

## 時間依存マルチメディアオブジェクトの表現方法

増永良文

図書館情報大学

概要：本稿ではODMG-93オブジェクトモデルを拡張してマルチメディアオブジェクトをモデル化できる方法を論じている。拡張は時間の扱いに中心にして次に示す2点でなされている：(a) オブジェクトそのものの定義の拡張。これによって値が時間の経過と共に変化する属性を定義できるようになること。(b) マルチメディアオブジェクト間の時間的依存関係をオブジェクト定義に明確に記述できる記述法を導入すること。このようにマルチメディア拡張されたデータモデルをOMEGAマルチメディアオブジェクトモデルと名付けている。このモデルにより様々な時間依存オブジェクトを表現できる。

## Modeling and Describing Temporal Multimedia Objects

Yoshifumi Masunaga

University of Library and Information Science

Abstract: This paper discusses a multimedia expansion to the ODMG-93 object model. This is done from the following two points: First, we expand it to allow attributes which are defined using time-varying functions. This expansion is modified so that we can apply it to sound and video data modeling. Second, we expanded the ODMG-93 relationship definition so that our model can specify 13 temporal relationships among temporal objects. The expanded object model is named the OMEGA multimedia object model.

## 1. はじめに

オブジェクト指向データベースシステムはマルチメディアデータベースシステム、CADシステム、コンピュータ援用協調作業支援システム、など様々な先進的データベース応用を効率良くこなせるデータベースシステムとして大いに期待されている。また実際オブジェクト指向データベースシステムを使って新しいデータベース応用が開発されているのも事実である。我々もこのような期待から、オブジェクト指向データベースシステムを核にしたマルチメディアデータベースシステムOMEGA(Object-oriented Multimedia Environment for General Application)を構築中である[Masu91]。

しかし、中核となるべきオブジェクト指向データベースシステムは現在、その属性値が時間の経過と共に変化するオブジェクトや、オブジェクト間の時間的依存関係（例えば順序や同期など）を表現することが出来ず、動画や音といったその性質が時間依存しているオブジェクトを表現しようとする場合に問題が生じていた。

そこで、我々は「時間依存オブジェクト(temporal objects)」を定義することで動画や音を統一的に表現出来るマルチメディアデータベースを構成する基本枠組みとしての「時間依存拡張オブジェクト参照法」を提案した[Masu93]。さらにオブジェクト間の時間的な相互依存関係を表現するため、今回マルチメディアオブジェクト間の時間的相互依存関係の定義を、古くはAllen[Alle83]、最近ではLittleらの研究[LiGh93]を参考にしつつ、一方ODMG-93オブジェクトモデル[Cau94]で提案されたオブジェクト間の参照一貫性記述のための考え方と記述法を（上位互換に）拡張して、それを行なえるような方法を考案した。

本稿では、次章でOMEGAオブジェクトモデルの基本的枠組みを与えた後、上記の問題を3、4章に分けて論じる。その結果OMEGAマルチメディアオブジェクトモデルが明確になる。5章に今後の課題を述べる。

## 2. OMEGAマルチメディアオブジェクトモデル

### 2.1 概観

OMEGAでは「全てがオブジェクトである」というオブジェクト指向の「異質性の同質化」の特性に立脚してマルチメディアデータの組織化を行なってきた[Masu85,87,87b]。つまりテキスト、図形、静止画、動画、音といったマルチメディアデータは全て「オブジェクト」と見做されている。従って、OMEGAデータモデルは基本的機能としてオブジェクトを記述できる能力が前提になる。更に様々なマルチメディアデータを記述できるために機能拡張が必要である。このようなマルチメディアデータの記述能力を持ったOMEGAデータモデルを我々はOMEGAマルチメディアオブジェクトモデルと呼ぶことにする（簡単にOMEGAモデルという場合もある。）

さて、マルチメディアオブジェクトは従来の文字・数値データに比べて、「空間的」、「時間的」、そして「複合オブジェクト」の特徴を持つ。ここに空間的とはオブジェクトが1次元、2次元、あるいは3次元の広がりや空間的相互関連性を持つという意味である。時間的とはオブジェクトの諸性質が時間の経過と共に変化する、かつマルチメディアオブジェクト同士が時間的相互関連性を有するということである。またマルチメディアオブジェクトは元来複合オブジェクトである。例えばオーディオビジュアルオブジェクトは動画オブジェクトと音オブジェクトの複合オブジェクトである。また様々なマルチメディアオブジェクトを空間的・時間的に組み合わせて新たなマルチメディアオブジェクトを構築することができる。OMEGAマルチメディアオブジェクトモデルは上記のような特徴をモデル化できるように設計されねばならない。

### 2.2 ODMG-93オブジェクトモデル

OMEGAマルチメディアオブジェクトモデルを設計するにあたりまず問題になるのが標準的でなければならないということである。これまでSmalltalkやC++といったオブジェクト指向プログラミング言語、それらのオブジェクトを永続化できるデータベース管理システムであるオブジェクト指向データベース管理システムが幾つか開発されてきたが、不幸にして統一された標準オブジェクトモデルは存在しなかった。

しかし、最近ODMGによりODMG-93オブジェクトモデルが標準オブジェクトモデルとして提案され大きな影響を与えている[Cau94]。それはオブジェクトを定義するインタフェース定義とオブジェクトの実装(implementation)を切り離し(これはOMGがObject Request BrokerのInterface Definition language (IDL)で提案した考え方であるが)、オブジェクトとは何かを良く考察して参考するに十分値する。この意味で、OMEGAマルチメディアオブジェクトモデルはODMG-93オブジェクトモデルを基本にして、一方そこでは全く考慮されていないオブジェクトの空間的・時間的諸性質をうまくモデル化する能力を持つよう定義することとした。

ここで、以下の議論をより明確にするためここで簡単にODMG-93オブジェクトモデルの要点を示す。ODMG-93が提案しているものは、1) Object Model, 2) Object Definition Language, 3) Object Query Language, 4) Object Manipulation Language, 5) Language Binding, その他である。特にオブジェクトモデルについてまとめておけば次のようになる: オブジェクトモデルの要素はobjects, types, operations (behavior of an object), properties: attributes (properties of an object)と relationships (properties between objects)である。ここにobjectsはモデリングの基本要素(primitives)である。objectsにはmutable objectとimmutable objects (=リテラル)がある。オブジェクトはタイプに類別される。同じタイプのオブジェクト同士は共通の特性(characteristics), すなわち振る舞い(operation)と性質(property)を持つ。性質は属性(attributes)と関連性(relationship)に分類される。注意すべきはODMG-93オブジェクトモデルではクラスという言葉は使われない。タイプはそのインタフェース(interface)をオブジェクト定義言語(ODL)で指定(specify)することによって定義される。タイプインタフェース指定とこのタイプの実装の一つの組をクラスと言っている。簡単に言えば、ODMG-93オブジェクトモデルをC++にlanguage bindingした結果得られる個々のOODBMS製品上での、いわゆるクラスを言及するために使用する。

なお、ODMG-93オブジェクトモデルでは本来、例えばSmalltalkクラスが持っている二つの側面、つまりそのクラスに属するオブジェクトが共通に持つ特性としてのタイプと、そのクラスに属するインスタンスとしてのオブジェクトの集合を峻別し、後者を外延(extent)と称しているが、本稿では特に立ち入らない。タイプの定義はその構文をOMG IDLに基づいているODLを使い定義されるがその一例を次に示す。

【例題1】ODMG-93オブジェクトモデルによるオブジェクトタイプの定義例

```
interface Student: Person {
    extent students;
    attribute integer [6] student_id;
    attribute Set<struct<string degree_name, Year degree_year>> degrees;
    relationship takes: Set<Course> inverse Course::is_taken_by;
    <operation_list>
};
```

この例の簡単な説明は次のとおりである。PersonオブジェクトタイプのサブクラスとしてStudentオブジェクトタイプが定義されている。(従ってname, addressなどの属性はPersonオブジェクトタイプから継承されよう。)外延はstudentsである。属性を定義する際のattributeというキーワードの指定はオプションである。同様に関連性を定義する際のrelationshipというキーワードの指定もオプションである。この関連性はオブジェクト間の参照一貫性(referential integrity)を表わす。つまり、この例では学生オブジェクトはtakesという属性を持っているが、その属性値はCourseオブジェクトタイプのオブジェクトの集合であるが、それらのオブジェクトはis\_taken\_byという属性を持っており、それらの間にinverse関連性が指定されているので、もしある学生オブジェクトが削除されたらその学生がとっていた科目オブジェクトの属性is\_taken\_by値からその学生オブジェクトも削除すべきことを指定している。(ODMG-93オブジェクトモデルではキー(key)も指定できるがここでは省略した。)なお、本稿では以下便宜的にクラス定義という言葉は、そのタイプと外延を定義するという意味で使用することとする。

## 2. 3 OMEGAマルチメディアオブジェクトモデルの基本的特徴

本節ではOMEGAマルチメディアオブジェクトモデルの基本的な特徴をまとめて、これからの議論の助けとする。

- (1) OMEGAマルチメディアオブジェクトの定義は先述の通りODMG-93オブジェクトモデルに準拠しているが、(i)時間の経過と共にその値が変化する属性（以下これを時間変化属性という）を定義できるようにattribute定義が拡張されていること、(ii)マルチメディアオブジェクト間の時間的依存関係が記述できるようrelationship定義が拡張されていることの2点で拡張がなされている。この詳細は3章と4章で述べる。
- (2) OMEGAマルチメディアオブジェクトは一般に複合オブジェクトである。このこと自体至極当然のことであるが、次のような意味で通常のIS-PART-OF関係を越える意味( semantics)を有する。つまり、マルチメディアオブジェクトの構成要素オブジェクトは空間的・時間的な諸関係を満たしつつ部品となっていることである。これは丁度ODAマルチメディア文書の共通論理構造を表現する場合、IS-PART-OF関係にSEQ, REP, CHO, OPT等の補助記号( arguments)を添える必要性が生じること[ODA91]に類似している。つまり、マルチメディアオブジェクトを表現する場合、それを構成する構成要素がどのような空間的・時間的意味合いをもって部品となっているかを適当な補助記号を導入して記述する必要があるということである。
- (3) オブジェクトにはそれがいつ誕生しいつ死亡したかという概念をもっているものもある。例えば、ある音楽の演奏はそのような属性を持つ。このようなオブジェクトは時間区間( time interval)としてモデル化されている[Alle83]。そこでは時間区間の2項関係；equal, starts, meets, など計13個が定義されている。OMEGAオブジェクトモデルではこれらの関係が上記(1)項(ii)のマルチメディアオブジェクト間の時間的依存関係を表現するためと、上記(2)項のマルチメディアオブジェクトの要素オブジェクト間のIS-PART-OF関係を記述するための補助記号を定義する際にも使用される。ここで、時間区間オブジェクトのタイプ定義を示し、更にOMEGAテンポラルオブジェクトを定義しておく。

**【定義1】 OMEGA時間区間オブジェクトクラスの概略**

```
interface TimeIntervalObject: Object {
    attribute Time start_time_point;
    attribute Time end_time_point;
    //other attributes
    float duration() {return start_time_point - end_time_point;};
    //other operations
};
```

さて、OMEGAでは出生と死亡の概念を持つオブジェクトを生成するクラスを次のように規定する。

**【定義2】 OMEGA時間依存オブジェクトの定義**

OMEGAオブジェクトが時間依存オブジェクト( temporal objects)であるとは、そのオブジェクトタイプがタイプTimeIntervalのサブタイプであるときをいう。

以下、より詳しく時間依存(マルチメディア)オブジェクトについて論じる。

### 3. 時間依存マルチメディアオブジェクト

#### 3.1 時間依存マルチメディアオブジェクトとは

マルチメディアデータは従来の文字・数値型データと異なり、テキスト、図形、画像など空間的な広がりを持つデータと動画や音など時間の経過と共にその諸元が変化するデータがある。我々は前者のタイプのオブジェクトを「空間依存マルチメディアオブジェクト」、後者を「時間依存マルチメディアオブジェクト( temporal multimedia objects)」と言うことにする。従来のマルチメディアデータ管理はどちらかと言えば空間依存オブジェクト管理に重点がおかれており、この理由の一つにはデータベースのモデル化や管理システムの設計において「時間」という難しい問題を避けて通れることがあると考えられる。

時間依存マルチメディアオブジェクトは、より厳密には次に掲げる2つの課題がある。

- (a) マルチメディアオブジェクト自体の属性値が時間の経過と共に変化する性質をどのように表現する

か

(b) マルチメディアオブジェクト間の時間的依存関係をどのように表現するか  
本章ではまず(a)項を論じる。(b)項は次章で論じる。

### 3. 2 時変属性

動画や音といった時間依存オブジェクトは、単に出生と死亡属性を持つという特性の他に、それが生存している期間そのオブジェクトの属性値が時間的に変化するという特性を有している。例えば、(コンピュータで扱える) デジタルな表現の世界でいえば、動画はフレームと呼ばれる静止画の(1秒間に30枚の)時系列であり、音はサンプリング定理と量子化を施した結果得られるデジタル値の時系列で表わされる。

OMEGAデータモデルでは属性値が時間の経過と共に変化するオブジェクトを直接表現するメカニズムを持たせている。これは従来関数データモデルでそのような仕掛けが提案されたことがあった[WuDa92]が、ここではそれがオブジェクト指向データベースの枠組みでモデル化されている[Masu93]。

#### 【定義3 (時変オブジェクト)】

時変オブジェクト(time-varying objects)とは値が時間の経過と共に変化する特性を有している属性を1個以上有しているオブジェクトを言う。またそのようなオブジェクトを生成するタイプを時変オブジェクトタイプという。

ここで時変マルチメディアオブジェクトを”新星”オブジェクトの例を用いて段階的に示す。まず、属性値が時間的に変化しない場合、つまり輝度が一定の場合は通常のオブジェクト定義と同様に行なえる。

#### 【例題2】星クラスの定義—輝度が一定の場合—

```
interface Star {
    attribute float brightness;
    //other attributes
    float brightness() {return brightness;};
    //other operations
};
```

次に、新星のように輝度が時間的に変化する場合を考える。このとき輝度が時間関数 $f(t)$ で変化すると仮定しよう。すると輝度は上の例のようにfloat型の属性brightnessと定義したのではその時間変化を表わせないから不適當である。そこで、もし輝度が

```
float brightness=f(DateTime t);
```

と言う具合に時間関数 $f$ で定義することができれば、新星の輝度変化をオブジェクト定義に直接取り込めると考えられる。一方、新星は星である。そこで、その輝度が時間関数 $f(\text{DateTime } t)$ で変化する新星のタイプNovaを次のように定義することを考える。

#### 【例題3】新星タイプの定義—輝度が時間関数 $f(t)$ で変化する場合—

```
interface Nova: Star{
    float brightness=f(DateTime t);
    //other attributes
    float brightness(DateTime  $\tau$ ) {return f( $\tau$ );};
    //other operations
};
```

まず、タイプNovaでは属性brightnessは上書きされている。また、ある時刻 $\tau$ での輝度 $f(\tau)$ を返すoperationがbrightness(DateTime  $\tau$ )である。またクラスDateTimeはシステムに組み込みの汎用クラスライブラリに既に定義されているものとする。タイプNovaの新しいオブジェクトaNovaを生成する文は次のように書く。

```
Nova aNova=new Nova();
```

この結果、時間関数 $f(t)$ を属性brightnessの値として持つオブジェクトaNovaが生成される。このタイプNovaに示されるように、その値が時間関数で変化するよう定義された属性を時変属性(time-varying

attribute)とすることに。現在このような属性はC++でもSmalltalkでも定義されていない。なお、タイプDateTimeは実世界のデータモデル化という観点で捉えているので絶対時間（ここでは、例えば西暦年月日時分秒）を表わしているとする。

さて、上記新星の時変属性brightnessの定義；

```
float brightness=f(DateTime t);
```

という書き方はクラスNovaの新星は時間関数f(t)によって輝度に変化するものであるという規則が観測の結果見いだされている時にこのように定義できる。しかし、観測から常にそのような関数関係が見つかるかどうかは分からない。このような状況では、観測結果が”観測データ”としてデータベースに格納されていることが想定される。もしそうなら、この観測データを使用して関数関係をシミュレートすることで、時変クラスNovaを実質的に定義することができると思う。そのためまず観測データを格納するクラスの定義をする。

【定義4】 タイプObservedDataの定義

```
interface ObservedData {
    attribute Nova observed_nova;
    attribute DateTime observation_date_time;
    attribute float observed_brightness;
    //other attributes
};
```

さて、タイプNovaの新星の輝度をクラスObservedDataに蓄積された観測データをもとに次のように計算する。注意しないといけない点は例題3の場合Novaクラスの新星はすべて時間関数f(t)でその輝度に変化した。観測データを用いて輝度を決定する場合、新星毎にそれぞれ変化しているを表わす属性brightnessの値を測定して、測定データからは観測の対象となった新星と観測日の二つのパラメータを指定して輝度が検索できるように書いたSQL文を用いて決める。すると観測データを用いた場合の新星クラスの定義を次のように与えることができる。

【例題4】 新星クラスの定義—観測データを用いる場合—

```
interface Nova: Star {
    float brightness=g(DateTime d) {
        Nova *n=this;
        return query("select x.observed_brightness
                    from ObservedData x
                    where x.observed_nova =*n
                    and x.observation_date_time=d");};

    //other attributes
    float brightness(DateTime d) {return g(d);};
    //other operations
};
```

ここにquery("select x.observed\_brightness from ObservedData x where x.observed\_nova=\*n and x.observation\_date\_time=d")は観測の対象となった新星(=\*n)の観測日時(=d)の輝度を返すオブジェクトSQL文を表わすこととし、しかるべきオブジェクトSQL処理形の存在を仮定している。時間はここではクラスDateTimeで表わしている。

つまり観測データを用いた場合にも例題3に場合と同様な表現形式で、新星の輝度が時間的に変化することが表わせた。なお、観測データをもとにして属性brightnessを定義する考え方は、マルチメディアオブジェクトを構成する音や動画を、OMEGAオブジェクトモデルではあくまでメディアム毎に決まるアクセプタによる観測データにしかすぎないという解釈を与えることができ、その結果、時間拡張オブジェクト参照法と称するマルチメディアデータベースの構成法を示すことができる。

## 4. OMEGAモデルにおけるマルチメディアオブジェクト間の時間依存関係のモデル化

### 4.1 マルチメディアオブジェクト間の時間依存関係

オブジェクト間の時間依存性をどのように規定するかという問題はこれまで認知科学や人工知能の分野で研究されてきた経緯がある。例えば、Allen[Alle83]は知識表現の立場から時間区間(temporal interval)の特性を試み、before, equal, meets, overlaps, during, starts, finishesの7個の2項関係とequalを除く6個の関係の逆関係(inverse relation)からなる計13個の時間依存関係を同定し、時間線表(time line)を用いた分析を行った。マルチメディアデータの時間依存性については、上記の結果をもとにLittleら[LiGh93]が研究を行ない、一般にn-項時間依存関係を導入してマルチメディアデータ間のタイミングを表現した。例えばスライド(=イメージ)と説明(=音)を同期させながら発表を行なうという場合を”木”(tree)表現でマルチメディアデータ間の同期関係を表現する時間依存データモデルを提案している。

さて我々はマルチメディアオブジェクト間の時間依存関係(temporal relationship)を規定しようとしているが、それはより広い視野からは、一般にオブジェクト間には様々な関連性が存在するのであり、マルチメディアオブジェクト間の時間依存関係はその一例にしかすぎない。このような立場から過去を振り返ると、リレーショナルデータモデルではキー制約、ドメイン制約、外部キー制約(=参照一貫性)などがあり、標準リレーショナルデータベース言語SQLでもそれらが定義可能となっていた。一方、オブジェクトモデルでは、そもそもオブジェクトモデルの標準がないと言うハンディがあるが、ODMG-93オブジェクトモデルで参照一貫性がキーワードinverseを使って指定できるように提案されていることは大いに評価できる。つまり、本稿の研究は巨視的にはオブジェクト間に存在する様々な関連性を記述できる体系を築き上げることであるが、ここでは特にマルチメディアオブジェクト間の時間依存関係を表現するための統一的な構文をODMG-93オブジェクトモデルの上位互換で規定することを目的とする。

### 4.2 オブジェクト間の時間依存関係

先に時間依存オブジェクトを定義し、それをタイプTimeIntervalのサブタイプとして定義することを述べた(2.3節)。これは実世界が空間と時間から構成されている限り不可避な事実である。つまり時間依存オブジェクトはstart\_time\_point(以下stpと略)とend\_time\_point(以下etp)なる属性を有し、stpとetpは時刻を値として取る。2項組(stp, etp)を時間区間(time interval)と言う。必然的にこのようなオブジェクトはduration = etp - stpで定義される時間幅(time duration)を属性として持つ。

さて、m1とm2を時間依存オブジェクトとする。Allenが示した13個の時間依存関係をまとめると次のようになる：(以下m1.stpはオブジェクトm1のstp値を表わす。他も同様。)

- 1) equal(m1, m2) iff m1.stp=m2.stp and m1.etp=m2.etp
- 2) before(m1, m2) iff m1.etp<m2.stp
- 3) after(m1, m2) iff m2.etp<m1.stp
- 4) during(m1, m2) iff m2.stp<m1.stp and m1.etp<m2.etp
- 5) contains(m1, m2) iff m1.stp<m2.stp and m2.etp<m1.etp
- 6) overlaps(m1, m2) iff m1.stp<m2.stp<m1.etp and m2.stp<m1.etp<m2.etp
- 7) overlapped\_by(m1, m2) iff m2.stp<m1.stp<m2.etp and  
m1.stp<m2.etp<m1.etp iff overlaps(m2, m1)
- 8) meets(m1, m2) iff m1.etp=m2.stp
- 9) met\_by(m1, m2) iff m2.etp=m1.stp iff meets(m2, m1)
- 10) starts(m1, m2) iff m1.stp=m2.stp
- 11) started\_by(m1, m2) iff starts(m2, m1)
- 12) finishes(m1, m2) iff m1.etp=m2.etp
- 13) finished\_by(m1, m2) iff finishes(m2, m1)

なお、10)と11)および12)と13)については形式上の違いのみであることに注意する。この意味では時間依存オブジェクト間の時間区間関係は11個あるといえる。

我々は医療データの効果的な組織化と検索のため治療（オブジェクト）を時間区間として捉えて定式化する研究に着手している。そこでは治療が行なわれなかった時間区間も治療が行なわれた時間区間と同じように大事であるとの認識から”空時間区間” (null time interval)を導入している[FAKM94]。この概念は複合マルチメディアオブジェクトを構築する際大事な働きをする。

#### 4. 3 ODMG-93オブジェクトモデルにおける参照一貫性の表現

ODMG-93ではオブジェクト間の参照一貫性を表現するのに関連性inverseを導入した。先に例を示したように、StudentタイプオブジェクトとCourseタイプオブジェクト間の参照一貫性を表すのに、タイプStudentのインタフェース定義で次の関連性を定義する一方、タイプCourseのインタフェース定義でも相対する関連性定義をすることにより、学生と科目間の参照一貫性を表現した。

【例題5】ODMG-93オブジェクトモデルにおける参照一貫性の定義例

```
interface Student: Person {
    .....
    relationship takes: Set<Course> inverse Course::is_taken_by;
    .....
};
interface Course {
    .....
    relationship is_taken_by: Set<Student> inverse Student::takes;
    .....
};
```

ここで注意すべきは、inverseの関連性は直接参照関係にある2個のオブジェクト間のしかるべき属性、この例題の場合はtakesとis\_taken\_by、の間で定義されていなければならない。ちなみにODMG-93オブジェクトモデルでの関連性定義のBNFの概略は次のとおりである：

【定義5】ODMG-93オブジェクトモデルでの参照一貫性のBNFの概略

```
<relationship_spec> ::= [relationship] <target_of_path>
    <traversal_path_name_1> inverse <inverse_traversal_path>
    <traversal_path_name_1> ::= <string>
    <target_of_path> ::= <target_type> | <collection_type><<target_type>>
    <target_type> ::= <type_name>
    <inverse_traversal_path> ::= <target_type> :: <traversal_path_name_2>
    <traversal_path_name_2> ::= <string>
```

しかし、マルチメディアオブジェクト間の時間依存関係を表現しようとする、時間依存関係にあるオブジェクトは様々な存在しているから上記参照関係に見るようにオブジェクト間の参照関係によるオブジェクト間の直接的な繋がりを記述できればよいとする能力では不十分である。換言すれば、マルチメディアオブジェクト間の時間依存関係を記述するにはODMG-93オブジェクトモデルを少なくとも次の2点で拡張しなければならない：

- (a) traversal\_pathの定義を拡張して複合マルチメディアオブジェクトの任意の構成要素オブジェクト間の時間依存関係を記述できるようにすること。
- (b) inverseに加えて 1) equal, 2) before, 3) after, 4) during, 5) contains, 6) overlaps, 7) overlapped\_by, 8) meets, 9) met\_by, 10) starts, 11) started\_by, 12) finishes, 13) finished\_byなる時間依存関係を定義できるようにすること。

そこで次節にこの2つの条件を満たすOMEGAマルチメディアオブジェクト間の時間依存関係を表わすBNF定義の概略と典型例を示す。



#### 4.4 OMEGAにおける時間依存関係の表現法

【定義6】 OMEGAオブジェクトモデルにおける時間依存関係のBNFの概略

```
<relationship_spec> ::= [relationship] <target_of_relationship>
    <forward_traversal_path_name> <relationship_name>
    <backward_traversal_path>
<relationship_name> ::= inverse | equal | before | after | during |
    contains | overlaps | overlapped_by | meets | met_by |
    starts | started_by | finishes | finished_by
<target_of_relationship> ::= <target_class> |
    <collection_class><target_class>
<forward_traversal_path_name> ::= <traversal_path_name> |
    <traversal_path_name>.<forward_traversal_path_name>
<backward_traversal_path> ::= <target_class>::<backward_traversal_path_name>
<backward_traversal_path_name> ::= <traversal_path_name>|
    <traversal_path_name>.<backward_traversal_path_name>
```

ここで、複合マルチメディアオブジェクトを例にして時間依存関係の表現を示す。例えば、あるオーケストラの演奏を、一つは「音」オブジェクト、もう一つは「動画」オブジェクトからなるオーディオビジュアルオブジェクトと捉えたとき、構成要素としての音オブジェクトと動画オブジェクトは時間依存関係として、“同じ時刻に始まり”、“同じ時刻に終了する”という関係（つまり2項関係equal）を満たさないとはいけない。OMEGAでは上記の関係記述定義でそれが行なえるということである。その概略を次に示す。

【例題6】 equal関係を用いたマルチメディアオブジェクトの時間依存関係の定義の一部

```
interface Music :public TimeInterval {
    attribute char* title;
    attribute SoundCompo s_compo;
    attribute VideoCompo v_compo;
    //other attributes
    relationship Sound s_compo inverse Sound::is_s_compo_of;
    relationship Video v_compo inverse Video::is_v_compo_of;
    //other operations
};

interface Sound: public TimeInterval{
    attribute Music is_s_compo_of;
    //other attributes
    relationship Music is_s_compo_of inverse Music::s_compo;
    relationship Video is_s_compo_of.v_compo equal
        Video::is_v_compo_of.s_compo;
    //operations
};

interface Voice: public TimeInterval{
    attributes Music is_v_compo_of;
    //other attributes
    relationships Music is_v_compo_of inverse Music::v_compo;
    relationship Sound is_v_compo_of.s_compo starts
        Sound::is_s_compo_of.v_compo;
    //operations
};
```

## 5. 今後の課題

### (1) マルチメディアクラスライブラリの整備

マルチメディアデータをモデル化するには本稿で示した2つの拡張だけではまだ不十分であると認識している。つまり、我々はマルチメディアデータの表現に適したクラス階層も提供することが大切である。クラス階層には大別すると、システム組み込みの基本的な「汎用クラスライブラリ」、応用ドメインをサポートするための「特殊クラスライブラリ」、そしてユーザがある特殊な応用目的で構築する「独自クラスライブラリ」がある[Tay90]。汎用クラスライブラリはリテラルやアレイ、セット、バッグなどの基本的なクラスを提供するが、マルチメディアデータの表現に必要なクラスは提供していない。そこでそのようなクラスは特殊クラスとして、通常クラスベンダから提供されるべきものであるが、現在それも十分なものはない。従って、マルチメディアデータを組織化するにはそのための特殊クラスを整備しておく必要がある。マルチメディアデータ組織化のための基本クラスライブラリの設計は今後の最重要課題である。

### (2) OMEGAマルチメディアオブジェクト質問言語の設計

ODMG-93 OQLと両立したマルチメディアオブジェクト質問言語(MQL)を設計できることになる。ここでは、マルチメディアオブジェクト間の時間依存関連性を使用した“質問”が表現できねばならない。例えば、あるオブジェクトとstart time pointが同じ別のオブジェクトを求めるなどという質問を属性stpやetpなどではなく、時間依存関係startsを使って直接書ける構文を定義することなどがポイントとなる。今後速やかにこの問題に取り組む計画である。

【謝辞】 OMEGAプロジェクトを遂行してくれている増永研究室の諸氏、ご激励を賜る様々な方に心より感謝申し上げます。

### 【文献】

- [Alle83] J. F. Allen: Maintaining Knowledge about Temporal Intervals, Communications of the ACM, Vol.26, No.11, pp.832-843, 1983.
- [Catt94] Cattell, R. (ed.): The Object Database Standard: ODMG-93, 169p., Morgan Kaufmann, 1994.
- [FAKM94] 福田紀彦, 天竺俊之, 金森吉成, 増永良文: オブジェクト指向データベースにおける時間軸データモデル, 情報処理学会データベースシステム研究会, 研究報告DBS98-1, 8p., 1994.5.
- [LiGh93] T. D. C. Little and A. Ghafoor: Interval-Based Conceptual Models for Time-Dependent Multimedia Data, IEEE Transactions on Knowledge and Data Engineering, Vol.5, No.4, pp.551-563, 1993.
- [Masu85] 増永良文: マルチメディアデータベースシステムの概念設計, 図書館情報大学研究報告, Vol. 4, No. 1, pp.9-26, 1985.
- [Masu87b] 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.
- [Masu87] Y. Masunaga: Multimedia Database: A Formal Framework, Proc. IEEE Computer Society Symposium on Office Automation, pp.36-45, 1987.
- [Masu91] Y. Masunaga: Design Issues of OMEGA: An Object-Oriented Multimedia Database Management System, Journal of Information Processing, Vol. 14, No. 1, pp. 60-74, 1991.
- [Masu93] 増永良文: マルチメディアデータモデルOMEGAにおける音データと動画データへのサポート, 情報処理学会データベースシステム研究会, アドバンスト・データベースシステム・シンポジウム'93会議録, pp. 163-179, 1993.12.
- [ODA91] 開放型システム間相互接続の基本モデル JIS X 5003-1987 参考 S 007(v2.0) 文書交換形式 (ODA)実装規約 規約編, 日本規格協会, 230p., 1991.
- [WuDa92] G. Wu and U. Dayal: A Uniform Model for Temporal Object-Oriented Databases, Proc. of Eighth International Conference on Data Engineering, pp. 584-593, 1992.