

分節化に基づいたマルチメディア・データのモデル化

牛尼 剛聡 渡邊 豊英

名古屋大学大学院工学研究科情報工学専攻

映像データ、画像データ、音声データ等のマルチメディア・データは時間的 / 空間的な広がりを持ち、その時間的部分区間 / 空間的部分領域によって異なる意味内容を表現する。マルチメディア・データを計算機上で管理・利用する際にはマルチメディア・データの全体と部分を適切にモデル化する必要がある。これまで、オブジェクト指向データモデルがマルチメディア・データの管理に有効であるとされてきたが、オブジェクト指向データモデルにおける全体と部分とのモデル化は複合オブジェクトを用いて部分を組織化するボトムアップ的な手法に基づいている。我々はトップダウン的に全体と部分をモデル化可能なサブジェクト指向データモデルを提案し、マルチメディア・データの全体から部分を生成する操作である分節化の概念を定義し、サブジェクト指向データモデルで分節化を実現する。

Multimedia Data Modeling on the Basis of Segmentation Concept

Taketoshi USHIAMA Toyohide WATANABE

Department of Information Engineering, Graduate School of Engineering,
Nagoya University

Multimedia data are effective to represent different aspects of distinct entities in the real world according to their temporal and spatial characteristics. In this case, the relationship "Is-Part-Of" is successful to specify such temporal and spatial characteristics. Until today, the object-oriented data model has been considered as a candidate for managing multimedia data, but it can provide only a modeling facility for the relationship Is-Part-Of in a bottom-up manner. In this paper, we propose first the subject-oriented data model which provides the facility to define/manipulate the relationship Is-Part-Of among a collection of data in a top-down manner, and then introduce the concept of segmentation for multimedia data. Under these preparations, we describe how to apply the segmentation concept to the subject-oriented data model with a view to preserving the temporal and spatial characteristics of multimedia data among individually distinguished data segments.

1 まえがき

近年、映像データ、画像データ、音声データなどのマルチメディア・データが計算機上で一般に利用されるようになった。マルチメディア・データの論理的な構造は時間的な広がり、空間的な広がりを有するために、一つのマルチメディア・データが表現している意味内容は時間的な区間や空間的な領域によって異なる。従って、マルチメディア・データの管理や利用の対象はマルチメディア・データ全体のみならず、その時間的な部分区間や空間的な部分領域であることが要求される。このようなことから、マルチメディア・データの全体と部分の意味内容に基づいたモデル化が重要である。

一般に、データの全体と部分とをモデル化する手法は大別して次の二種に分類される。

1. 部分を組織化して全体を構成するボトムアップ的な手法
2. 全体の特徴に基づいて部分を生成するトップダウン的な手法

ボトムアップ的な手法では全体が部分によって規定される。これは、同種/異種の複数のマルチメディア・データを統合する際に有効である。一方、トップダウン的な手法では全体が部分を規定する。これは、マルチメディア・データの内容記述時の対象部分の指定や、編集操作時のマルチメディア・データの部分の切出しに有効である。

これまでに、マルチメディア・データの管理に関してオブジェクト指向データモデル [1] の有効性が主張されてきた [2,3]、その理由の一つは、それぞれのマルチメディア・データをオブジェクトとして表現することによってオブジェクト間の相互参照が可能になり、マルチメディア・データ間の複雑な概念的・論理的な関係を表現できることであった。オブジェクトの相互参照によるマルチメディア・データの全体と部分のモデル化は複合オブジェクトを利用して行なわれる。複合オブジェクトはすでに存在しているオブジェクトをタプル、集合、リストなどに組織化したものであり、複合オブジェクトを用いたモデル化は、部分から全体を構成するボトムアップ的な手法である。しかし、オブジェクト指向データモデルではトップダウン的に全体と部分をモデル化する一般的な機構は提供されていない。

本稿では、マルチメディア・データの全体と部分を

トップダウン的にモデル化するための手法を提案する。まず、全体と部分をトップダウン的にモデル化可能なデータモデルとしてサブジェクト指向データモデルを提案する。また、マルチメディア・データの論理構造に基づいて全体から部分を生成する操作であるマルチメディア・データの分節化を定義する。そして、サブジェクト指向データモデルを用いてマルチメディア・データの分節化を実現する方法を述べる。

2 関連する研究

Woelk ら [2,4] はオブジェクト指向アプローチを用いてマルチメディア・データを管理する手法を提案している。この手法は種々の形態のデータをそれぞれ異なるオブジェクトとして表現し、それらの集約と関連によって組織化するボトムアップ的な手法に基づいている。

OVID [5] はビデオ・オブジェクト・データモデルに基づいて実装されているビデオ・データベースシステムである。ビデオ・オブジェクト・データモデルは実世界上の映像とその映像を含むシーンを共にビデオ・オブジェクトとしてモデル化する。ビデオ・オブジェクトはオブジェクト識別子と時間区間と属性と値の集合である。それぞれの映像はオブジェクト識別子によって区別され、一つの映像のそれぞれのシーンは表示区間によって区別される。このデータモデルでは、表示区間の包含関係に基づいて、同一映像を表す複数のビデオ・オブジェクトの間で動的に属性値を継承する機構を提供しており、トップダウン的に映像データの構造を記述可能である。しかし、ビデオ・オブジェクト・データモデルは型 (クラス) の概念を持たないため、大量のデータを扱うには不適切である。さらに、それぞれのビデオ・オブジェクトの内容は汎化階層に埋め込まれる必要があるが、汎化階層が大規模になるにしたがって管理や更新が困難になると考えられる。

多態オブジェクト・モデル [6] では、一つのオブジェクトがそれぞれ相異なるクラスに属する複数のインスタンスを保持可能である。一つのオブジェクトが有するインスタンスはクラスを指定することによって同定される。従って、多態オブジェクト・モデルは、一つの実体が相異なる種類のマルチメディア・データを用いて表現される場合のモデル化に適する。しかし、一つのオブジェクトを構成するそれぞれインスタ

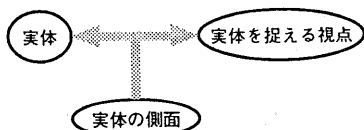


図 1: モデル化の概念

ンスはクラス名によって識別されるため、一つのオブジェクトが同一のクラスに属する複数のインスタンスを持つことが不可能である。これは、一つの実体が複数の同種のデータによって表現されている場合に対処できない。

3 モデル化の基本的な枠組

我々は、全体と部分をトップダウン的にモデル化するために、全体からの部分の生成操作を「対象に対する側面の認識」と捉える。具体的には、「対象」、「捉える視点」、および「対象の側面」の3つの概念を用い、「対象の側面」を「対象」と「捉える視点」との関連に対応付ける（図1を参照）。この概念に基づいてマルチメディア・データの全体と部分をトップダウン的にモデル化する際には、マルチメディア・データを一つの「対象」、マルチメディア・データが表現している内容を「捉える視点」とし、マルチメディア・データの時間的な部分区間 / 空間的な部分領域を「対象の側面」とする。具体例を図2に示す。図2はある画像データを「対象」とし、そこに表現されている「りんご」と「バナナ」が「捉える視点」である。画像と「りんご」との関連に画像データ中のりんごを表現する部分領域を対応付けている。同様に、その画像と「バナナ」との関連に画像データ中のバナナを表現する部分領域を対応付けている。

4 サブジェクト指向データモデル

サブジェクト指向データモデルはオブジェクト指向データモデル [1] の拡張として定義可能である。本稿ではオブジェクト指向データモデルの代表として ODMG オブジェクト・モデル [7] を採用し、それに対する拡張という形式で説明する。対象として ODMG オブジェクト・モデルを採用した理由はそれがオブジェクト指向データベースの標準として提案されている ODMG-93 の一角をなすものであり、オブジェクト指向データモデルの基本的かつ一般的な機能をほ

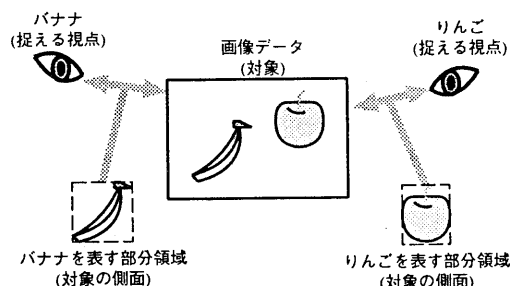


図 2: マルチメディア・データの内容記述の例

は全て有していると考えられることにある。以下に、ODMG オブジェクト・モデルの概要を示す。

- モデルの基本単位はオブジェクトである。
- オブジェクトはいくつかの型に分類できる。ある特定の型の全オブジェクトは、共通のふるまいと共通の状態の範囲を示す。
- オブジェクトのふるまいは、その型のオブジェクトに対して実行され得る操作の集合により定義される。
- オブジェクトの状態はオブジェクトがプロパティの集合に格納する値により定義される。これらのプロパティは、オブジェクト自身の属性か、一つ以上の他のオブジェクトとの関連のどちらでも良い。

これらに基づきサブジェクト指向データモデルを以下に定義する。

4.1 セグメント

サブジェクト指向データモデルでは、モデルの新しい基本単位としてセグメントを導入する。すなわち、我々のデータモデルにおけるモデル化の基本単位はオブジェクトとセグメントである。セグメントは以下のように定義される。

【定義 1】 セグメント

セグメントはソースとサブジェクトとの間の1対1関連である。ソースはオブジェクトか他のセグメントであり、サブジェクトはソース以外のオブジェクトである。また、オブジェクトと同様に、セグメントは状態とふるまいを持ち、型に分類される。セグメントはそれが生成された時点でソースの状態とふるまいを全て

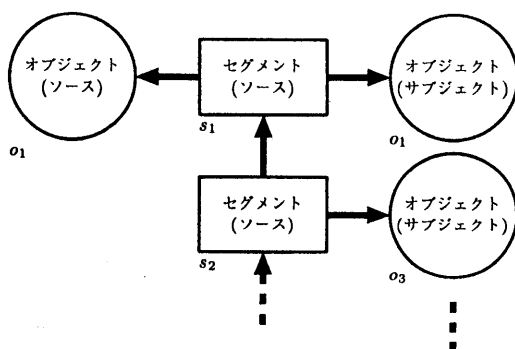


図 3: セグメントの構造

継承する。継承した値はセグメントにおいて再定義可能である。 □

ソースとサブジェクトはある一つのセグメントに注目したときに定まる相対的な概念である。図 3 にサブジェクトの構造の例を示す。図中の円と長方形はそれぞれオブジェクトとセグメントを表し、セグメントから引かれた矢印はセグメントを関連として捉えたときにそれを構成する要素を示す。セグメント s_1 はオブジェクト o_1 をソースとして、オブジェクト o_2 をセグメントとして定義されている。同様に、セグメント s_2 はセグメント s_1 をソースとして、オブジェクト o_3 をセグメントとして定義されている。さらに、セグメント s_2 をソースとしてセグメントを定義することも可能である。このように、セグメントは他のセグメントのソースになることが可能であるため、全てのセグメントは、ソース-セグメントの関係に基づいた木構造の中に埋め込まれる。データベース内には、こうした木構造が複数存在しても構わない。

サブジェクト指向データモデルでは、オブジェクトは実世界の対象に対応し、セグメントは対象のある視点から捉えた側面に対応する。つまり、サブジェクトの定義におけるソースは対象に対応し、サブジェクトはその対象を捉える視点を表し、サブジェクトは結果として得られた対象の側面に相当する。

オブジェクトの有する重要な性質の一つにオブジェクト識別性がある。オブジェクト識別性により、オブジェクトは状態の値によらず一意に識別可能であるこ

とが保証される。これは、実世界上の実体に対して、その付随的な性質が変化しても存在の同一性を認める人間の直観に一致する。そこで、オブジェクトに対するオブジェクト識別性のように、セグメントに対して実体の側面に対する人間の直観に一致する性質を付加することが重要である。そのために、セグメントに 3 種類の識別性を付加する。

【定義 2】 セグメントの有する識別性

セグメントは以下に示す 3 つの識別性を有する。

【セグメント識別性】 他のオブジェクトとセグメントからそのセグメント自身を一意に識別する性質。

【ソース識別性】 そのセグメントのソースが常にデータベース内に存在することを保証し、それを一意に識別する性質。

【サブジェクト識別性】 そのセグメントのサブジェクトが常にデータベース内に存在することを保証し、それを一意に識別する性質。

□

サブジェクト指向データモデルでは、オブジェクト識別性とセグメントの有する 3 種の識別性の全てを実現する必要がある。通常、オブジェクト指向データモデルでは、オブジェクト識別性を実現するためにオブジェクト識別子が用いられる。オブジェクト識別性の実現はそれぞれのオブジェクトを一次的に区別することであるため、オブジェクト識別子はそれぞれを区別可能な表現であることだけが要求される。一方、セグメントの有する 3 種の識別性はオブジェクト識別子と同様な一次的な表現では実現不可能である。そこで、我々はセグメント識別子を導入し、それをセグメントに与えることによってセグメントの識別性を実現する。セグメント識別子は、オブジェクト識別性がオブジェクト識別子によって実現されていることを前提にして、オブジェクト識別子を定められた構造に従って組織化したものである。

【定義 3】 セグメント識別子

セグメント識別子は以下のように再帰的に定義される。

1. 2 つのオブジェクト識別子 oid_1, oid_2 の対 (oid_1, oid_2) はセグメント識別子である。
2. セグメント識別子 sid , オブジェクト識別子 oid の対 (sid, oid) はセグメント識別子である。

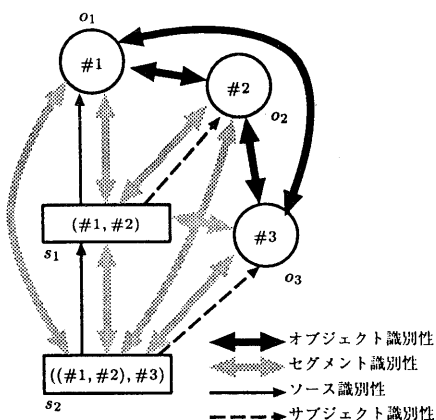


図 4: 識別性

それぞれのセグメントが有するセグメント識別子は、そのソースのオブジェクト / セグメント識別子が第一要素として与えられ、そのサブジェクトのオブジェクト識別子が第二要素として与えられる。これにより、セグメントの識別性は全てオブジェクト識別子とセグメント識別子から与えることが可能である。すなわち、それぞれのセグメントに対して、そのセグメント識別子全体がセグメント識別性を与え、セグメント識別子の第一要素がソース識別性を与え、第二要素がサブジェクト識別性を与える。

オブジェクトとセグメントが有する識別性の例を図4に示す。ここで、円と円内の文字はそれぞれオブジェクトとオブジェクト識別子を、長方形と長方形内の文字はそれぞれセグメントとセグメント識別子を表す。太い黒矢印、太い灰色矢印、細い実線の矢印および細い破線の矢印は、順にオブジェクト識別性、セグメント識別性、ソース識別性およびサブジェクト識別性を表す。

セグメントが有する3つの識別性により、ソースとサブジェクトの識別子から、それらを用いて生成されたセグメントを一意に同定可能であることが保証される。従って、セグメントの同定にソースとサブジェクトがセグメントの識別子を直接保持する必要がない。このことによって、新しくセグメントを生成する際にソースとサブジェクトを変更する必要がないという性

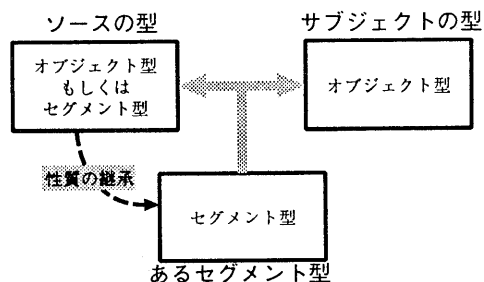


図 5: セグメントの型

質が導かれ、全体と部分との関係をトップダウン的にモデル化可能となる。

4.2 セグメント型

オブジェクトが型に分類されるのと同様に、セグメントも型に分類される。以降、セグメントが分類される型をセグメント型、オブジェクトが分類される型をオブジェクト型と呼び、両者を区別する。なお、両者を区別する必要のないときは単に型と呼ぶ。

セグメントの型はそれ自体がセグメントである。従って、サブジェクトの型はソースとサブジェクトを有する。セグメントの型が有するソースは型であり、サブジェクトはオブジェクトの型である。前者をソースの型と呼び、後者をサブジェクトの型と呼ぶ。セグメント型のインスタンス（セグメント）が有し得るソースとサブジェクトの範囲は、そのソースの型とセグメントの型によって定義される。すなわち、それぞれのセグメントが有するソースは、そのセグメントが属する型のソースの型のインスタンスでなければならず、セグメントのサブジェクトは、そのセグメントが属する型のセグメントの型のインスタンスでなければならない。また、型はそのインスタンスの状態とふるまいを定義し、インスタンスの集合を保持可能であるとする。型のインスタンスの集合をその型のエクステンと呼ぶ。

セグメントの型はソース型の性質を全て継承する。セグメントの型の概念を図5に示す。新しいセグメント型の定義にはソース型やサブジェクト型の定義の変更は必要ではないために、セグメントの型は付加的に定義できる。ふるまいは操作の集合である。ふるま

いは操作のインタフェースを表す操作シグネチャの集合として型に定義される。型はそれ自身がオブジェクト / サブジェクトであるため、状態とふるまいを有する。型はメタデータのインスタンスであり、型のふるまいはメタデータに定義される。インスタンス生成のために、セグメントの型のメタデータには次の組み込み操作が定義される。

- `create_segment(src:SRC,sbj:SBJ)->seg:SEG`
- `segment(src:SRC,sbj:SBJ)->seg:SEG`

操作 `create_segment` はセグメントの型 `SEG` のソース型 `SRC` のインスタンスとサブジェクト型 `SBJ` のインスタンスを引数としてセグメント型 `SEG` のインスタンスを生成し、そのセグメント識別子を返す。操作 `segment` は引数に与えられたオブジェクト / セグメントをソースとサブジェクトとして有するセグメントが存在するときはそのセグメント識別子を返し、存在しない場合は `nil` を返す。

セグメントの型は以下の組み込み操作を持つ。

- `source()->src:SRC`
- `subject()->sbj:SBJ`

操作 `source` はソース識別性に基づいてセグメントのソースの識別子を返し、`subject` はサブジェクト識別性に基づいてセグメントのサブジェクトの識別子を返す。

5 映像データのトップダウン的なモデル化

マルチメディア・データの全体と部分をトップダウン的にモデル化するために、マルチメディア・データの全体から部分を生成する操作を形式化することが必要である。そのために、マルチメディア・データの論理的な構造に基づいてマルチメディア・データの全体から部分を生成する操作である分節化の概念を導入する。本章では、マルチメディア・データの一つである映像データの分節化を定義し、分節化のサブジェクト指向データモデルでの実現について述べる。

5.1 映像データの分節化

映像データは2次元画像の1次元的な並びである。映像を構成する個々の画像データをフレームと呼ぶ。映像データの分節化とは映像データに対して対象フレーム区間とフレームの対象矩形領域を指定することである。形式的には、フレームの並びを表す軸を F とし、フレームの空間的な広がりを表す2つの軸を

それぞれ X , Y とするとき、対象フレーム区間を表す軸 F 上の開始フレーム f_1 と終了フレーム番号 f_2 、および X - Y 平面上の対象矩形領域の対角を示す二点 (x_1, y_1) , (x_2, y_2) を指定することである。分節化の結果として指定された映像データの部分を部分映像データと呼ぶ (図6を参照)。

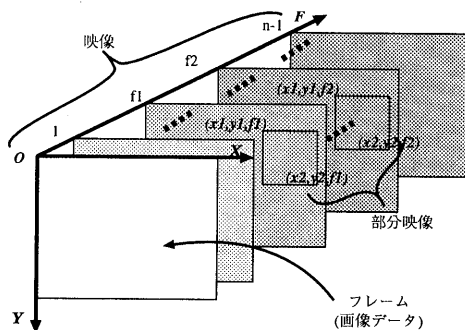


図 6: 映像の分節化

5.2 サブジェクト指向データモデルを用いた分節化の実現

サブジェクト指向データモデルにおいて分節化は以下のように実現される。

1. マルチメディア・データを属性値として含むオブジェクトを生成する。
2. そのオブジェクトをソースとするセグメントを生成する。このとき、ソースからセグメントに対して、マルチメディア・データをはじめとする属性値が継承される。
3. 時間的な区間や、空間的な領域に関する情報を、セグメントの属性値として定義する。

図7にマルチメディア・データの分節化の概念を示す。

6 スキーマと問合せの例

具体的な映像データベースのスキーマ作成例を図8に示す。図において、実線の長方形はオブジェクト型を、破線の長方形はセグメント型を、楕円はリテラル型をそれぞれ表す。また、それらの中に現れる文字は型の名前を表す。セグメント型から出る太い実線の矢印はセグメント型のソースの型を、および太い

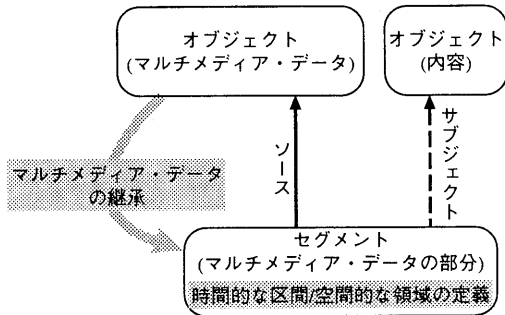


図 7: 分節化の実現

破線の矢印はセグメント型のサブジェクトの型を指している。細い実線の矢印は関連を、細い破線の矢印は属性を表している。それらの矢印に付けられたラベルはそれぞれ関連名と属性名を表す。オブジェクト型 *Movie* は一つの映像を表すオブジェクトの型であり、映像データをビット列として保持するための属性 *movie_data* が定義されている。オブジェクト型 *Person* は一人の人物を表すオブジェクトの型であり、オブジェクト型 *Action* は一つの行為を表すオブジェクトの型である。セグメント型 *Spot_Of_A_Person* は映像中に表現されている人物に基づいて分節化を行うための型であり、ソースの型として型 *Movie* を有し、サブジェクトの型として型 *Person* を有する。また、セグメント型 *Spot_Of_An_Action* はセグメント型 *Spot_Of_A_Person* において分節化された映像の部分に対して、そこに登場する人物が行う行為に基づいてさらに分節化を行うための型であり、ソースの型として型 *Spot_Of_A_Person* を有し、サブジェクトの型として型 *Action* を有する。

図 8 に示したデータベース・スキーマに対する問合せを SQL 風の表記に従って表現した例を以下に示す。

```
select aMovie
from  aMovie in Movie
      john in Person
      poul in Person
where john.name = "John"
      and poul.name = "Poul"
      and Spot_Of_A_Person.segment(aMovie,john)
                                     != nil
```

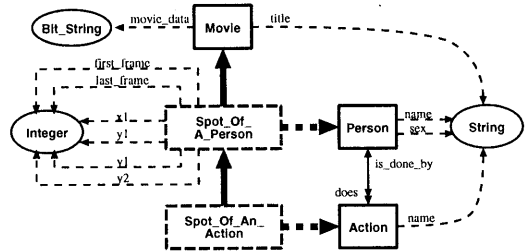


図 8: 画像データベースのスキーマ例

```
and Spot_Of_A_Person.segment(aMovie,poul)
                                     != nil
```

この質問は、John を撮影したスポットと Poul を撮影したスポットを共に含む映像データを検索する。この質問式中の Select 節は変数 *aMovie* を検索結果として返すことを示している。From 節では型 *Movie* の変数 *aMovie* と、型 *Person* の変数 *john* と *poul* を宣言している。Where 節では、変数 *john* と *poul* の表すオブジェクトが属性 *name* の値としてそれぞれ文字列 "John" と "Poul" を持つかを判定し、さらに型 *Spot_Of_A_Person* のエクステンツ内に、ソースとして変数 *aMovie* の表すオブジェクトを有し、サブジェクトとして変数 *john* が表すオブジェクトを有するセグメントが存在するか、またソースとして変数 *aMovie* の表すオブジェクトを有し、サブジェクトとして変数 *john* が表すオブジェクトを有するものが存在するかを判定している。

7 実装例

サブジェクト指向データモデルのプロトタイプを SUN SPARCstation-20 上で動作するオブジェクト指向データベース管理システム GemStone[8] 上でデータベース定義・操作言語 OPAL を用いて作成した。また、サブジェクト指向データモデルを用いた映像データベース・ブラウザを X ウィンドウ・システム上で Athena Widget を用いて C 言語で作成した。本映像データベース・ブラウザの提供する主な機能は以下のとおりである。

- データベース・スキーマとクラスのエクステンツのブラウジング
- オブジェクトとセグメントの生成、およびそれらの有する属性値と関連の定義

- 映像データの再生と分節化

図9は映像データベース・ブラウザのプロトタイプの実行例である。図中で上部に現れている最も大きいウィンドウはスキーマとクラスのエクステンツを表示する。図中で左下に現れているウィンドウで対話的にセグメントを生成し、右下に現れているウィンドウで映像の生成と対話的な分節化を行う。

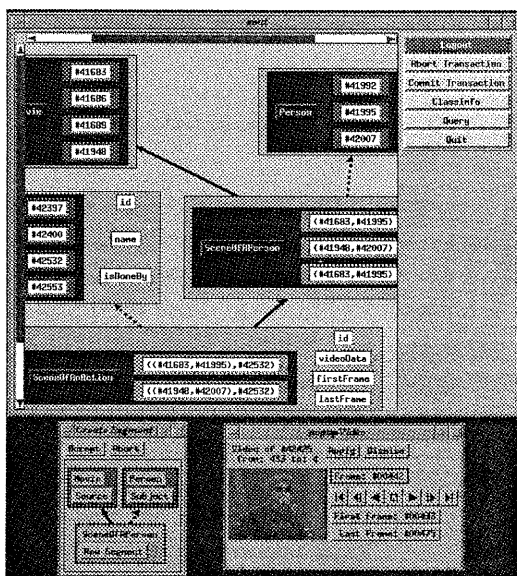


図9: 映像データベース・ブラウザ

8 あとがき

サブジェクト指向データモデルは以下の特徴により、全体と部分とのトップダウン的にモデル化を実現する。

- ソースからセグメントへのインスタンス間継承
- セグメントの識別性による全体と部分の関係の表現

マルチメディア・データの分節化はマルチメディアデータの論理構造に基づき部分を生成する操作であり、サブジェクト指向データモデルで実現可能である。

現在、以下の2点の機構を提供するためにサブジェクト指向データモデルを拡張している。

- セグメント間の演算操作によるセグメントの動的な生成機構の提供

- セグメントを用いたマルチメディア・データ間の同期機構の提供

前者は、セグメント間の演算を一つのオブジェクトとして表現し、それをサブジェクトとして演算対象のセグメントを動的に生成する。後者は、同期処理の対象となるマルチメディア・データに対する共通の軸をオブジェクトとして表現し、その軸をサブジェクトとして統合する対象のセグメントを生成する。生成されたセグメントは共通の軸上での自らの位置情報を保持するものとする。このとき、ある軸をサブジェクトとして持つセグメントを検索することにより、構成要素間がその軸上で同期した複合オブジェクトが得られる。

参考文献

- [1] Atkinson, M. et al.: The Object-Oriented Database System Manifesto, *Proc. of DOOD'89*, pp. 40-57 (1989).
- [2] Woelk, D., Kim, W. and Luther, W.: An Object-Oriented Approach to Multimedia Databases, *Proc. of ACM SIGMOD'86*, pp. 311-325 (1986).
- [3] 増永良文: マルチメディアデータベース総論, 情報処理, Vol. 28, No. 6, pp. 671-684 (1987).
- [4] Woelk, D. and Kim, W.: Multimedia Information Management in an Object-Oriented Database System, *Proc. of VLDB'78*, pp. 319-329 (1978).
- [5] Oomoto, E. and Tanaka, K.: OVID: Design and Implementation of a Video-Object Database System, *IEEE Trans. on Knowledge and Data Engineering*, Vol. 5, No. 4, pp. 626-643 (1993).
- [6] 石丸知之, 植村俊亮: オブジェクト指向データモデルにおけるオブジェクトの多重表現, 信学論, Vol. J78-D-I, No. 3, pp. 349-357 (1995).
- [7] Cattell, R.: *The Object Database Standard: ODMG-93*, Morgan Kaufmann (1994). (河辺和宏他訳: オブジェクト・データベース標準: ODMG-93 Release1.1, 共立出版 (1995)).
- [8] Maier, D. et al.: Development of an Object-Oriented DBMS, *Proc. of OOPSLA'86*, pp. 472-482 (1986).