図8に、我々が現在設計中のマルチメディアモジュールの構成図を示す。マルチメディアモジュールは、(1) コネクション管理モジュール、(2) エンティティ管理モジュール、(3) QOS管理モジュールおよび、(4) システムリソース管理モジュールで構成されており、図中、破線で囲まれた部分である。

以下、マルチメディアモジュールの各管理モジュールについて説明する。

(1) コネクション管理モジュール

コネクションのエンドポイントの管理を行なう。アプリケーションのメディアデバイスに対するアクセスはコネクションのエンド

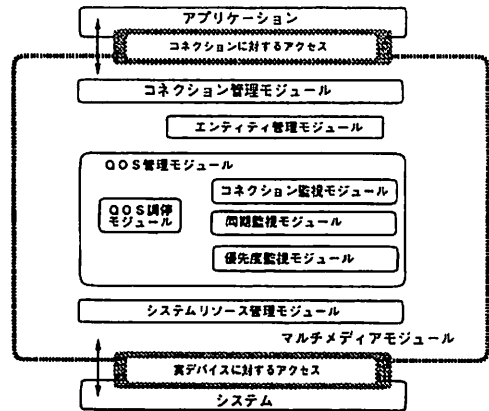


図 8: モジュール構成図

ポイントに対して行なうため、コネクション管理モジュールは、アプリケーションに、エンドポイントへの基本的な操作インタフェースを提供する。またコネクションの同期ポイントの管理も同様に行なう。

(2) エンティティ管理モジュール

静的なエンティティ(実デバイス)および、動的に生成、消滅するエンティティ(プロセス付きデバイス)の管理を行なう。分散して存在するエンティティの情報を、エンティティ管理モジュールで管理することにより、エンティティの分散透明化の支援を行なう。

(3) QOS管理モジュール

QOS管理モジュールには、マルチメディアモジュールが保証する品質毎にモジュールが存在し、それぞれのモジュールにおいて管理しているサービスの品質の監視を行なっている。現在、我々が保証している品質は、単一コネクションの品質、同期の品質、および優先度の品質であり、それぞれの監視モジュールが存在する。

負荷の動的な変動によりサービス品質の変更が必要な場合は、QOS調停モジュールがコネクション間でのサービスの品質を調停

しながら変更を行なう。

- (a) コネクション監視モジュール
各コネクションの品質の QOS が満たされているかどうか監視する。
- (b) 同期監視モジュール
同期を取っているコネクション間の同期の QOS が満たされているかどうか監視する。
- (c) 優先度監視モジュール
各コネクション間の優先度の QOS が満たされているかどうか監視する。
- (d) QOS 調停モジュール
各監視モジュールからの情報と、次に述べるシステムリソース管理モジュールからの情報を基にして、システム全体で各コネクションが適切なサービスの品質で処理されることを保証するように、コネクション間でのサービスの調停を行なう。

- (4) システムリソース管理モジュール
システムのリソースに関する情報を集め、QOS 管理 モジュールに提供する。

4.4 実装に関する考察

我々の提案したマルチメディアモジュールの実装には以下の3つの問題があり、現在検討中である。

リソースの定量化

システムリソース管理モジュールは、常にシステムのリソース情報を保持しておく必要がある。そのために、考慮すべきリソースを明確にし、そのリソース情報の定量化を行なう必要がある。

リソースのリザベーション

各コネクションの品質の保証のために、リソースのリザベーションを行なう必要がある。我々は特にネットワークのリザベ

ーションに重点をおき、ST-II[5] など既存のリザベーションを行なう通信プロトコルを基にして LAN における通信回線のリザベーションに関する研究を進めている。

また、ネットワーク以外の、例えば CPU パワーなどのリザベーションは、リソースのスケジューリングの共通的な問題として考察中である。

リソースのスケジューリング

コネクションの優先度の品質を保証するには、リソースのスケジューリング方式を見直す必要がある。我々は、現在提案されているスケジューリング方式のうち、リアルタイム処理に適した Rate Monotonic Algorithm[6]、優先度逆転を回避する Priority Inheritance Protocol [7] 及びデッドロックを回避する Priority Ceiling Protocol[8] を基に、それらアルゴリズムやプロトコルの問題点を解決しながらスケジューリング方式の改良を行なっている。

5 おわりに

我々が研究・開発を進めている 品質を付加したマルチメディア処理のための機能を OS に取り入れるために構築したマルチメディア処理モデルの提案を行ない、その実装に関する考察を行なった。

マルチメディア処理モデルのために新たに定義した同期や優先度の品質は、ハードウェアが提供する QOS に直接的には依存しないため、将来的な高性能なハードウェアを用いたシステムにおいても有用であると考えられる。

今後は、本稿で示した問題を解決しながら、我々の提案したマルチメディア処理モデルに基づいたマルチメディアモジュールの実装を行ない、評価を行なっていく予定である。

謝辞

本研究を行なうに当って、多くの貴重な御助言を頂いた奈良先端科学技術大学院大学 荒木研、福田研の皆様に感謝致します。

- [8] Tokuda, H. and Nakajima, T. : "Evaluation of real-time synchronization in Real-Time Mach," Proceedings of the USENIX Mach Symposium, 213-21, 262, 1991.

参考文献

- [1] 米田, 林, 松下 : "分散環境に適したマルチメディア制御モデル", 情報処理学会研究報告, マルチメディア通信と分散処理, 60-80, 1993.
- [2] 佐藤, 林, 岡田, 松下 : "論理式によるマルチメディア同期表現モデル", 情報処理学会研究報告, マルチメディア通信と分散処理, 55-7, 1992.
- [3] 船渡, 徳田 : "Real-Time Mach 3.0 における連続メディアサーバの実験—QOS 制御を取り入れた QuickTime Player の評価—", 情報処理学会研究報告, マルチメディア通信と分散処理, 61-11, 1993.
- [4] Anderson, D.P. and Homsy, G. : "A Continuous Media I/O Server and Its Synchronization Mechanism", IEEE, Computer, Oct. 1991.
- [5] 南部, 木原, 徳田, 塩野崎, 福嶋 : "マルチメディア統合環境におけるリアルタイム通信プロトコルへの課題", 情報処理学会研究報告, マルチメディアと分散処理, 60-11, 1993.
- [6] Weideman, N.H. and Kamenoff, N.I.: "Hartstone uniprocessor benchmark: definitions and experiments for real-time systems," Real-Time Syst. (Netherlands), vol.4, no.4, 353-82, Dec. 1992
- [7] 緒方, 和田, 追川, 西尾, 徳田: "Real-Time Mach 3.0 のマルチメディア処理に関する性能評価", 情報処理学会研究報告, マルチメディア通信と分散処理, 61-10, 1993.