

マルチメディアデータベースのオブジェクト構成と そのユーザインタフェースの設計

金森吉成 脇山俊一郎⁺ 増永良文⁺⁺

群馬大学 + 仙台電波高専 ++ 図書館情報大学

マルチメディアデータベースのユーザインタフェースを一般的に設計するために、言語による記述方法を提案した。このためにマルチメディア言語を定義した。言語は画面のレイアウト記述文法とマルチメディアの内容記述文法とからなる。言語で記述されたインタフェースを翻訳実行するマルチメディアインタフェースプロセッサがある。このプロセッサを異種分散オブジェクト指向データベース環境に適用した場合の振舞いについて検討した。環境としてOMGが提案にしているORB(Object Request Broker)による共通プラットフォームを使用した。また、医学分野の具体例を使ってマルチメディア言語によるインタフェース設計の概略を示した、

Object Organization in Multimedia Database and Designs of Its User Interface

Yoshinari Kanamori, Shunichiro Wakiyama* and
Yoshifumi Masunaga**

Gunma University

* Sendai National College of Technology

** University of Library and Information Science

In this paper, we have proposed a generic design method of user interface for multimedia database using a language. We defined the language for multimedia. The Languages have two parts: layout description grammar for display and contents description grammar for multimedia. The multimedia interface processor that we call compiles and executes the interface written by the language. When we applied the processor to the environment under heterogeneous distributed object-oriented database system, for example, common platform of Object Request Broker(ORB), we simulated the operations of the processor. We sketched out the interface for a concrete multimedia database in the medical field by use of the language.

1. はじめ

マルチメディアのオブジェクト指向データベースシステムをユーザが利用するためには、システムに高度なユーザ支援機構が備わっていないなければならない。

利用者はマルチメディアデータの使用において、いろいろな処理、操作を要求する場合も多く、インタフェースの仕様は多様になり、開発コスト、労力共に多大である。

従来、インタフェースの設計では一般的な手法がないために個別的に対応してきている。著者等も、医学分野のマルチメディアデータを対象としてオブジェクト指向データベースの設計をしてきたが[1、2]、ユーザインタフェースの部分は問題向きになっている。

今後、異種オブジェクト指向データベースシステムによる分散データベースシステム環境が生じてくると、計算機環境の変化のために開発済のユーザインタフェースを修正する場合も起きてくる。

このような観点から、汎用的な設計手法を開発することにした。本論文では、言語によるインタフェース設計の方針、構想について述べる。すなわち、設計するためのマルチメディア言語を提案する。

2. マルチメディアインタフェースのアーキテクチャ

2.1 アーキテクチャの概要

ネットワーク環境下にある異種オブジェクト指向データベースシステム(Heterogeneous OODBMS)の間では、標準オブジェクト指向データベースモデルが存在しないために、オブジェクトを共有することができない。データベースを共有するためには、データモデルの違いに依存しないオブジェクトの相互運用可能性が必要となってくる。

このような視点から、Object Management Group(OMG)からの提案によるObject Request Broker(ORB)の概念がある。

本論文では、以後、HyperDesk社の提案したORBの考え方に基づいて議論をする[3]。

図1に示すように、ORBによる共通プラットフォームを用いてオブジェクトの環境を標準化することを提案している。すなわち、オブジェクト名の管理、オブジェクトの生成、消滅の管理、オブジェクトの定義などを共通プラットフォームを介して標準化し、相互運用を目指している。

マルチメディアデータベースのユーザインタフェースの機能としては、音、画像、図形、動画、またはこれらの複合したものを利用者が直接操作して、可視化、可聴化することが求められる。

その際に、利用者が必要とするオブジェクトがどここのOODBMSにあるか、またOODBMSが異種であるかなどを意識させないようにシステムが支援することが必須である。

ORBによる共通プラットフォームの考え方は、このような機能を持つインタフェースを作成する観点からも非常に有効な手法である。

図1の詳細については、以後の議論を進めながら順次説明をしていく。

2.2 マルチメディア言語(MML)

(1) 言語の必要性

GUIの環境が整備されてきたから、X Windowシステムの下でユーザインタフェースを作成する際には、開発された様々なツールを利用できる。従来と比べれば、可視化ユーザインタフェースの開発はかなり容易になってきている。

しかし、これらのGUI開発のためのツールを利用してインタフェースを一般的に設計する手法は現在まだなく、問題向きに、個別的に対応し

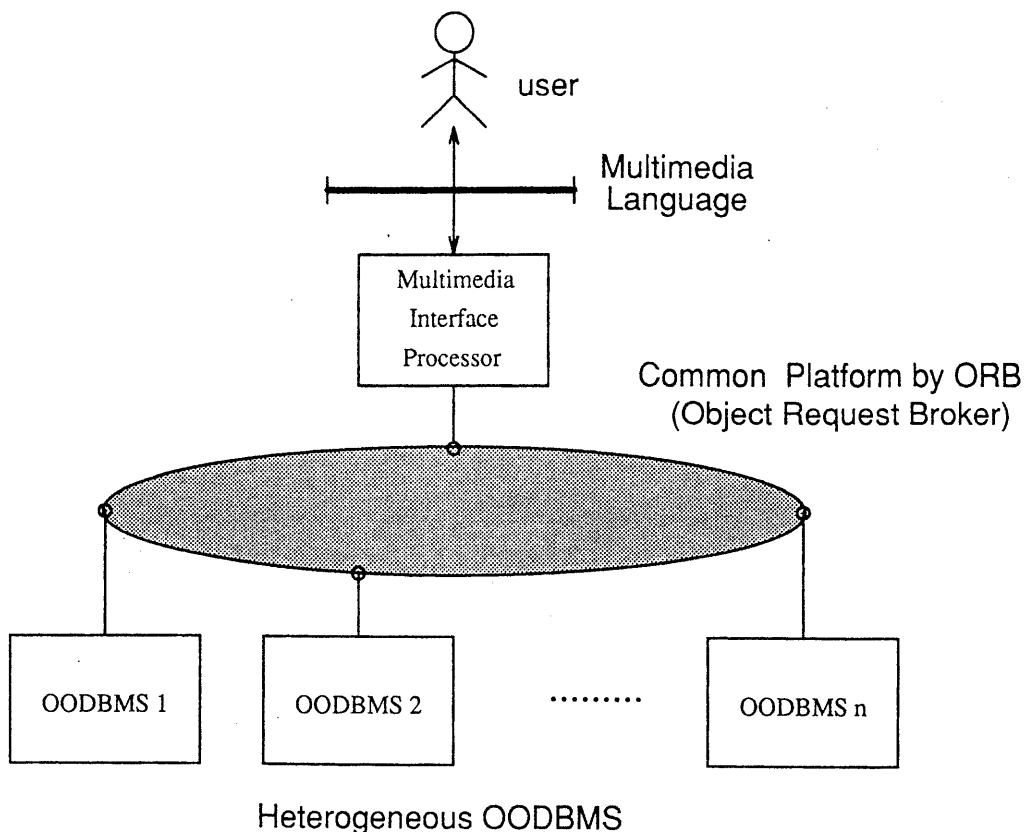


図1 マルチメディアインタフェースのアーキテクチャ概要

ている状況にあると言えよう。

システム全体におけるユーザインタフェース作成コストが80%近くにもなることを考えると、またマルチメディア環境での様々なデータベース応用が進展するためにも、汎用的な設計手法の確立は極めて重要な課題である。

このような問題認識により、ここでは異種OODBMS環境にも適用できる一般的な設計方法として、言語によるインタフェースの構造記述を提案する。

この言語をマルチメディア言語 (Multimedia Language, MML) と呼ぶことにする。

それによって記述されたインタフェースを解釈実行するプロセッサをMultimedia Interface Processor (MIFP) と名付ける。

図1に、ANSI/X3/SPARCの書き方による言語

MMLとインタフェースプロセッサMIFPを示す。

ユーザは、MMLを使用してORBのプラットフォームを介してOODBMSと対話することができる。

(2) MMLの定義

利用者とOODBMSとの対話のユーザの空間は、マルチメディア言語で記述されて、その結果がディスプレイ上に表現されるものとする。言語は、マルチメディアの内容記述と画面のレイアウト記述とからなる。

ここで、フォーマルな定義を与える。

Layout Description Grammar : G_L
 Contents Description Grammar : G_{Ci}
 Multimedia Grammar : MMG

とすると、MMGは $n+1$ 項組みである。

$$MMG=(G_L; G_{C1}, G_{C2}, \dots, G_{Cn})$$

次に、理解し易くするために図2を例にしてMMLを説明する。

1画面が4つのレイアウト $l_{0t}, l_{1t}, l_{2t}, l_{3t}$ からなり、かつ1つのレイアウトには簡単のために1つの内容 C_{jt} のみがあるものとする、時間 t におけるMMLは次式で与えられる。

$$L(MMG)=\{interface_t=(l_{0t}; C_{0t}, l_{1t}(C_{1t}), l_{2t}(C_{2t}, l_{3t}(C_{3t}))) \mid t\}$$

ここで、 $l_{it} \in L(G_L), C_{jt} \in L(G_{Cj})$

これは、時間経過 $t, t', \dots$ と共に変わっていくので、次のように表される。

$$interface_t \vdash_{MMG} interface_{t'} \vdash_{MMG} \dots \vdash_{MMG} \dots$$

図2のようなレイアウトの構成は、X Windowなどで普通に用いられている。入れ子の構造になっていて、小さいレイアウトは下のより大きなそれの上に重ねられているものとする。さらに、ビデオを内容とする場合などでは[4]、Compound Contents Grammerを定義する必要がある。典型例を示すならば、次式で与えられる。

$$G_{video} = \text{Composit} \langle G_{image}, G_{sound} \rangle$$

言語による表示レイアウトの記述例として、分散処理環境でのドキュメントの相互運用である文書交換形式 (Office Document Architecture, ODA) を挙げることができる[5]。ODAの場合は、レイアウトをいくつかのパターンに標準化できる。また、1つの文書の論理構造が時間的に変わらないので静的なレイアウト構造を与えればよい。本質的に音、動画のメディアを対象にしてない。

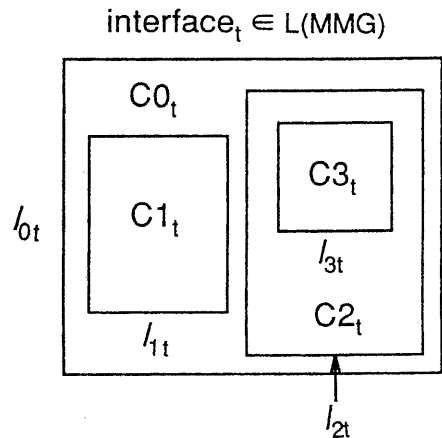


図2 マルチメディア言語 (MML) によるインタフェース記述例

一方、マルチメディアインタフェースでは、ユーザによっていろいろな対話形式があるから、レイアウトを標準化できない。しかも、利用者の使用中にデータベースとの対話形式が時間的に変わるから、動的なレイアウト構造記述の機能が必要不可欠である。音、動画などの時間的可変なメディアも対象にしなければならない。従って、MMLにはODAに比べてはるかに複雑な機能が要求される。

2. 3 マルチメディアインタフェースプロセッサ

ここでは、図1に示したマルチメディアインタフェースプロセッサ (MIFP) の機能についての詳細を説明する。

図3は、HyperDesk社のORBプラットフォームに、著者らが提案するマルチメディアインタフェースプロセッサがどのように関わるかを示している。用語は、HyperDesk社のそれを採用している。

図は、説明を解り易くするために、分散データベース環境の1部分を取り出した状況を想定している。

異種なOODBMSであるODB-1とODB-2がそれぞれ別のサーバー上にあり、画像オブジェクト

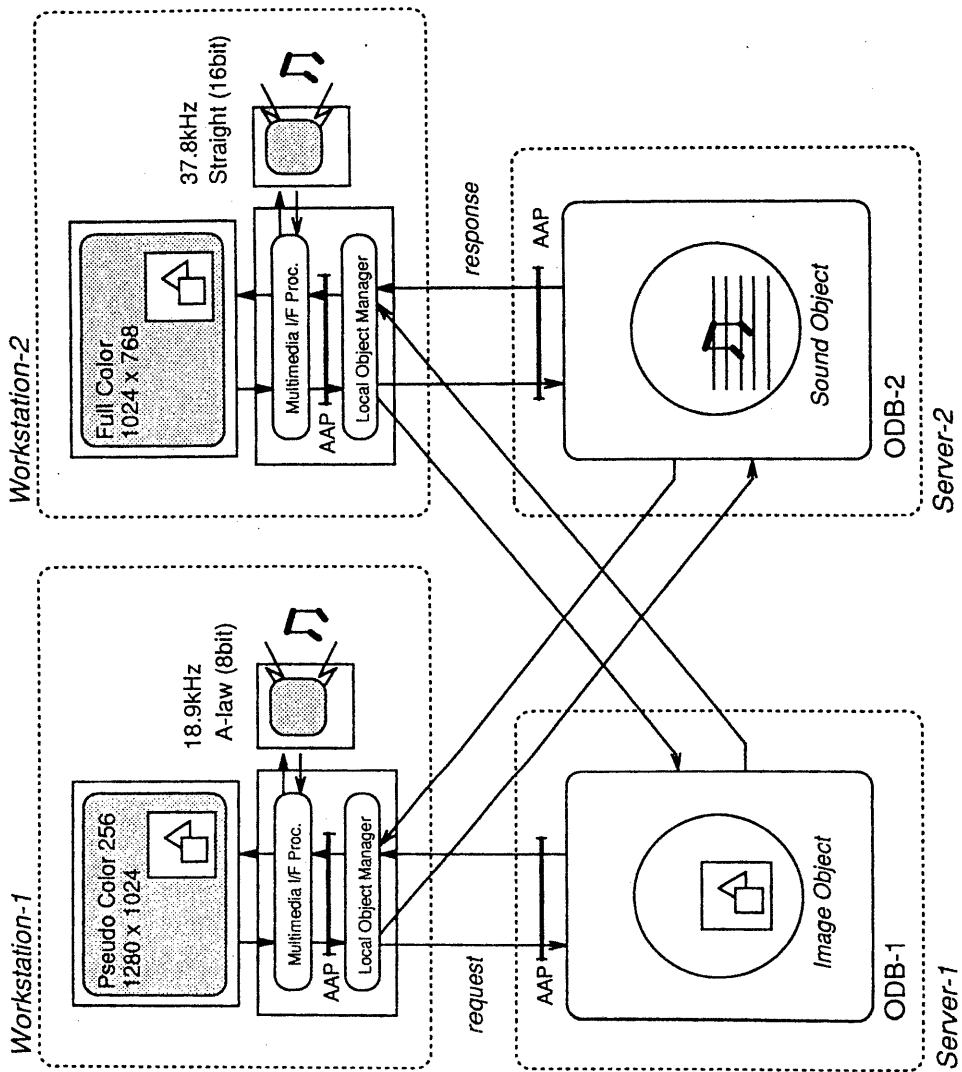


図3 分散データベース環境におけるマルチメディアインタフェースプロセッサ

と音オブジェクトがそれらにより管理されている。ディスプレイ性能、音の入出力形式が異なる2台のワークステーションが接続されている。各部の機能を、以下に簡単に説明する[3]。

(1) Local Object Manager(LOM)

LOMは、ORBの役割を果たしている。

LOMは、オブジェクトが互いに通信できるための通信マネジャーの役割を持っている。異種マネジャーとも通信することができる。すなわち、図のワークステーション1と2の異種LOMの間で通信が可能である。

オブジェクトにリクエストを出し、それに対するレスポンスを受け取ることができる。リクエストにより、メソッドを選び出して、それを起動することができる。他のLOMやアプリケーションプログラムへメソッドをディスパッチする。

オブジェクトの生成、消滅のライフサイクル、オブジェクト名、オブジェクトの定義についてのメタオブジェクトを持っている。

各ワークステーションは、LOM、MIFPを持ち、LOMを介してオブジェクトにリクエストする。

(2) Application Access Point(AAP)

オブジェクトが通信に用いるリクエストの構造を定義する。インタフェースであり、1種の言語と見ることができる。8種類のCallがあり、例えば、

Get_Request, Send_Request などがある。

あるオブジェクトにリクエストを送るクライアントは、応答するオブジェクトがどのコンピュータ環境で稼働しているかについての知識を必要としない。

AAPは、LOMとMIFPの間、LOMとODBの間に介在する。

(3) Object Database(ODB)

AAPを持つOODBMSである。オブジェクト

にIDを付けて、生成、消滅を管理する。オブジェクトにメソッドを結合させる。同種の他のODBへリクエストをディスパッチする。

(4) Multimedia Interface Processor(MIFP)

MMLで記述されたインタフェースを翻訳実行する。動的に変化する画面を管理する。

画面のレイアウト構造に基づき、各レイアウトに内容を表示する。アプリケーションプログラムの実行を管理する。

利用者がディスプレイ画面を見ながらマウスをクリックして出すイベントを受け取り、LOMにリクエストを出す。LOMからレスポンスを受け取り画面のレイアウト、内容を変更する。

2. 4 マルチメディアオブジェクトのメソッド

上述の共通プラットフォームの環境下で、OODBMSが管理するマルチメディアオブジェクトのメソッドとして何を準備すべきかを検討してみたい。

異種OODBMSが混在し、しかも各種ハードウェア環境の異なるワークステーションが接続されることを考えると、ユーザがアプリケーションプログラムを作り易くするための条件に重点を置くべきである。

また、環境の違うワークステーションで開発されたアプリケーションプログラムも修正することなく他のワークステーションでも稼働すべきである。

これらの要求から、オブジェクトはハードウェアに依存する部分をメソッドとして持つべきである。

この視点から、具体的に画像、音オブジェクトのメソッドとして何が用意されるべきかを考えてみる。

(1) 画像オブジェクトのメソッド

(a) 画像格納フォーマットから標準フォーマットへの変換、またはその逆変換

TIFF, GIF, JPEG etc. のフォーマットで記録されている画像を、X Windowシステムの標準であるポータブルフォーマット（ピクセル単位でRGBに各1バイトを割り当てる）に変換するメソッドである。

オブジェクトを表示するアプリケーションプログラムでは、圧縮された複雑な画像格納形式がカプセル化されているので操作が極めて簡単になり、非常に有効である。

(b) 拡大、縮小、座標変換などの処理

オブジェクトの座標系、サイズをリクエストした座標系、サイズに合わせて変換するメソッドである。

画像データを入力したときの座標系、入力の際のスキニング精度、入力画像の大きさなどの詳細をカプセル化するために役立つ。

例えば、1024 x 1024でスキニングした画像を256 x 256にする縮小する操作、あるいはデータを2048 x 2048に拡大するためにディザ処理をして補間するなどがある。

結果として、元の画質が修正されるが、適当な処理をメソッドに組み入れることによりほぼ等価な画質を生成する機能を持たせる。

このメソッドの有効な例として、3.2で述べる画像と図形オブジェクトとを重ねて表示するアプリケーションプログラムでの利用を挙げることができる。画像、図形の入力座標系、サイズがそれぞれ異なるので共通座標系への変換、両者間でのサイズの拡大縮小が必要であるが、プログラムからリクエストでメソッドが行うので利用者の負担が大幅に軽減する。

(c) ディスプレイへの表示

ディスプレイのハードウェア的性能であるRGBのビット数（フルカラー、256色など）、あるいはRGBの順序、カラーパレットの種類、画面サイズ（1024 X 1280, 1150 X 980, etc.）などが機種、メーカーにより異なる。

アプリケーションプログラム側でハードウェアの性能に合わせて表示のためにオブジェクトを操作するとディスプレイが異なる環境ではプログラムを修正しなければならない。

そこで、アプリケーションプログラムからはディスプレイ環境の条件を、オブジェクトにメッセージとして与えると、条件に合わせて変換してくれる機能をメソッドに持たせる。

例えば、オブジェクトの画像データがフルカラーで格納されているとする。一方、ディスプレイが256色表示の性能であれば、フルカラーの値をそのディスプレイに標準で用意してある256色のカラーパレットの値に変換する。

(2) 音オブジェクトのメソッド

入出力デバイスによる音のデジタル化精度、またデータ圧縮などによるデータ格納形式がいろいろある。

ADPCM, Straight8, Straight16, μ -law, A-lawなどの量子化方式、各種サンプリングレイト、モノ/ステレオ録音、このような様々なハードウェア条件の詳細に依存する部分は、利用者に多大な負担を強いることになる。

それ故、オブジェクトをリクエストしたハードウェア環境に合わせて音データを変換してくれる処理をメソッドが持つべきである。

3. マルチメディアインタフェースの設計

3.1 マルチメディアインタフェースの構成例

著者らが今までに研究対象としてきたマルチメディアデータベースを具体例にして、インタ

complex object

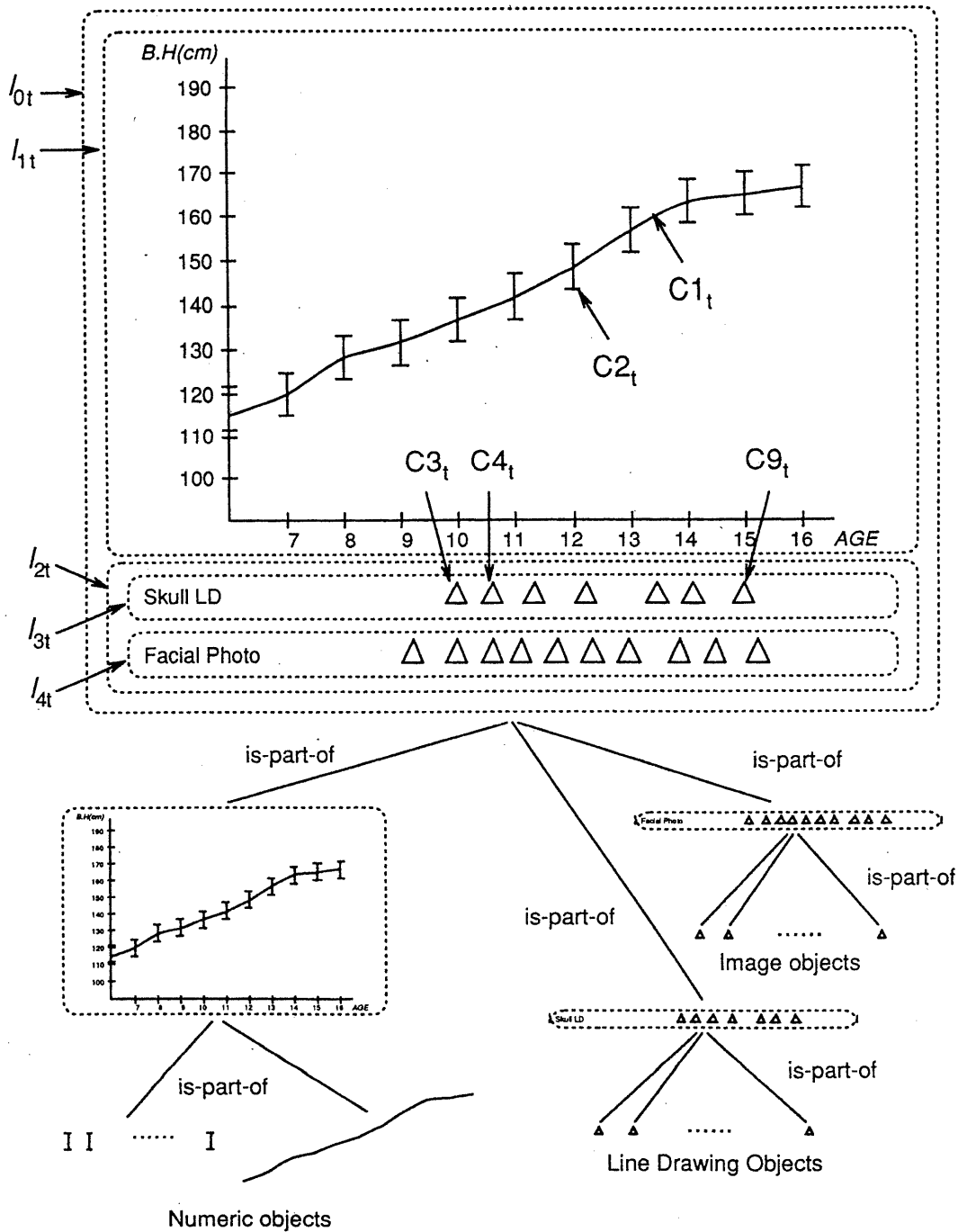


図4 マルチメディアインタフェースにおけるオブジェクト構成例 (症例データ)

フェースの設計を試みた。データベースは歯科矯正学の症例であり、画像、図形、数値の履歴データからなる [1、2]。

図4に、言語MMLで記述した結果を示す。

この図は、1症例のマルチメディアの履歴データを表す。横軸に時間（患者の年齢）を、縦軸に身長を取って描いたグラフと検査の画像 (Facial Photo)、図形 (Skull LD) データを何時採取したかを示す△印とから構成されている。

レイアウトは l_{0t} から l_{4t} の5個である。各レイアウトには、内容 C_{jt} が入っている。この図を is-part-of の関係で捉えることができる。画面全体のレイアウト l_{0t} の内容は、複合オブジェクト (Complex Object) になっている。各レイアウト l_{1t} から l_{4t} の内容は、複合オブジェクトを構成する1つの部品オブジェクトになっている。すなわち、画像、図形、数値集合のオブジェクトである。さらに、これらの部品オブジェクトは、個々の画像、図形、数値オブジェクトとなる。

ユーザは、複合オブジェクトを表す画面を見ながら、個々のオブジェクトをマウスでクリックして、イベントを発生し、データを可視化することができる。マルチメディアインタフェースプロセッサが図のような is-part-of の構造を管理しているから、ユーザが与えたイベントを解析してデータベースからオブジェクトを検索するようにリクエストを出す。

レイアウト内での並び順序が履歴を表している。それ故、言語によるインタフェース記述は、オブジェクトの時間管理にも有効である。

3. 2 画像、図形オブジェクトの操作

マルチメディアインタフェースのより高度な機能として、画像、図形オブジェクトにいろいろな操作を加える処理がある。



図5 画像と図形の重ね合わせ

例えば、図5に示すように画像オブジェクトに図形オブジェクトを重ねる操作である [2]。このような例では、オブジェクトのメソッドによる操作とアプリケーションプログラムでの処理を組み合わせることになる。

図3のワークステーションの上で動くものとする、操作の流れは以下ようになる。この一連の流れの中でマルチメディアインタフェースプロセッサ (MIFP) が果たす役割を理解することができる。

```
Superimpose () {  
    sync_check (request (LD, getdate),  
                request (PHOTO, getdate));  
    display (request (PHOTO, show, . . .));  
    display (request (LD, co_trans,  
                    get_trans_point(), . . .));  
}
```

これは、アプリケーションプログラムである。MIFPがプログラムの実行を管理する。

(1) ユーザがプルダウンメニューから Superimpose()を選ぶ。まず、sync_checkの部分が動く。重ね合わせする画像Photo, 図形LDの△印をクリックする。

(2) このイベントをMIFPが受け取る。MIFPはオブジェクトの日付をODBから検索するようにLOMにリクエストを出す。ODBがリクエストを受け取る。

(3) 検索したPhotoオブジェクトの日付とLDオブジェクトの日付の時間同期をアプリケーションプログラムでチェックする。時間同期が正しいなら次のdisplayに行く。

(4) Photoオブジェクトを検索するようにMIFPはLOMにリクエストを出す。検索した画像オブジェクトをメソッドで標準フォーマットに変換し、かつディスプレイのハード環境に合わせてデータを変換する。それをLOMに渡す。ここで、・・・の部分はディスプレイの環境を与えるパラメータである。

(5) MIFPは、画像オブジェクトの表示メソッドを起動する。表示終了イベントをMIFPに与える。

(6) MIFPは次のdisplayを起動する。ユーザが重ね合わせの基準点をPhoto上でクリックして与える。MIFPがリクエストをLOMに出す。LDの拡大、縮小、座標変換をオブジェクトのメソッドがする。LDオブジェクトのメソッドがPhotoの上にLDを重ねて表示する。終了イベントをMIFPに出す。

4. おわりに

マルチメディア言語を使用してマルチメディアデータベースのためのユーザインタフェースを一般的に設計する方法を提案した。異種オブジェクト指向データベース環境におけるマルチメディアインタフェースアーキテクチャについても議論した。今後、言語の詳細な仕様や、マルチメディアインタフェースプロセッサの仕様についてさらに検討が必要である

本研究は、1 部分文部省科学研究費課題番号 0 4 6 8 0 0 2 4 による援助を受けた。

参考文献

- [1] 金森吉成、脇山俊一郎、増永良文：
オブジェクト指向データベースの医療への応用、情報処理学会データベースシステム研究会資料、86-1 (1991)
- [2] 脇山俊一郎、金森吉成、増永良文：
オブジェクト指向データベースシステムによる医療画像データベースの実装、情報処理学会データベースシステム研究会資料、87-10 (1992)
- [3] Response to the Object Management Group, Request for Proposal for an Object Request Broker, HyperDesk Co.,
OMG TC Document 91.1.6, Part A
January, 1991
- [4] Masunaga, Y.: An Object-Oriented Approach to Multimedia Database Organization and Management,
DASFA, 190-200, April, 1989
- [5] 文書交換形式 (ODA) 実装規約
日本規格協会、1991. 3