

## MUMIN システムにおけるマルチメディア・モデル

富永 浩之 服部 隆志 柴山 悦哉

京都大学数理解析研究所

龍谷大学工学部

MUMIN システムは、UNIX ワークステーション上のマルチメディア・アプリケーション開発支援プラットフォームである。MUMIN のマルチメディア管理機能を設計するにあたり、実行環境としてのメディア・デバイス群、およびマルチメディア・データの統一的な記述を目的とするオブジェクト・モデルを導入した。このモデルでは、同期関係を含むマルチメディア・データの実行状態をデバイスに対する実際の制御と独立した形で表現できる。また、MUMIN のユーザインタフェース管理機能と整合し、インタラクティブなマルチメディア・ユーザインタフェース構築のための枠組も提示している。

## Multimedia model in the MUMIN System

Hiroyuki Tominaga Takashi Hattori Etsuya Shibayama

Research Institute for Mathematical Sciences, Kyoto University

Faculty of Science and Technology, Ryukoku University

We propose a new model which is designed to represent both multimedia data and their environments including devices for media. It will be introduced into the MUMIN system, a supporting system for development of multimedia applications on UNIX workstations. Abstract operations/relations (including abstract synchronization) of multimedia data can be described independently of their execution (play-time) environment. The model is fully compatible with the UIMS functions in the MUMIN system, to support development of interactive multimedia user-interface.

## 1 MUMIN システムの概要

MUMIN システム<sup>1</sup>は、UNIX ワークステーション上のアプリケーション開発を支援するプラットフォームである。MUMIN には、大まかに分けてユーザインタフェース管理システム (UIMS) およびマルチメディア管理システム (MMMS) の 2 つの側面があり、特にマルチメディア UIMS の実現を目的としている。

MUMIN の本体 (MUMIN プロセス) は、各ユーザに対して、X ウィンドウのクライアントとして 1 つだけ存在する。アプリケーション・プロセスは MUMIN プロセスのクライアントとなる。メディアを制御する各種の装置 (メディア・デバイス) は、それぞれ独立したサーバ・プロセス (メディア・サーバ) によって管理されており、アプリケーション・プロセスとメディア・サーバとの通信は MUMIN プロセスを介して行なわれる。なお、MUMIN 本体の拡張言語としては、Elk (Scheme インタプリタ) に並列オブジェクト指向機能を追加したものを採用している。

MUMIN の設計においては、オブジェクト指向パラダイムに沿った概念モデルを提示し、その実現に努めた。特に、UIMS 部分は、アプリケーションの UI 構成を表現するペリフェラル・モデル (服部 [10]) に基づいている。このモデルでは、1 つのアプリケーションを構成するオブジェクト群を機能の面から、本体モジュールをなすコア部と UI モジュールと考えられる幾つかのペリフェラル部に分割する。コア部とペリフェラル部の間には、両者にまたがるメッセージ通信を中継するエージェントを介在させる。エージェントを利用した間接アドレッシングにより、コア部とペリフェラル部は互いの変更による影響を受けずに情報交換を行なうことができる。

以下の本論では、主に MUMIN の MMMS 部分における基本的な設計方針と、そのベースとなる概念モデル (マルチメディア・モデル) について述べる。このモデルでは、ペリフェラル・モデルの延長上にマルチメディアを表現するための諸概念を導入している。

## 2 マルチメディア管理機能の設計目標

MUMIN の MMMS は、プラットフォーム上で作動する各種のマルチメディア・システムを統合的に管理する部分である。マルチメディア・システムとしては、メディア・デバイスなどブリミティブなリソースを提供するメディア・サーバから、ハイパーメディア・システムのような複雑なマルチメディア・データ構造を利用するアプリケーションまで様々なレベルが想定される。したがって、メディアを捉える抽象度に応じて幾つかの階層に分け、それぞれの抽象度に対応したデータ記述や、実行環境との関係の定義ができることが望ましいと思われる。

### 1. デバイス管理

マルチメディア環境を提供する外部プロセス (メディア・サーバ) とリソース (メディア・デバイス) を要求するクライアント (アプリケーション) の関係を扱う。まず、再生装置などリアルタイムかつ持続的な占有が必要な物理デバイスに対しては、排他制御を行なう必要がある。特に、ネットワーク上に分散するデバイスの利用に際しては、複数のユーザ (MUMIN) との競合が問題になる。

また、実際に使用可能なデバイスが実行時まで特定されない状況において、従来の方法でアプリケーションを開発するには、必要なデバイスを設定するための様々な処理を想定し

<sup>1</sup>京都大学、龍谷大学、オムロン (株)、オムロンソフトウェア (株)、住友電気工業 (株)、(株)PFU、富士ゼロックス (株)、京都高度技術研究所による共同研究で開発されている。

ておかなければならない。アプリケーション開発の効率を上げるには、デバイス依存の副次的な処理を分離させ、本体部分の設計に関しては開発環境と実行環境の差異を意識せずに行なえるようにしなければならない。また、本体に対するデバイス依存部分の独立性により、幾つかのアプリケーションに共通するデバイス依存処理を1つのツールとして活用させることも考えられる。

## 2. データ管理

従来のマルチメディア・システムにおいては、音声や画像などのプリミティブなデータに関しては互換性が考慮されても、それらを時間的、空間的に組み合わせた複合データについては、アプリケーションやオーサリング・システムごとに専用のデータ構造を設計することが多かった。しかし、ソフトウェアの部品化、再利用化を進めていくためには、アプリケーション間で交換するデータの共通な形式を定める必要がある。そのためのデータ形式においては、各種のデバイスによる固有の制御を必要とするデータをプリミティブなオブジェクトとみなし、その上に共通の同期関係記述を与えた複合オブジェクトを構成するのが自然である。

MMMSの機能としては、データベース上の静的なデータ管理の他に、動的なデータの実行管理が必要である。また、データにおける本体部分とデバイス依存部分の分離は、抽象的な同期関係をシミュレートするための部分と、プリミティブなオブジェクトにおけるデバイスとの接続処理および固有の制御情報の管理を行なう部分とに対応する。

## 3. メディア管理

拡張性のあるマルチメディア・システムを構築するためには、マルチメディア・データの共有だけでなく、さらにそれを利用するための手続きや環境まで含めた部品化、再利用化を行なわなければならない。そこで、マルチメディア・データ、条件分岐などの制御構造、ユーザインターフェース部品などを構成要素とするモジュール(これ自体で一つのアプリケーションとして動作する)を考え、それを単位として、より複雑なアプリケーションの構築ができるようにする。入出力装置の結合状態の管理など、デバイス間を関係も考慮したMMMSが必要にある。

メディア・レベルのモジュールは、ペリフェラル・モデルに従った構造を持つため、GUI生成ツールなどのMUMINのUIMS機能を、メディア・レベルに対応するように拡張することも容易であると考えられる。これにより、インタラクションを含むマルチメディア・ユーザインターフェースを対話的に生成できるようになり、また各種オーサリング・システム構築のための標準的の枠組としても利用できる。

## 3 マルチメディア・モデルの特徴

マルチメディア・モデルでは、MMMS機能を用いるためのインタフェース・レベルとして、デバイス、データ、メディアの3つを設ける。アプリケーションやユーザは、要求する実行環境の抽象度に応じて適切なレベルを用いることができる。各レベルでは、プリミティブとなるオブジェクト・クラスと、複数のプリミティブをまとめて扱うための構造概念(ストラクチャと呼ぶ)が定められている。ストラクチャを1つのアプリケーションとみると、構造を規定する本体(コア部)に対し、構成要素であるプリミティブ群がメディア依存の部分(ペリフェラル部)を表現する。各レベルでのMMMS機能は、ストラクチャをクライアントとし、プリミティブ部分

に実行環境との照会その他のサービスを提供する。また、プリミティブおよびストラクチャは、上位レベルからみると、そのレベルでのプリミティブに対するペリフェラル部(ポート部と呼ぶ)に相当すると考えられる。

### (1) デバイス・レベル

デバイス・レベルは、MMMS 機能のうち、排他制御や例外処理などのデバイス管理を主とする。アプリケーションが、直接デバイスを要求し、デバイス・マネージャがサービスを提供する。

#### ● 抽象デバイス

抽象デバイスは、メディア・サーバが管理するデバイスと対応し、排他制御の基本単位となる。また、アプリケーションとデバイスとの通信を仲介し、プロトコルの差異を吸収する。特に、デバイスからのエラー情報その他の例外メッセージを、アプリケーションには影響を与えずに適当なマネージャに送ることができ、デーモ的な役割が実現される (fig 1 の group manager)。また、ペリフェラル・モデルのエージェントと同様に、動的なデバイスの変更に対応することができる。

#### ● グループ・マネージャ

アプリケーションが複数のデバイスを同時に使用する場合、要求単位ごとにグループ・マネージャを生成して管理する。グループ・マネージャと抽象デバイスの参照関係で、アプリケーションによるデバイスの占有をシミュレートする。アプリケーションは、グループ・マネージャを通して、デバイスの追加使用と解放を行なう。

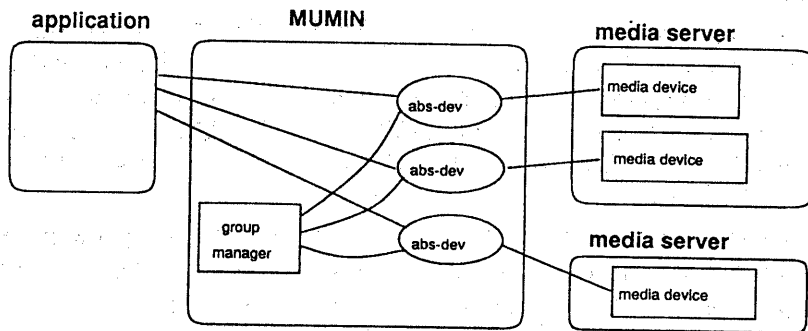


fig 1 : device level

### (2) データ・レベル

アプリケーションは、抽象データの実行を通して、間接的にデバイス制御をデータ・レベルは、MMMS 機能のうち、メディアのデータベース上の静的な記述および実行時の動的な状態の管理を行なう。

抽象データは、1つのマルチメディア・データとして機能するオブジェクト群の階層構造で表される。アプリケーションから見て1つのペリフェラル部に相当する。アプリケーションとUIの分離と同様、マルチメディア環境に依存する部分と独立した抽象的なメディア記述を目指している。さらに、イベント通信の機構や時間進行を表す内部状態を導入し、データ実行のシミュレートや、抽象的なメディア記述から実際のマルチメディア環境への反映を実現する。

- 素データ

素データは、データ・レベルにおけるプリミティブなインタフェースである。素データは、1つのデバイスで解釈される物理的な記録媒体や1つのデジタル・データに対応する。データを実際に解釈するメディア・デバイスの種類に応じて、CDやPCMなどの下位クラスに分かれる。素データには、デバイスに対する要求や自身の実行制御に関する制約を属性として与えることができる(4.1を参照)。

- 複合データ

複合データは、素データおよび同期関係データからなる階層構造で、基本的な同期を表現している。同期関係には、下位データ間の同期関係を表し、順次走行 serial、並列走行 parallel の2種類がある。

複合データにおけるデバイス依存部分は、素データに集約されており、複合データ自身はデバイスと独立に抽象的な同期実行を行なえる(4.2を参照)。複合データの実行を具体化する場合は、必要な複数の抽象デバイスをまとめて要求し、各素データと接続する。このレベルでのデバイス管理としては、データの同期情報も解析して、デバイスの占有を必要な期間のみに限定したり、また将来必要なデバイスを予約しておく機能が考えられる。また、抽象データに対しての例外処理では、制御エラーや割り込みなどが発生したとき、同期関係を一時的に保存しておいて、必要な処理を行ない、途中からの再実行を可能にする機能が必要である。

### (3) メディア・レベル

抽象メディアは、一般的なマルチメディア・データのオブジェクト表現に相当する。

- 基本メディア

基本メディアは、メディア・レベルでのプリミティブであり、それ自身で1つのアプリケーションとしての構造を持っている。コア部の記述により、複雑なデータ制御を行なうことができる。また、ペリフェラル部のうち、マルチメディアUIに相当する部分をメディア・ポート部として捉えたものである。基本メディアでは、結合リンクを用いて抽象データ間の入出力関係を表すことができる。同期関係データが時間的なモジュールを表すのに対し、結合リンクは(相対的な)空間配置を表している。また、結合リンクのサブクラスとして抽象機能やデータ変換を表現する枠組とすることができる(4.3を参照)。

- 複合メディア

基本メディアを組み合わせて、さらに上位階層の抽象メディアを作ることができる。

## 4 抽象データと抽象メディアの解説

### 4.1 マルチメディア・データベース

抽象データ式は、マルチメディア・データを共有するための統一的な記述形式である。ペリフェラル・モデルにおけるダンプ情報に相当し、構成部品となるオブジェクトの属性リストからなる。抽象データ式は、マルチメディア・データベースに適切な名前で登録される。データ属性は、属性名と値の組であり、抽象データを実行環境に合わせて、リアライズするための情報である。デジタル・データにおいては、データの実体があるファイル名やフォーマット・パラメータ、物理的なデータにおいては記録媒体の識別情報や再生時間などがある。

素データ式 := (クラス データ属性 ...)  
 同期関係データ式 := (クラス データ属性 ...)  
 抽象データ式 := 素データ式 | (同期関係データ式 抽象データ式 ...)

## 4.2 抽象データの実行

抽象データの実行において、特に時間経過だけを捉えた場合を走行という。抽象データは、走行状態として、走行開始前 ready、走行中 active、走行終了 done の3種類をとる。また、走行中は、各抽象データ固有の時間進行があり、それに伴い現在位置を表す時間ポインタの値が変化すると考える。

抽象データの同期走行は、以下のように行なわれる (fig 2 を参考)。

- 階層構造による同期

同期関係は、モジュール実行に相当し、下位の走行は上位の走行フェーズの部分と捉えられる。したがって、上位オブジェクトが下位に走行 (run) を命令し、下位は正規終了を上位に告げる (return) か、上位の終了により強制終了 (cut) となる。素データでは、それぞれのクラスに応じて、自身の正規終了の仕方を属性として与えておくことができる (時間 time-size など)。順次走行 serial では、自身の走行開始と同時に下位リストの先頭を走行開始にさせる。以下順に、下位の終了通知と同時に次のデータを走行開始にし、最後のデータの終了通知で自身の走行を正規終了する。並列走行 pararell では、下位の全てのデータを同時に走行開始させ、全ての走行終了で自身を正規終了する。

- イベントによる同期

各抽象データ部品には、複数の終了指定イベント (match-event) と 1つの発行イベント (send-event) を登録しておくことができる。走行中のデータが正規終了したとき、登録された発行イベントを抽象データ全体に送る (実際には、ルートに送って、下位に再帰的に伝えられる)。強制終了の場合は、イベントの発行を行なわない。また、自分が走行中に、受け取ったイベントが登録された同期イベントに一致したら、正規終了となる (一致しない場合は、下位にそのまま同期イベントを送る)。走行中でないデータは、イベントを無視する。

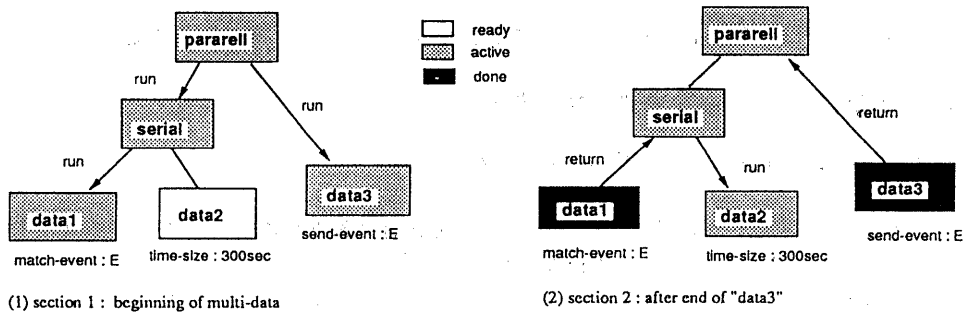


fig 2 : abstract execution of multi-data

### 4.3 抽象メディアとマルチメディア・システム

抽象メディアを、マルチメディア・アプリケーションを設計するための基本的なモデルとみることができる。特に、抽象データの加工や編集を MUMIN 上でシミュレートするとき、また動的に入出力先を変更するような複雑なメディア制御の場合に用いる。

入出力機器が管理する仮想的なデータを扱うため、抽象データの部品として、以下の基本データ (basic-data) も用いる。

sound(音声) still(静止画) motion(動画)  
 video(音声と動画の一体化した映像) null(時間進行)

基本データを用い、機器間のメディア・データの流れにおける各断面もオブジェクト化して、より精密なシミュレートを行なう。例えば、LD プレーヤの扱うデータは、LD データだけでなく、音声出力端子上の音声データも含めて考える。ポートの役割に応じて、入力、出力、素データを保持するストレージに区別する。ストレージに格納された抽象データは、その抽象メディアに占有されており、他の抽象メディアから直接的に操作や参照を行なうことができない。逆に、当の抽象メディアでは、内部状態の参照や更新だけでなく、抽象データの同期構造なども変更することができる。入出力ポートでは、ポート間の参照が行なわれる。ある抽象メディアの入力ポートが他の抽象メディアの出力ポートを参照している時、それは機器間の入出力の結合を表している。また、1つの抽象メディア内での入力ポートから出力ポートへの参照は抽象データの加工を表している。メディア機器の各機能の実行を、データ間に抽象操作を表す結合リンクを張ることと捉える。抽象操作のリンクは、広い意味のデータ変換を表している。例えば、fig 3 で、CD プレーヤの演奏は CD から音声への変換とみることができる。また、プログラム演奏機能は、単なるデータの実行だけでなく、同期データの作成という概念を含んでいる。抽象メディアは、こうした抽象データと抽象操作リンクの集合体をダイナミックに管理する。抽象操作は、抽象メディアのコア内のオブジェクトであり、抽象操作は、素材となる幾つかの抽象データを参照し、それらを加工・編集して結果としての抽象データをポートに割り当てる。複合的な抽象メディアにおいては、個々の操作ベリフェラルを1つにまとめ、またポートについては、複数の抽象メディアに渡る抽象操作で連絡することになる。

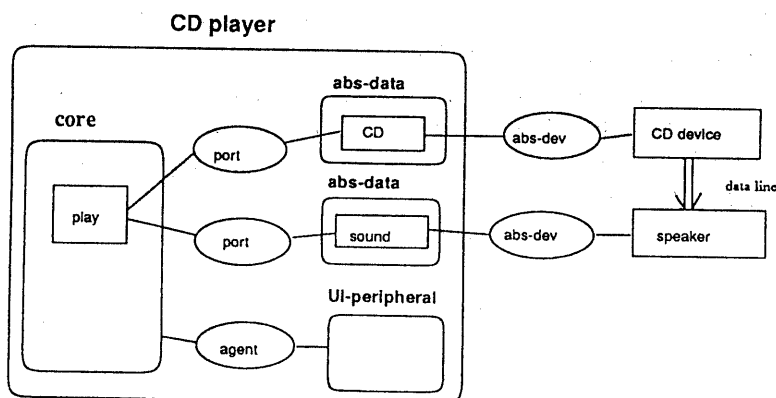


fig 3 : abstract media

## 5 まとめ

マルチメディア管理システム的设计のため、マルチメディアを適切な抽象レベルで捉える概念モデルを導入した。特に、UIMS機能とも整合性を保つマルチメディアのオブジェクト表現として抽象データおよび抽象メディアを提示した。抽象データでは、階層構造を保ちつつ、イベント通信を加えた同期機構を採用している。これは、GUIの構築手法を抽象データにも生かすためである。なお、MHEG [11]では、デジタル・データの標準化として、一般的なマルチメディアのオブジェクト表現について述べられている。本論では、物理的なデバイスなどの実行環境との関係において、インタフェース・レベルの設定やメディア依存部分の分離という観点から、オブジェクト表現を捉えている。

## 謝辞

MUMINプロジェクトに参加、協力していただいたすべての方々に深く感謝の意を表します。

## 参考文献

- [1] 新井潤 他：マルチメディア・ワークステーションの実現に向けて，日本ソフトウェア科学会第8回ソフトウェア研究会（関西）SW-91-8-6（1991）
- [2] Bass, L. and Coutaz, J. : *Developing Software for the User Interface*, 1991, Addison-Wesley
- [3] MacLean, A. et. al. : User-Tailorable Systems: Pressing the Issues with Buttons, *Human Factors in Computing Systems: CHI '90* (1990) pp. 175-182
- [4] Myers, B. A. : Encapsulating Interactive Behaviors, *Human Factors in Computing Systems: CHI '89* (1989) pp. 319-324
- [5] *Objectworks/Smalltalk User's Guide*, ParcPlace Systems
- [6] 大谷浩司 他：GMW ウィンドウ・システム上のアプリケーション構築について，コンピュータソフトウェア Vol.7 No.1 (1990) pp. 45-60
- [7] 暦本純一 他：エディタを部品としたユーザインタフェース構築基盤：鼎，情報処理 Vol.31 No.5 (1990) pp. 602-611
- [8] Sibert, J. L., Hurley, W. D. and Bleser, T. W. : An Object-Oriented User Interface Management System, *Computer Graphics: SIGGRAPH '86* Vol.20 No.4 (1986) pp. 259-268
- [9] 立川江介, 福井眞吾, 宮下洋一：制御部品を用いたユーザインタフェース構築支援システム，日本ソフトウェア科学会第6回大会論文集 (1989) pp. 429-432
- [10] 服部隆志：MUMINシステムにおけるユーザ・インタフェース管理，日本ソフトウェア科学会第9回大会論文集 (1992) pp. 365-368
- [11] 安田浩：マルチメディア符号化の国際標準，丸善