

# スキーマダイアグラムを用いたマルチメディアデータベース 検索手法

岸邊 佳世子 † 家富 誠敏 †† 富井 尚志 ††† 有澤 博 †††

† 横浜国立大学大学院 環境情報学府  
†† 横浜国立大学 エコテクノロジーシステムラボラトリー  
††† 横浜国立大学大学院 環境情報研究院  
〒240-8501 横浜市保土ヶ谷区常盤台 79-7  
E-mail : {kayo, eto, tommy, arisawa}@arislab.dnj.ynu.ac.jp

## あらまし

本論文では、マルチメディアデータベースの利用者がスキーマやインスタンスの状況を直観的に把握できることを目的として、データベースを視覚化し、利用者が見たい・知りたいデータを自由にブラウジングすることができるインタフェースを提案する。マルチメディアデータベースでは複雑なスキーマに沿って大量のインスタンスが蓄積されている場合が多い。その際、スキーマ全体、インスタンスの全てを限られた画面上に視覚化することは物理的にも不可能であると同時に、必ずしも利用者の直観的なブラウジングを手助けすることにならない。特に、その大量のインスタンスの中から利用者が必要とするインスタンスを探し出すのは大変困難である。そこで本論文では、インタフェースを用いてマルチメディアデータベースを視覚化するとともに、そのスキーマおよびインスタンスのうち、必要とする部分を自由に抜き出して視覚化する手法として“ふるい”操作の提案を行う。

## Diagram-based Querying for Multimedia Database

Kayoko KISHIBE †, Masatoshi IETOMI ††, Takashi TOMII †††, and Hiroshi ARISAWA †††

† Graduate School of Environment and Information Sciences, Yokohama National University  
†† Ecotechnology System Laboratory, Yokohama National University  
††† Faculty of Environment and Information Sciences, Yokohama National University  
79-7, Tokiwadai, Hodogaya-ku, Yokohama 240-8501 JAPAN  
E-mail: {kayo, eto, tommy, arisawa}@arislab.dnj.ynu.ac.jp

## Abstract

In this paper, we propose the interface which the user can browse the visualized schema and instances, for the purpose of the user can grasp intuitively the schema and instances of multimedia database. In many cases, a lot of instances are accumulated along with the complicated schema in the multimedia database. In that case, it is physically impossible to visualize all the schema and instances on a screen, and a user becomes unable to perform browsing intuitively. It is very difficult especially to discover the instance which a user needs out of huge instances. Then, in this paper, while a multimedia database is visualized using an interface, as a method of the operation called “sift operation” that extracts freely the portion needed among the schema and an instance, is proposed.

## 1 はじめに

近年の計算機技術の発達により、文字、数値だけでなく、画像、映像、音声、3次元グラフィックスなどの大容量なマルチメディアデータを計算機上で取り扱うことが可能となってきた。それに伴い、マルチメディアデータを統括的に蓄積、管理、運用するためのデータベース、いわゆるマルチメディアデータベースの重要性が増してきている。

ところでマルチメディアデータベースに限らず、データベースを利用する際には、データベースの構造(スキーマ)や入力されている値(インスタンス)の状況を把握することが重要である。データベースの中にどのようなデータがどのように蓄積されているかを知ることによってデータベースを有効に活用することが可能になるからである。しかし、マルチメディアデータベースにおいては、映像、音声、画像などの様々なデータを、データを解釈して得られる多様なインデックス情報とともに蓄積することが求められ、そのためスキーマは一見複雑な構造となることが多い。このような一見すると複雑なデータベースのスキーマ、さらにインスタンスの状況を一般利用者が把握することは困難を伴うことが予想される。

これを解決するためには、マルチメディアデータベースのスキーマやインスタンスを視覚的に提示することで利用者の直観的理解を助ける手法が考えられる。特に画像や映像、グラフィックスのように視覚的なデータを扱うマルチメディアデータベースであればこそ、スキーマやインスタンスを視覚的表示することにより、より直観的な情報の把握が可能になると期待される。

このような考えに基づき我々は、拡張したERダイアグラムを用いてスキーマやインスタンスを概念構造そのままに視覚的表示するユーザインタフェースAVIS(Advanced Visualizer of Instance and Schema)を提案している。このAVISを用いることでデータベースを視覚化し、そのスキーマやインスタンスの状況を直観的に把握することが可能になる。

しかし、データベースには非常に大量のインスタンスが入っており、この全てを画面に表示することは、限られた画面上では物理的に困難であると同時に、利用者が自分の知りたい情報を知る上での妨げにもなってしまう。またスキーマに関して、イン

スタンスと比較すればデータ数は多くはないが、必ずしもその全てを表示することは望ましくない。このとき、もし利用者が必要とする部分がデータベース全体ではなく、その一部分であるならば、利用者の要求する部分だけを抜き取り、表示することができれば有用であると考えられる。

本研究では、視覚的にデータベースの概念構造を表示し、それに対して操作できるインタフェースの設計とその実現手法について議論するとともに、本インタフェースを用いて、スキーマダイアグラム上に特に着目しているインスタンスを例として挙げた図を描くことにより、必要な部分のみを取り出して表示するという、データベースに対する“ふるい”という操作の手法を提案する。“ふるい”操作では、利用者には必要のない情報をデータベースから文字通りふるい落とすことで必要とするスキーマやインスタンスだけを画面に表示させる。これにより直観的な操作で複雑なスキーマや大量のインスタンスから利用者にとって必要な部分のみを取り出すことが可能になる。

## 2 視覚的インタフェース AVIS

マルチメディアデータベースは一見して複雑なスキーマを持ち、そこに大量のインスタンスが蓄積されている場合も数多い。そのため、データベース利用者がそのスキーマやインスタンスを見たい、知りたいときに、スキーマやインスタンスの状況を把握することは困難となる。そこでマルチメディアデータベース利用者が、スキーマやインスタンスの状況を直観的に把握できることを目的とし、マルチメディアデータベースを視覚化[1][2][3]し、その中から利用者が見たい・知りたいデータを自由にブラウジングできるインタフェースとして、我々はAVIS(Advanced Visualizer of Instance and Schema)[4][5]を提案している。このAVISの概念構成を図1に示す。

AVISを用いてデータベースのスキーマやインスタンスをブラウジングするためには、次のような機能が必要であると考えられる。

- データベースにおけるスキーマやインスタンスの直観的な視覚化
- データベースに蓄積されているデータから着

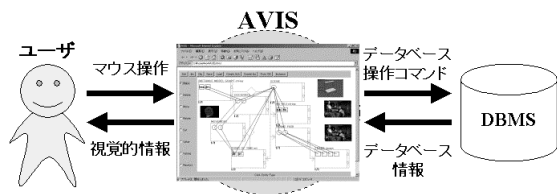


図 1: AVIS の概念構成

目している部分を抜き出して表示すること(ふるい操作)

- “ふるい” 操作によって抜き出した結果をブラウジングできること

まず本章において、視覚化およびブラウジング手法について述べ、次に次章において、本論文で新たに提案する“ふるい”操作について述べる。

## 2.1 データベースの直観的な視覚化とブラウジング

データベースを利用する上では、スキーマやインスタンスの状況を把握することが重要である。例えば、マルチメディアデータベースには、文字、映像、画像などのいろいろな値だけでなく、その解析情報が加えて蓄積される。そのため、そのスキーマはそれらデータ間のつながりが相互参照の入り組んだ一見複雑な構造となる。このようなマルチメディアデータベースのスキーマやインスタンスを視覚的にブラウジングすることでデータベースのスキーマ、およびインスタンスの複雑な関係を把握することが容易になると考えられる。

ところで、データベースの、特にスキーマの視覚化手法としては、既に意味データモデルである ER モデルに基づくスキーマのダイアグラム(図式的)表現が有用であることが知られている。我々はこの ER ダイアグラム [6] を拡張した表記法として、拡張 ER ダイアグラムを提案し、特にマルチメディアデータベースのデータベース化において有効であることを示してきた [7][8]。その視覚的インタフェースである AVIS ではこれを用いてスキーマの視覚化を行い、インスタンスもまたスキーマに重ねてダイアグラムで表現している。拡張 ER ダイアグラムで

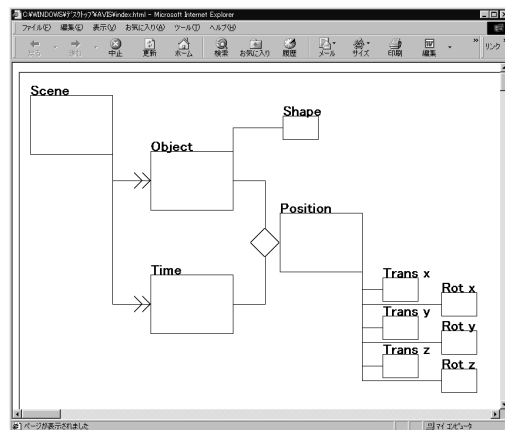


図 2: AVIS を用いたスキーマ表示例

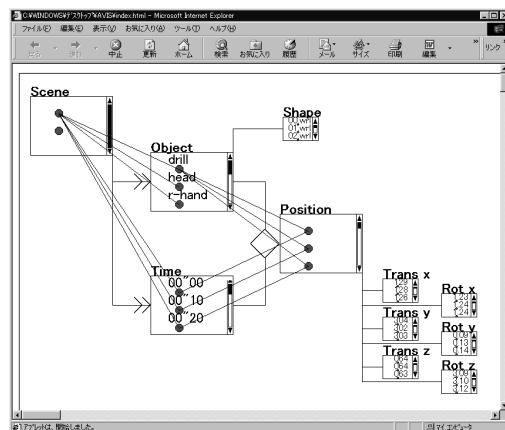


図 3: AVIS を用いたスキーマ・インスタンス表示例

は ER ダイアグラムと同様に entity-type(主体型) と relationship-type(関連型) という 2 つの概念によって対象世界をモデル化する。この拡張 ER ダイアグラムを用いたスキーマの視覚化例を図 2 に示す。

図 2 のスキーマ図では、ある時空間中の場 (Scene) に存在する物体 (Object) およびその場の時点 (Time)、またある特定の物体がある特定の時間に存在する位置 (Position) を表している。図 3 はさらにこのダイアグラムを用い、スキーマに重ねてインスタンスを表示した例である。図 3 では entity-type に属する entity が丸(値を持つ場合は値を併記)で表され、それらの間のつながり (relationship) が線分で表現されている。

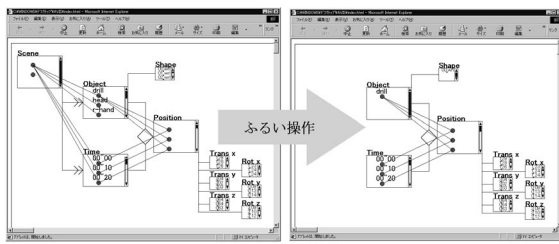


図 4: ふるい操作とその結果

我々が提案している視覚的インタフェース AVIS では、この拡張 ER ダイアグラムを用いてデータベースのスキーマ、およびインスタンスの視覚化を行っている。

このように、データベースの視覚化を行うことでデータベースのスキーマ、およびインスタンスの複雑な関係をブラウジングによって直観的に把握することが容易になると考えられる。ブラウジングを行う上では、entity-type に属する entity を entity が持つ値で sort したり、スクロールバーを用いて表示する entity を変更することができるが、これらブラウジング機能の詳細については本論文では扱わない(文献 [4] 参照)。

### 3 視覚的インタフェース AVIS を用いた“ふるい”操作

第 2 章では、我々の提案している視覚的インタフェース AVIS について述べた。このように視覚的インタフェースを用いてデータベースのスキーマやインスタンスを表示することで、直観的にそれらを見ることができ、データベースの中身をブラウジングする際に有用であると考えられる。

しかし、スキーマ全体、インスタンスの全てを画面上に視覚化することは物理的にも不可能であると同時に、必ずしも利用者の直観的なブラウジングを手助けすることにならない。特に、その大量のインスタンスの中から利用者が必要とするインスタンスを探し出すのは大変困難である。このことから利用者が必要とする部分のみを取り出して、ブラウジングしたいと考えることは自然な要求だと考えられる。

本章では、このように利用者が必要とする部分だ

けを取り出して視覚化する手法として、“ふるい”操作の提案を行う。

#### 3.1 データベースにおける“ふるい”操作

マルチメディアデータベースにおける、複雑なスキーマ、および大量のインスタンスの中から、利用者が必要とする部分のみを取り出してブラウジングするためには、その部分のみをまるでふるいにかけるように取り出してブラウジングする、という操作が考えられる。本論文では、このような操作を“ふるい”操作と呼ぶ。

“ふるい”操作は具体的には次のようにして行う。まず利用者が取り出したいデータを

- スキーマのどの部分に着目して“ふるい”操作を行うのかを記述する
- “ふるい”操作によって取り出したいインスタンスの条件を記述する

ことによって示し、次に、実際にこの“ふるい”操作をデータベースシステムが実行することにより、その結果として取り出されたスキーマおよびインスタンスを視覚化して表示する。これによって利用者がブラウジングを行う上で表示することが不必要と判断した部分を“ふるい”落として必要な部分だけを取り出して画面上に視覚化することができる。このふるい操作は、既に我々が提案しているマルチメディアデータベース検索言語である MMQL[9] における検索式の結合子で結ばれた要素操作のひとつに対応している。

#### 3.2 “ふるい”操作のための基本要素

本節では、ふるい操作を記述するために必要な基本要素について述べる。

AVIS ではふるい操作を行う際、どのようなデータを取り出したいのかを視覚化されたスキーマダイアグラム上に図として描く。これによって、

- スキーマダイアグラムと照らし合わせて直観的に取り出したいデータを記述することができる。
- 図として描くことによってどのように“ふるい”を行うのかを直観的に把握することができる。



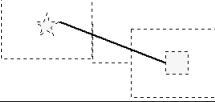


entity変数		entityを表す変数 (root)
		entityを表す変数 (root以外)
つながり		2つのentity変数間の つながり (経路)
条件式		entity変数が 満たすべき条件
Label		entity変数の値を 参照する為のラベル

図 5: ふるい操作の基本要素

といった利点が考えられる。図としてふるい操作を記述する際の基本要素は図 5 のとおりである。この基本要素は 3.1 節で述べた 2 つの操作にそれぞれ対応しており、entity 変数とつながりという要素を用いて取り出したい(着目している)部分を記述し、条件式と Label を用いて取り出したい条件を記述する。

以下でこれらの要素について詳しく述べる。

### 3.2.1 ふるい操作における取り出したい部分の記述

AVIS を用いたふるい操作では、取り出したい部分を記述するために、以下の 2 つの基本要素を用いる。

#### 1. entity 変数

ある特定の entity-type に属する任意の entity を表す変数のことをここでは entity 変数と呼ぶ。entity 変数は指定した entity-type 中を範囲として、それに属する entity 上を動きまわる変数である。entity 変数は図上では星型 (root を表す)、または四角形 (root 以外を表す) のいずれかで表される。(図 5)

#### 2. 2 つの entity 変数間のつながり (relationship)

entity を entity 変数で表現したとき、2 つの entity 変数間のつながりを図 5 のように 2 つの entity 変数間を線を用いてつなぐことで示

す。entity 変数間をつながりて示すことによって、それぞれの entity 変数が表す entity 同士が、実際につながりを持っていることを表す。

以上の要素を用いることで、ふるい操作において取り出したい部分を記述することが可能となると考えられる。

ところで、実際に取り出したい部分を記述する上では、通常、ある entity-type に主眼を置いてその entity-type から参照したい他の複数の entity-type へのつながりを考えることが多い。例えば図 2 のデータベースにおいて、物体 (Obj)、時点 (Time)、および位置 (Position) について考えたとき、『物体 (Object)』に主眼を置いた場合には、「あるシーン中に存在するある特定の物体だけに限ってその位置の時間的変化を追いたい」、また『時点』に主眼を置いた場合には、「ある特定の時点において、あるシーン中に存在している Object の位置を知りたい」というように明らかに観点が違ってくる。したがって、取り出したい部分を記述するには、主眼を置く場所を明確に記述する必要があると考えられる。

ここではこの主眼を置く entity-type の entity 変数を特に root 変数と呼び、図 5 のように、entity-type 中に星型を図示することで記述する。また、この root 以外の entity 変数は図 5 のように、entity-type 中に四角形を図示することで記述する。

この基本要素を用いた図 2 のデータベースにおけるふるい操作の例として、「名前が “drill” である物体 (Object) の形状 (Shape) と、その位置 (Position) の時間的変化を示す」というふるい操作を図 6 に、またその結果および結果のグラフィックス表示をそれぞれ図 7、図 8 に示す。

図 6 では、Position という entity-type 中に root 変数を、またこの entity-type 中の entity とのつながりを見たい entity-type 中にその他の entity 変数を記述することでこの操作における取り出したいデータの構造を表現している。

### 3.2.2 ふるい操作における条件の記述

AVIS を用いたふるい操作では、取り出したい条件を記述するために、以下の 2 つの基本要素を用いる。

#### 1. 条件式

## 名前が“drill”のObjectを選択

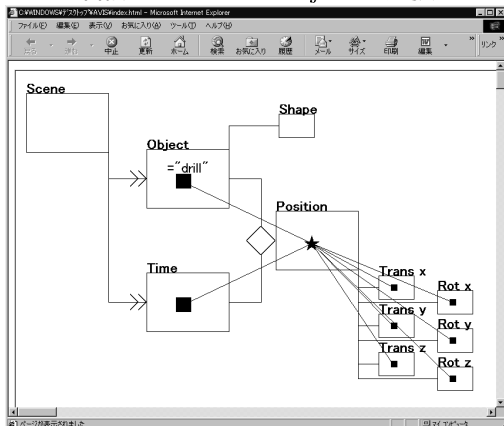


図 6: ふるい操作 1 の条件

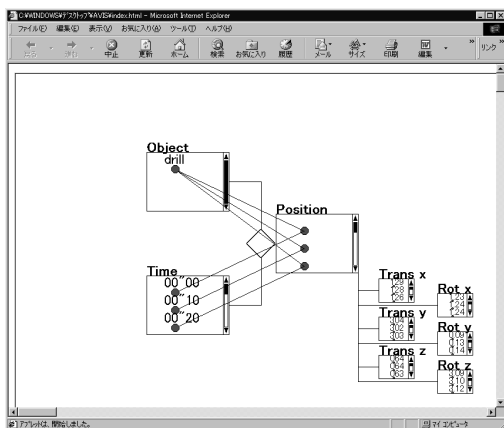


図 7: ふるい操作 1 の結果

大量のインスタンスの中から利用者にとって必要な部分がどの部分であるのかを条件および複数の条件の論理和 (OR)、論理積 (AND) を用いた条件式を記述することによって示す。このとき、各 entity-type 中の entity 変数は記述された条件を満たすかどうかチェックされ、条件を満たしているもの、およびその条件を満たしているものつながりを持っているものだけが取り出される。

## 2. Label

条件式を用いてデータベース中から取り出したい条件を記述する際、直接インタフェースに文字や数値のデータを入力することで、そ

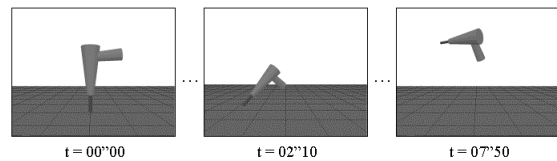


図 8: ふるい操作 1 の結果のグラフィックス表示

の定数と比較するだけではなく、他の entity-type 中の entity の値と比較することも考えられる。このような場合、他の entity 変数が表す entity の値を Label を用いて参照する。このように entity 変数に Label をつけることによってこの entity 変数が表す entity の値を Label によって参照することができる。

以上の要素を用いることで、ふるい操作において取り出したい条件を記述することが可能となると考えられる。

このように条件でふるいをかけたインスタンスに対して、表示するかどうかを記述する際には、表示したい entity 変数を選択し、その選択された entity 変数を表す図形の色を黒く塗りつぶすことによって図示する。これによってデータベースにより指定されたスキーマおよびインスタンスだけが抜き出されて表示される。但し、もともとのスキーマには存在していた entity-type を表示する必要が無い場合には、つながりの両端の entity 変数が存在する entity-type 間に仮想的な relationship-type を生成してスキーマを表示する。

条件式を用いた例として図 6 のふるい操作では、取り出したい物体 (Object) として「名前が“drill”である」という条件を記述している。このように条件付けを行いたい entity 変数の上に条件式を記述することで、その entity 変数に対する条件を記述する。

### 3.2.3 仮想的な entity-type

ふるい操作を行う際、データベース中の entity や entity の列に対して演算を行うことで新しい entity を作成して見たいという要求が考えられる。

このような場合、3.2.2 節で述べた Label を用いて演算を行うことによって、もともとのスキーマには

各 Object における位置の x 座標の平均を計算

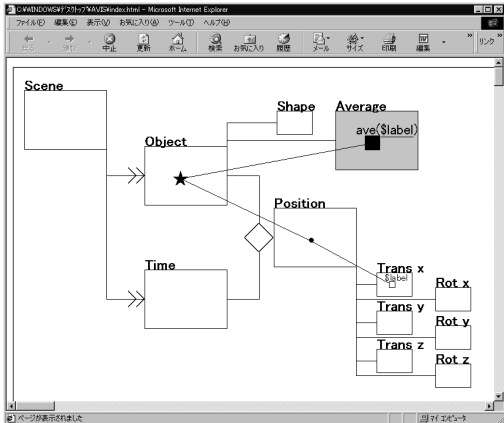


図 9: ふるい操作 2 の条件

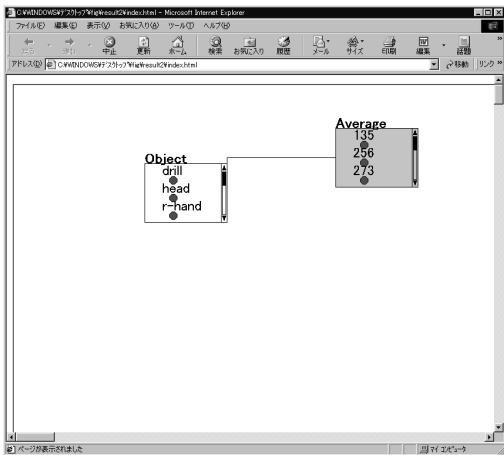


図 10: ふるい操作 2 の結果

存在しない entity-type および entity をまるでデータベース中に存在しているかのように仮想的に見ることができる。

このような仮想的な entity-type を作成するふるい操作の例として、図 2 のデータベースにおける「ある特定の物体 (Object) の位置 (Position) とその x 座標 (Trans x) の平均値を示す」というふるい操作を図 9 に、またこのふるい操作を行った結果を図 10 に示す。

図 9 ではもともとデータベースには存在しなかった Average という entity-type が新しく記述されている。このように仮想的な entity-type を記述することでその entity-type 中に新しく仮想的な entity

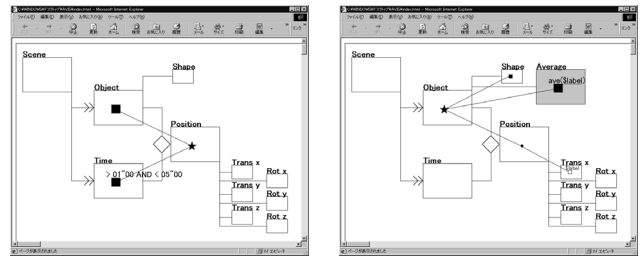


図 11: ふるいの組み合わせ (左:1段階目 右:2段階目)

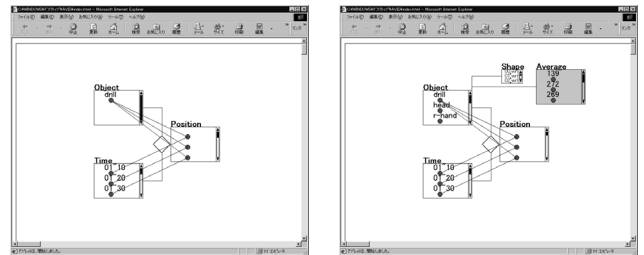


図 12: 図 11 の結果 (左:1段階目の中間結果 右:最終結果)

を作成することが可能となる。

### 3.3 ふるい操作の組み合わせ

これらのふるい操作では基本操作を組み合わせることで着目した entity とその entity からつながっている entity の塊として取り出すことができる。しかし、

- つながっている entity のさらにその先につながっている entity を取り出したいとき
- ふるいで取り出した値、または生成された仮想 entity の値を参照してさらにふるいを行いたいとき

には、これだけでは行うことができない。そこで複数のふるい操作を組み合わせることで、前述のような複雑な操作を行うことを可能にした。

このようなふるいの組み合わせ例として、図 2 のデータベースにおける、「時点 (Time) が 1 秒から 5 秒までの間に存在する物体 (Object) について、その物体の形状、およびその位置の平均値を示す」と

いうふるい操作を図 11(左:1 段階目 右:2 段階目) に示す。また、1 段階目のふるいを行ったあとの中間結果、さらに 2 段階目のふるい操作の結果を図 12 に示す。

この例では、「物体の位置 (Position) を root として時点と形状を取り出す。但し、時点の条件として値が 1 秒～5 秒の間のみ」という 1 段階目のふるいの結果に対して、さらに「1 段階目に取り出された物体 (Object) を新たに root として、形状と平均 x 座標を取り出す」という 2 段階目のふるいを組み合わせてかけることによって、目的とする結果を取り出して視覚化を行っている。

このように、複数の段階にわたってふるいを記述することで、ふるいで取り出された entity を使って、さらにその先につながっている entity を取り出すことや、取り出された entity の値を参照した比較や演算を行うことができる。このとき、1 段階目、2 段階目、とふるい操作を手順的に記述することで操作を行う順番を明示的に指定することができる。このようにふるいを複数段にわたって記述することにより、MMQL[9] を用いた条件検索に相当する操作を行うことができる。

## 4 まとめ

本論文では、視覚的データベースインタフェース AVIS を用いたふるい操作について述べた。本手法により、データベースブラウジングを、視覚的・直観的な操作で行うことが可能になる。

本手法では、視覚的に表示したスキーマ上に直接、ふるい操作を記述することができる。また、そのふるい操作を行う際に、繰り返して複数段のふるい操作を行うことで、より複雑なふるい操作を順序を認識しながら行うことができる。これにより、一般利用者でも簡単な操作でマルチメディアデータベースのような複雑なスキーマ、および大量なインスタンスの中から必要な部分のみを取り出すことが可能になると考えられる。

## 参考文献

[1] Stuart K. Card, "Visualizing retrieved information: A Survey", IEEE Computer Graph-

- ics and Applications, Vol.16, No.2, pp. 63-67, 1996
- [2] 古畑 理香, 藤代 一成, 市川 哲也, 竹島 由里子, "GADGET/IV: 情報可視化の半自動設計支援環境", 第 11 回データ工学ワークショップ (DEWS2000), 2000
- [3] 磯部 成二, 黒川 清, 塩原 寿子, 飯塚 哲也, "視覚化による多次元データ分析システム: INFORVISER", 情報処理学会論文誌, Vol.40, No.5, 1999
- [4] 家富 誠敏, 森崎 慎之, 富井 尚志, 有澤 博, "マルチメディアデータベースのための視覚的データベースインターフェース AVIS の実装", 電子情報通信学会論文誌, VOL.J84-D-I No.6, pp. 777-789, 2001.
- [5] 森崎 慎之, 家富 誠敏, 富井 尚志, 有澤 博, "マルチメディアデータベースのためのデータ定義ダイアグラム", 情報処理学会研究報告, Vol.99, No.61, 99-DBS-119, pp. 165-170, 1999.
- [6] P. P. Chen, "The Entity-Relationship Model: Toward a Unified View of Data", ACM Transactions on Database Systems, Vol.1, No.1, 1976.
- [7] H. Arisawa, H. Nagae, Y. Mochizuki, "Representation of Complex Objects in Semantic Data Model "AIS" and Implementation of Set Operators", IEICE TRANSACTIONS, vol.E74, No.1, 1991.
- [8] H. Arisawa, T. Tomii, H. Yui, and H. Ishikawa, "Data Model and Architecture of Multimedia Database for Engineering Applications", IEICE TRANS. INF. & SYST., vol E78-D, No.11, 1995.
- [9] 富井 尚志, 有澤 博, "マルチメディアデータベースにおける映像モデリングと操作言語", 電子情報通信学会論文誌, Vol.J79-D-II No.4, pp. 520-530, 1996.