

## マルチメディア・ワークステーション - P I E

前川 守\* 太田昌孝\* 荒野高志\* 河内谷清久仁\* 野口佳一\*\*

\*東京大学理学部

\*\*株式会社CSK総合研究所

## 1. はじめに

コンピュータはますます高機能になり、その応用を広げているが、より効果的でユーザフレンドリなシステムとするため、文字、図形、画像、音などの様々なメディアを統合的に処理するシステムの必要性が高まっている〔1. 2〕。ここで紹介するマルチメディア・ワークステーション - P I E (Professional's Integrated Environment)は、このようなマルチメディア統合環境を提供することを目的としている。本報告ではその概念、設計、インプリメンテーションについて述べる。

P I Eにより作られたマルチメディア・オブジェクトの例を図1に示す。これには、文字、アニメーション、音の三種類のメディアが含まれている。ユーザがアニメーション開始キーを押すと、画面の中の車が動き出し衝突する(図2参照)が、この時、同時に衝突音が聞こえるしくみになっている。

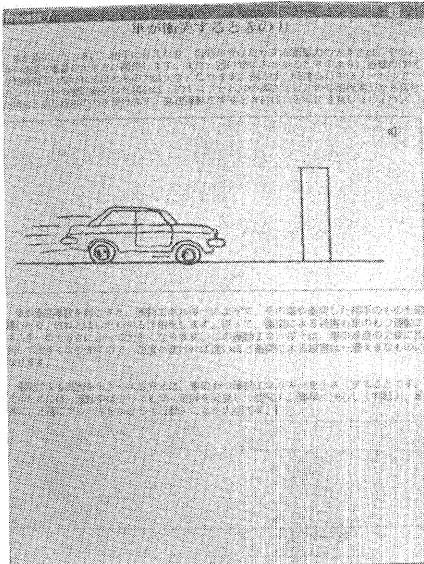


図1 P I Eによるマルチメディア・オブジェクトの例

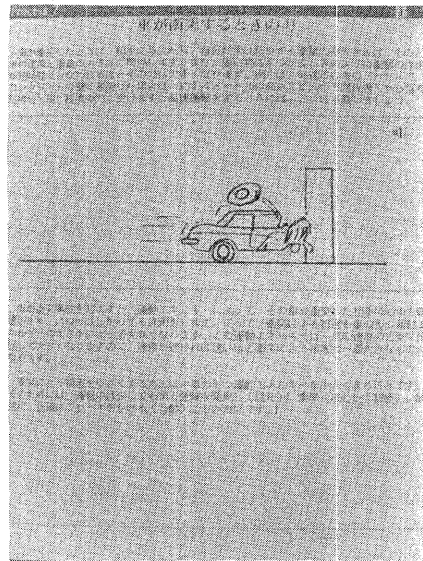


図2 アニメーション実行後のスクリーン

本報告は、全体を八つの章に分け、第2章でメディアの分類について、第3章ではマルチメディア・オブジェクトの表現方法と、マルチメディア・ワークステーションの構築に適した情報構造について述べる。さらに第4章では、マルチメディア・エディタ、第5章、第6章では、視覚オブジェクトと音オブジェクトについて、第7章ではP I Eのインプリメンテーションについて、それぞれ説明し、第8章を本報告のむすびとする。

## 2. メディアの分類

メディアは、情報蓄積や情報交換の媒体である。ここでは、このメディアをつぎのように分類する。

## (1) 視覚と音

日常生活において、人間の目と耳は情報を受け取る重要な役目をもっているが、ここでは、目から受け取る情報を視覚情報、耳から入ってくる情報を音情報と分類する。

## (2) 構造化と非構造化

ある情報が、コンピュータ処理に適した構造をもつとき、それを構造化情報と定義し、そうでないものは非構造化情報とする。ほとんどのメディアは、この構造化と非構造化の両面をもっている。例えば、文字は一連の文字コード(アスキーコード等)によって表現することができるが、一方で、手書文字や印刷文字としても表すことができる。一般に、非構造化情報は、同一のものに対してでも構造化情報より多くの情報

を含んでいることが多い。例えば、手書文字は単に文字の記号にとどまらず、筆者特有の書き方といったような非構造化情報も含んでいる。また、音情報は文字のシンボルとして表されるだけでなく、話し手の気持ちや雰囲気伝える役割も果たす。こう考えると、非構造化情報が構造化情報に変換されたとき、その情報の一部が欠落する恐れがあることを認識しなければならない。情報の構造化は、効果的なコンピュータ処理を行うために不可欠であるが非構造化情報は人間の知覚における所産であるといえる。

### (3) 静的と動的

時間の次元を持つメディアを動的メディア、その他を静的メディアと分けることができる。アニメーションや人間の声などは動的メディアの一例といえる。

以上のように分類すると、オフィスなどで使われる紙メディアは、視覚的に非構造的な静的メディアとみることができる。PIEでは、その対象をこのようなメディアから動的メディアの領域まで広げることを目的としている。

## 3. マルチメディア情報構造

第1章で述べた目的を満足するマルチメディア・ワークステーションを構築するには、マルチメディア情報構造の統合が鍵である。本章では、コンピュータによるマルチメディアの作成や操作に適したデータ構造を提案するとともに、PIEのシステム構造を示す。

### 3.1 コンピュータにおけるマルチメディア情報

メディアは、人間どうしの情報の交換や表現のための媒体である。このような媒体は、コンピュータによっても実現される。コンピュータでマルチメディア情報の蓄積や処理を効率的に行うためには、情報をコンピュータ処理に適した形態で表現し、蓄積しなければならない。この表現のことをメディア構造と呼ぶことにする。このメディア構造がコンピュータから外部世界へ出力されるときには、人間が見たり、聞いたりする形態（人間が理解しやすい形態）で表現される必要がある。この外部表現のことをメディア表現と呼ぶことにする。これらの関係を図3に示す。このメディア表現の例としては、音をとまなうアニメーションがある。このアニメーションのコンピュータ表現は、メディア構造である。このように、二つの情報表現形態を持つのは、システムがメディア表現のみしか持たなければ、編集能力が非常に制限されてしまうからである。たとえば、典型的な例として文字の削除すらできない従来のタイプライタを上げることができる。高度で複雑な情報操作というものは、情報がコンピュータ処理に適した形態で表現されて初めて可能となる。

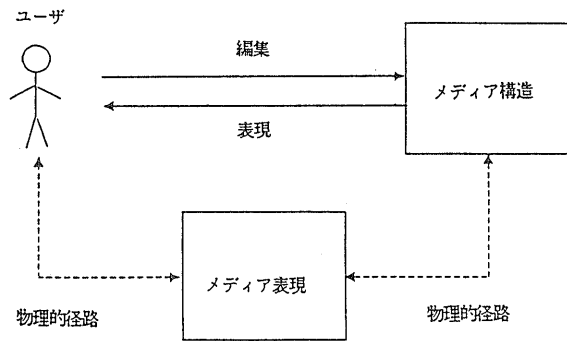


図3 マルチメディア編集のためのユーザ対話モデル

図3に示すように、編集はユーザとメディア構造の間で行われる。ユーザは、メディア構造に対して直接情報の作成や追加などの編集操作を行うことができる。このような編集操作の間、メディア構造はメディア表現として表示される。

実際の情報交換は、図3に示すようにメディア表現を通して行われる。これは、コンピュータ・ネットワークの階層構造に類似している。このメカニズムを効率的に行うには、メディア構造に対する修正や追加が直ちにメディア表現に反映されなければならない。さらに、ユーザが簡単にメディア表現から情報構造を理解できるシステムでなければならない。

### 3.2 メディア構造

メディア構造は、つぎのような要求を満たさねばならない。

- 1) マルチメディア情報を広範囲で表現できること。
- 2) ユーザが情報の作成や編集を簡単に行なえること。

データ構造としては、いろいろ考えられるが複雑な構造にするとつぎのような問題を引き起こす。

- 1) 多くのプログラマでない専門職の人は、複雑な構造を使えない。
- 2) プログラマでない専門職の人に、複雑な情報構造をわかりやすく表現する効果的方法は存在しない。

これらの理由により、ここでは木構造を採用し、この木構造が多くの応対に有効であることも確認した。

メディアを木構造で表現した例を図4に示す。木の最下位（リーフ）の部分を実質的・オブジェクトと呼ぶ。この実質的・オブジェクトは、ただ一つのオブジェクトからなる情報オブジェクトでありまた、それ以上別のオブジェクトに分けられない情報オブジェクトのことである。たとえば、文字、線、音などがそうである。

このように、マルチメディア情報オブジェクトを木構造で表現すると、つぎに示す二つの編集環境が必要となる。

- 1) 木構造を作成、更新するための環境。
- 2) 実質的・オブジェクトを作成編集するための環境。

前者は、木エディタにより実行され、後者は実質的・エディタにより実行される。実質的・エディタは、メディアに依存しているため個々のメディアについて準備されている。この実質的・エディタの採用は、情報構造の簡潔さを保ち、かつ編集能力を増加させる。

図5は、マルチメディア・オブジェクトの例である。この例は、文章の中に画像を含み、さらに画像の中に説明文を含む。この例を木構造で表現すると、図6のようになる。

### 3. 3 フォーマット情報および

#### 実質的情報

木構造は、マルチメディア・オブジェクト間の論理関係により定義される。木構造で表されるメディア情報を、構造情報と呼ぶことにする。しかし、木構造自身は、メディア構造をメディア表現に変換するための十分な情報を含んでいない。たとえば、図6に示した木構造だけでは、システムは富士山の絵をどの位置にどのように表現すればよいかわからない。

これらを決定するためには、つぎに示す二つの情報が必要である。

- 1) 実質的情報
- 2) フォーマット情報

実質的情報は、実質的固有の

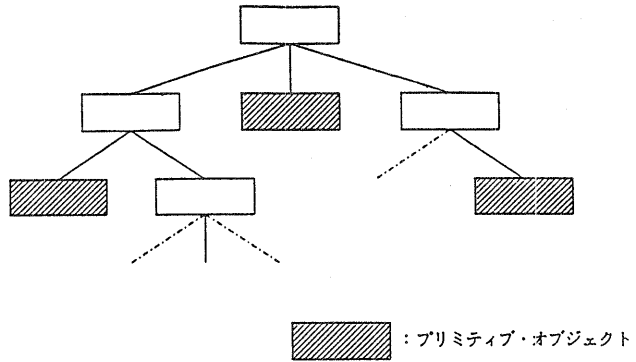


図4 メディアの木構造表現

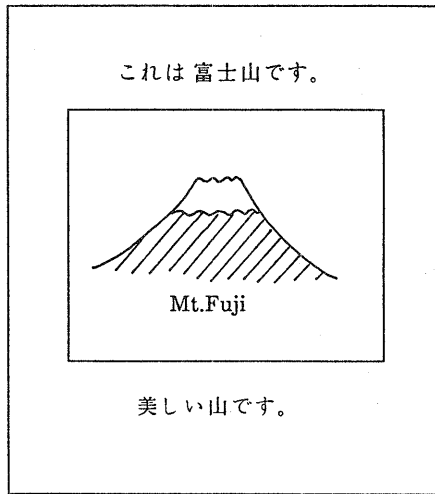


図5 マルチメディア・オブジェクトの例

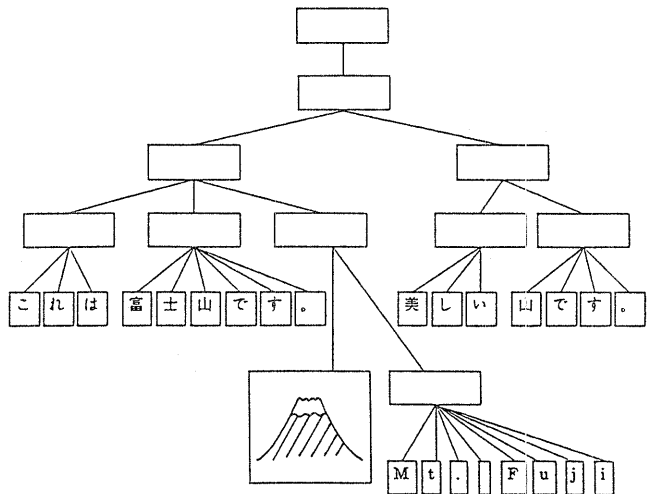


図6 図5のメディア構造

情報である。たとえば、富士山の形はプリミティブ情報である。フォーマット情報はメディア表現を規定するために用いられ、この一部は、木構造のノードに連結している。すなわち、フォーマット情報によって富士山の絵の位置が規定される。

PIEのフォーマットは、Scribe〔3〕に似た環境メカニズムを用いている。すなわち、フォーマット情報は、プリミティブ・オブジェクトのフォーマティングを規定する変数の集合体として、フォーマティング実行中に保持される。これらの変数を環境変数と呼ぶ。環境変数は、書き変えない限り、上位ノード（先祖ノード）から下位ノード（子孫ノード）へと受け継がれる。最上位ノード（ルート）におけるグローバル変数の値は、デフォルト値として与えられる。

環境変数は、プリミティブ・オブジェクトをフォーマティングする時に、その詳細を決定するために参照される。環境変数には、文字のフォント、図の位置、音量などといったフォーマット情報がある。現在、PIEは四つのタイプのフォーマット情報とそれらに対応する環境変数をサポートしており、この四つのフォーマット情報にはブーリアン、整数、浮動少数、ストリング（コード）がある。

### 3.4 システム構造

PIEのシステム構造を図7に示す。木エディタがPIEの中核であり、このエディタが、メディアの種類に応じて木構造を管理するとともに、フォーマタなどのモジュールを起動する。プリミティブ・エディタは、それぞれのメディア形態に対して用意され、プリミティブ・オブジェクトの作成や編集に用いられる。フォーマタは、メディア構造からメディア表現を作り出すとともに、個々のプリミティブ・オブジェクトのフォーマティングを行うプリミティブ・フォーマタを起動する。本報告では、フォーマタの概念をマルチメディア・オブジェクトを含む範囲まで拡張した。ユーザからの編集命令は、コマンド・トランスフォーマが、フォーマタにより作られたレイアウトデータをもとに、メディア構造に対する命令集合に変換する。

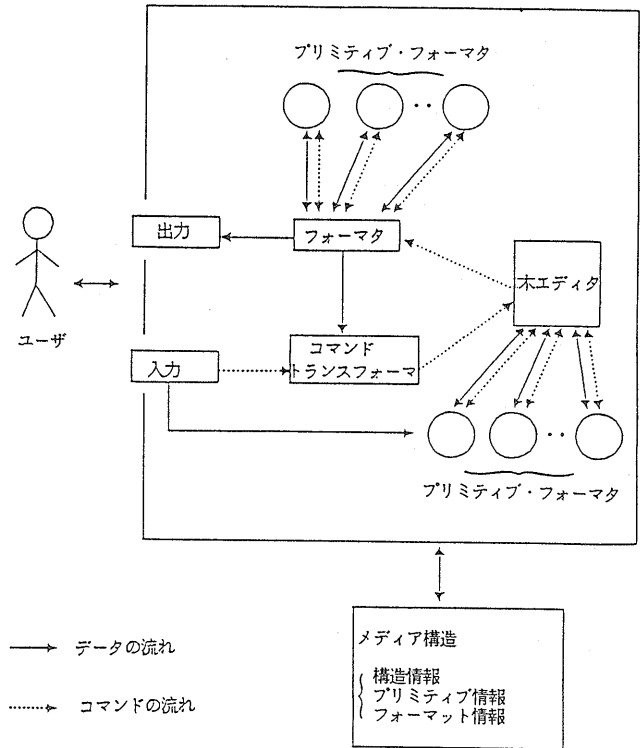


図7 PIEのシステム構造

### 4. 木エディタ

本章では、木エディタの機能について説明する。木エディタは、木をたどりながら構造情報とフォーマット情報の修正や編集を行う。編集時に注目しているオブジェクトは、ディスプレイ上で反転表示される。このオブジェクトのことを、ポインティッド・オブジェクト（注目オブジェクト）と呼ぶ。木エディタの機能は、LISPシステムの構造エディタに似ているが、この木エディタでは、全体のオブジェクトと現在注目しているオブジェクトの両方を見ながら木構造を編集することが可能である。この木エディタは、以下に示すような機能を持っている。

#### 1) ポインタの移動

ポインタは、木構造の最上位、最下位、現在注目しているノードの前後へと自由に移動する。

#### 2) 新しいノードの作成

新しいノードは、最下位ノードあるいは現在注目しているノードの上下の場所に作成する。

#### 3) 注目中のノードの削除

現在注目しているノードとそれに連なる下位ノードをまとめて削除するか、あるいは、注目中のノードのみを削除し、それに連なる下位ノードは削除するノードの上位ノードに引き渡す。

4) 注目中のオブジェクトの切り出し

現在注目しているオブジェクトを仮のバッファにコピーする。

5) 切り出しオブジェクトの移動

4) によって切り出されたオブジェクトを、注目中のノードの下位あるいは前後のノードに移動する。

6) 注目中のオブジェクトのフォーマット情報編集

木エディタは、それぞれのフォーマット情報の名前や種類を含んだファイルを読み、この情報にもとづいてユーザが指定したフォーマット情報を変更する。

7) プリミティブ・エディタの起動

注目中のプリミティブ・オブジェクトに対応するプリミティブ・エディタを起動する。

以上の機能は、ファンクションキーにより指示される。もし、これらのキーがプリミティブ・エディタ実行中に押されると、プリミティブ・エディタは動作を中断し指定された構造の編集が行われる。そして、その後まだプリミティブが注目中のオブジェクトに含まれている場合は、プリミティブ編集を再開する。この操作方式は、ほとんどのマルチメディア・オブジェクトの作成や処理に有効である。伝統的なドキュメント処理は、このエディタが備えている機能の一部にすぎないのである。

## 5. 視覚オブジェクト

視覚メディアは、記号、図形、画像、動画、コンピュータグラフィックスなど、人間の目によって知覚されるさまざまな情報を含んでいる。そして、これら視覚メディアの情報量は非常に多いということがいえる。例えば、512X512 解像度の白黒の絵は256kビットの情報を持つことになる。もちろん、これらはすべての情報がその絵を見る人に伝えられるわけではなく、冗長な部分もかなりある。しかし、一見冗長な部分も実は視覚メディアのわかりやすさにとって、たいへん重要な役割を果たす。文字のフォントや位置には重要な意味があるし、内容の切れ目や字下げなどは、文章を読みやすくするためのテクニックでもある。このように、視覚メディアを作り出すために、ユーザは冗長部分を含め、さまざまな情報を、なんらかの方法で指定しなければならない。つまり、視覚メディアの最終的な品質は、ユーザがこれらの情報をどれくらい正確に指定できるかによって決まるのである。

ユーザインタフェースのために最も重要な要素は、ユーザ自身が、自分の行った指定の効果を即座に確認できるということである。しかし、イメージを実時間で処理することは難しい面がある。三次元のラスターコンピュータグラフィックスなどとも言えることだが、このような場合、コンピュータには、ユーザが行った指定を確認できるようななんらかの機能が備わっていなければならない。

### 5. 1 視覚プリミティブ・オブジェクト

視覚オブジェクトには次のようなものがあげられる。

- 1) 文章オブジェクトのための文字（漢字を含める）と記号
- 2) 二次元コンピュータグラフィックスのための点、線、長方形
- 3) 単なるビットイメージ
- 4) ワイヤフレームの三次元コンピュータグラフィックス

これらはすべてPIEでサポートされており、その各々については後でさらに詳しく説明する。視覚オブジェクトのためのフォーマット情報には、オブジェクトの色と、ある視覚オブジェクトを他の視覚オブジェクトに重ね合わせるビット操作という二種類がある。

### 5. 2 文字プリミティブ

文字処理の一つの問題点は、七千近くにのぼる漢字の入力方法であるが、PIEでは、一般的なカナ漢字変換を採用している。約八万文節のカナの読みが、その文法情報とともに辞書に備えられている。変換は、文節単位で行われるため、その文節構造は、カナ漢字変換によって、自動的に木構造に変換されるようになっている。すべての文字は、JIS規格6226による文字コードで蓄積されている。また、文字は、それ以上細かく分けられない最小単位であるため、PIEには文字プリミティブの編集機能は備えられていない。

文字の出力に関しては、さまざまな形態が可能で、通常のワープロ同様、その環境変数に応じて、フォント変更、字下げ、ページ付けなどの機能がサポートされている。

### 5. 3 図形プリミティブ

点、線、長方形は構造化された図形を書くためのプリミティブである。点は、その位置によって記憶され線は二つの点の位置によって表され、長方形は、対角線上の二つの座標の位置で決められる。図形プリミティブはすべて、一つの長方形の枠内にふくまれているとよい。その位置は、指定されたフォーマット情報によって決められ、環境変数により保たれる。座標は、図全体の大きさによる、相対的な指定と、絶対的な指定がともに可能である。図形プリミティブの編集は、これら点、線、長方形プリミティブの挿入、移動、削除を行う。位置は、マウスのようなポインティングデバイスで指定するが、適当なフォーマット指定を行うことで"×"マークや円を、点によって作ることも可能である。また、線や長方形は、点線、太線二重線のような指定もできる。

### 5. 4 ビットイメージ・プリミティブ

ビットイメージ・プリミティブは、ファクシミリからの入力や、ペイントシステムによって描かれ、ビットの二次元配列によって表される。

ビットイメージ・プリミティブの編集は、ペイントシステムと呼ばれ、筆のようなもので、ビットイメージを描いたり、消したりすることができる。拡大、縮小、回転も、このペイントシステムで可能である。

### 5. 5 ワイヤフレーム・コンピュータグラフィックス・プリミティブ

ワイヤフレーム・コンピュータグラフィックスは、アスキーコードによって表されるが、現在はまだ実験段階であるため、PIEでは、木構造を使用している。このワイヤフレームシステムを、ラスタグラフィックスと組み合わせると、高速なユーザインタフェースを提供できるであろう。また、簡単なコマンド操作で、三次元空間内で線を自由に回転、縮小、拡大、移動することが可能となろう。コンピュータグラフィックスを作成、編集するためには、通常のエディタを使用する。回転、サイズ変更、移動などの程度は、環境変数で指定される。また、これらすべての三次元変換は、第3章で説明したデータ構造に基づいて行われる。

### 5. 6 アニメーション

アニメーションは、動的視覚メディアである。アニメーションをサポートするためのフォーマット情報は、フレーム番号を指定する。開始キーによって、アニメーションが起動されると、環境変数の値と一致するフレームの視覚オブジェクトが表示される。このように、フォーマット情報が、順次アニメーションフレームの視覚オブジェクトを指定することによって、アニメーションの動きが表現される。

## 6. 音オブジェクト

音には人間の声、音楽、鳥のさえずり、道路の騒音など、さまざまな種類がある。コンピュータの音メディアには、警告を知らせるブザーがあるが、最近ではこの他に、パソコンにおける自動演奏や、音声を利用したユーザインタフェースなども見られる。

音情報は、その形態によっていろいろ違ったメディア構造で表現することができるので、音メディア処理について述べる前に、まず、音メディアの分類についてふれる。音メディアは、図8に示すように三つのグループに分類することができる。このように、音声オブジェクトは文字コードに、音楽オブジェクトは楽譜にそれぞれ変換することができるが、その他の音オブジェクトは、メディア構造には変換できない場合もある。

非構造化情報	対応する構造化情報
音 声	文字コード (文章メディア)
音 楽	楽譜
音	なし

### 6. 1 音プリミティブ・オブジェクト

ここで、音メディアにはどのようなプリミティブ・オブジェクトがあるのかということが問題になる。図8において楽譜のように、構造的表現を持ちうるものを構造化音プリミティブ(楽譜プリミティブ)とし、音声、音楽そのもの、その他騒音のような音を非構造化音プリミティブと分類することができる。その編集方法においても、楽譜などの構造化音プリミティブは木エディタによる編集が可能であるが、非構造化音プリミティブについては、別の編集方法が要求される〔4〕。

図8 音メディアの分類

## 6. 2 非構造化音プリミティブ

PIEでは、マイクやスピーカを使って、生の音を入力することができる。マイクを通じて入力されたデータは、デジタル化され、PIEのメモリ内に蓄積されるが、メモリ内のこれらの情報は、デジタル/アナログ変換によって出力することができる。音声メモの、相対的あるいは絶対的音量やトーンは、フォーマット情報によって指定できる。

マルチメディア・ドキュメントにおける音プリミティブは、図9に示すような音マークによってディスプレイ上に示される。ユーザがマウスでこの音マークを選ぶと、非構造化音プリミティブのフォーマタが起動されプリミティブが出力される。また、アニメーションなどによってもこの非構造化音プリミティブのフォーマタが起動されるようになっている。

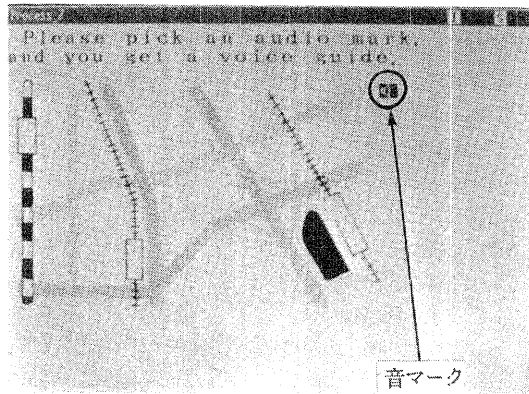


図9 音マークの例

## 6. 3 楽譜プリミティブ

楽譜プリミティブをサポートするための要素としては、楽譜エディタ、楽譜を解釈し演奏する楽譜フォーマタ、手書きや印刷の音符の認識機能などをあげることができるが、これらは現在まだ研究の段階である。

## 6. 4 課題

音メディアに関する問題点として、つぎのようなことがあげられる。

- 1) 変換をどう実現するか。
- 2) ハードウェアの構成はどうあるべきか。
- 3) 音メディアが、アニメーションなどどのように統合されるべきか。

音声変換の方法はいろいろ考えられるが、PIEでは、“音声-文字コード辞書”（それぞれの音節が文字コードに対応している表）を利用して、文字メディアを音声に変換している。これは簡単なメディア変換の手法であり、この方法によって再生される音質は、音声合成によるものより、比較的良質である。また、この方法によると、表題や表の読み合わせが効果的に行われる。ただ、音情報が蓄積されるメモリ容量によっては、語数が制限されるという欠点がある。

## 7. インプリメンテーション

PIEは、Apollo domain DN660 ワークステーション上でインプリメントされている。システムはC言語で書かれており、他のApolloワークステーションにも移植可能である。図10にPIEのシステム構成を示す。

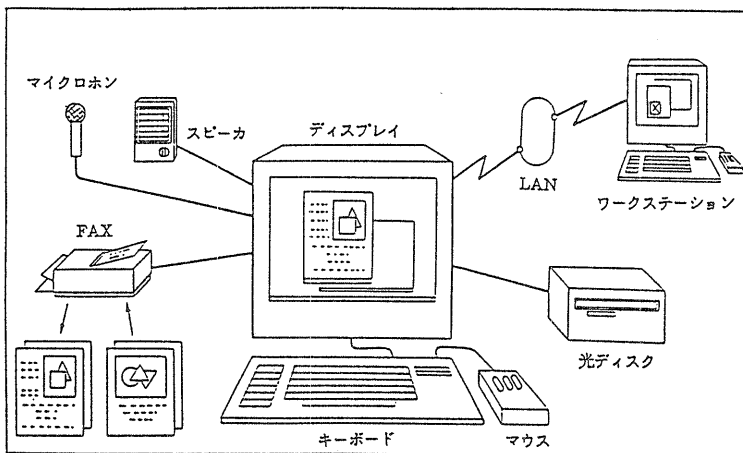


図10 PIEのシステム構成  
(7)

## 8. むすび

本報告では、マルチメディアワークステーションPIEの基本概念、構造、インプリメンテーションについて述べた。PIEの目的は、マルチメディア・ワークステーションを構築することと、その設計やユーザインタフェースを実験するという二点である。これまでの研究や実験から、今後さらに改善していかなければならない点をまとめると、つぎのようになる。

- 1) マルチメディア情報オブジェクトを表現するための木構造は、さまざまなドキュメント処理において効果的ではあるが、知的処理を行うためには、さらに意味的構造が必要である。また、よりユーザフレンドリなインタフェースを備えるためには、効果的な言語が必要と思われるが、この点については他の機会に報告したい。
- 2) PIEの性能向上のために、幾度かそのシステム設計を変更したが、自然なマルチメディア・オブジェクト処理を行うためには、さらに高度な性能、特にスピード面での改善が必要である。このためには、より強力なワークステーションが不可欠になってくる。

## 謝辞

PIEシステムは、文部省の“民間との共同研究”制度により、株式会社CSK総合研究所と東京大学理学部情報科学科前川研究室との間の共同研究の一環として開発されているものであり、文部省研究協力室に感謝の意を捧げたい。また、日本アポロコンピュータ株式会社には研究推進の上で種々の御協力をいただいております。ここに深く感謝する次第である。なお、表題の著者の他に、坂村 健、清水 徹、松為 彰、濱野純、鳥居秀行、居初千春の諸氏が、本プロジェクトの参加者または貢献者である。

## 参考文献

- (1) Maekawa, M. et al. "Multimedia Machine." INFORMATION PROCESSING 83, R. E. A. Mason (ed.) Elsevier Science Publishers B. V. (North-Holland) IFIP, pp. 71-77.
- (2) Donohue, III, B. P. "The Integration of Multimedia Communications." Proc. of AFIPS, National Computer Conference, Vol. 52, 1983, pp. 367-371.
- (3) Reid, B. K. "Scribe: A Document Specification Language and its Compiler." CMU, Department of Computer Science, CMU-CS-81-100, October 1980.
- (4) Buxton, W., et al. "The Evolution of the SSSP Score-Editing Tools." Foundations of Computer Music, edited by C. Roads and J. Strawn, Cambridge: MIT Press, pp. 376-402.