

### 3次元空間データベースにおけるデータモデルと アクセス管理機構について

上浦 真樹<sup>†</sup> 依田 和也<sup>†</sup> 田島 敬史<sup>‡</sup> 田中 克己<sup>\*</sup>

<sup>†</sup>神戸大学大学院自然科学研究科情報知能工学専攻

<sup>‡</sup>神戸大学工学部情報知能工学科

<sup>\*</sup>神戸大学大学院自然科学研究科知能科学専攻

{kamiura,yoda,tajima,tanaka}@in4wolf.in.kobe-u.ac.jp

我々は、都市情報提供、都市景観シミュレーションなどの多目的利用が可能となるような、地域情報 Web システムを構築中である。その際、多数の組織が別々に保有する、建物や道路などに関する3次元グラフィクス情報を、VRMLを用いてデータベース化することによって、情報統合を実現する。本稿では、これらの3次元空間データのための part-of 関係や details-of 関係を考慮したデータモデルと、そのモデル上でのアクセス制御機構について述べる。

### A Data Model and Access Control Mechanism for 3-Dimensional Spatial Databases

Masaki Kamiura<sup>†</sup> Kazuya Yoda<sup>†</sup> Keishi Tajima<sup>‡</sup> Katsumi Tanaka<sup>\*</sup>

<sup>†</sup>Division. of Computer and Systems Engineering,  
Graduate School of Science and Technology, Kobe University

<sup>‡</sup>Division. of Computer and Systems Engineering,  
Faculty of Engineering, Kobe University

<sup>\*</sup>Division of Intelligence Science,  
Graduate School of Science and Technology, Kobe University

We have been constructing an urban information web system which integrates several applications, such as urban information service and urban view simulation. In order to integrate 3D-graphical data on structures, roads, and so on, which was originated in different information resources, we construct a database of those information by using VRML. In this paper, we explain a data model for those 3-D data which have part-of and details-of relationship, and an access control mechanism on that data model.

## 1 まえがき

我々はこれまでVRMLを用いて、都市に関わる大量・大規模な3次元情報を、オブジェクト指向データベースシステムで管理し、データの統合や共有をはかり、多目的利用できる環境を構築することを目指した地域情報 Web システムを提案してきた [1][2]。

データは建築設計、都市設計などで作成されたCADデータをVRML形式に変換したものを格納・統合する。これにより、WWW上で3次元グラフィックスオブジェクトの中からリンクによる情報の参照、HTML文書から3次元グラフィックス情報の参照などが可能になる。

このような、都市情報のVRMLデータベースを構築する際の課題として、

### ● 属性情報の欠如

VRMLではソースのどの部分が意味のあるオブジェクトを成しているか、またどういった性質、属性を持つかといった記述を行うことが困難である。

### ● リンク情報の欠如

CADデータを変換して得たVRMLデータにはリンクが無い。地域情報Webにおいては、オブジェクトに対し関連する情報をリンクによって付加したいという要求がある。しかし、それぞれのオブジェクトにアンカーをつける作業は非常に煩雑である。そこで、付加された属性情報や形状データから系統的に「条件を満たすすべての～にリンクを張る」といった要求に対応できる方が望ましい。

等があり、これらを解決してきた [1]。

本稿では、次の課題として3次元空間データベースにおけるアクセス制御について焦点をあて、3次元空間情報に特有の関係である part\_of 関係 (包含関係)、details\_of 関係 (詳細関係) に基づいたアクセス制御の機構を提案する。

3次元空間データベースにおけるアクセス制御の問題や特徴として以下の事柄が挙げられる。

### ● データは part\_of 関係を持つ

都市という公共の空間は様々な建築物や、水道、ガス、電力といった都市設備等のデータから構成され、これらもまたさらに細かいデータから成り立っている。

### ● データの詳細度に応じたアクセス制御

3次元構造物のデータは同じデータを表示す

るにも、その詳細度というパラメータが考えられる。よってデータはオブジェクト単位で複数の詳細度を持つと考える。この詳細度に応じたユーザのアクセス権を設定しアクセス制御を行う。

### ● 視点の位置に依存したアクセス制御

オブジェクトはどの視点から見ているかという文脈によって異なった表示がなされ、複数の表現を持つ。そこで我々は、オブジェクトと視点の距離に応じて異なる詳細度を表示するアクセス機構を提案する。さらに視点との距離だけではなく視点の位置 (方位) に依存したアクセス制御も必要になると考えられる。

以降、本稿では、2節で試作システムの概要、3節でモデル、4節でアクセス制御の機構に関して述べる。

## 2 試作システムの概要

### 2.1 VRML (Virtual Reality Modeling Language)

VRML [3] は WWW 上で3次元グラフィックスシーンを作成し、ハイパーテキストの形で利用するために開発された言語である。

VRMLで記述されたデータは、ノードとフィールドからなる。ノードは階層構造を持ち、フィールドは、ノードのためのパラメータである。1つのVRMLデータは、シーングラフと呼ばれるノードの階層構造に配置される。

### 2.2 VRML オブジェクトデータベースの機構

本研究では、1節で述べた諸課題を解決するために、VRML オブジェクトデータベースを次の2つに分け、これらを相互に接続する形で実現している。

#### 1. シーングラフデータベース

CADのデータ群を各種コンバータに通して得たVRMLを複数のシーングラフ構造のノードオブジェクト群として格納したデータベース。

#### 2. 概念オブジェクトデータベース

1つの構造物など意味のあるオブジェクト (一般にシーングラフ内の部分木に対応) を構造物の種別毎に用意された各クラスのインスタンスとして保有するデータベース。このオブジェクトには、対応するシーングラフの部分木の根ノードへのポインタを持たせると共に、物理的な part-of 構造を記憶する属性や、その

他の記述情報を格納する属性群を持たせる。

つまり、パーザーにより実際にはメモリ上に得られる階層構造を OODBMS に格納することで永続化し、その OODBMS の中で情報を記述しようというものである。シーングラフ DB と概念オブジェクト DB の関係のイメージを図 5 に示す。

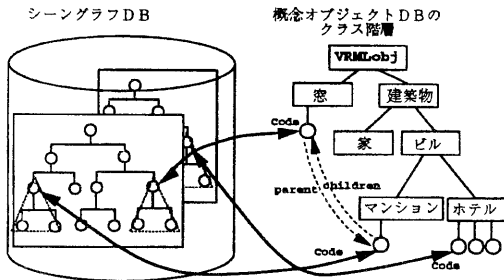


図 1: シーングラフ DB と概念オブジェクト DB

通常のファイルベースの VRML の場合、検索はファイル名単位で行うか、あるいは、フルテキストサーチを用いて必要な部分を検索するしか方法がない。しかしデータベースを用いることにより、先に述べたシーングラフのノード階層構造のデータ以外に、意味のあるノード群のまとまりや、その属性情報、オブジェクト間の物理的な part-of 構造や instance-of 関連の情報が別途格納されているため、以下のような検索が可能となる。

- 属性情報による検索
- オブジェクトの種別 (instance-of 関連) による検索
- オブジェクトの物理的な包含関係 (part-of 関連) による検索
- 位置座標にもとづく空間的な検索

検索によって得られた概念オブジェクトから VRML を生成するアルゴリズムは、次のようになる。図 5 において、クラス:マンションのインスタンスが検索結果であった場合、まず、対応するシーングラフ DB 内の部分木から VRML のコードを取りだし、次に part.of の children である窓クラスのインスタンスを参照し、その対応する VRML コードを取り出す。これらの得られた VRML コードをマージし、VRML データを作り出す (図 2 参照)。

また、結果の表示には結果オブジェクトの表示だけでなく以下のような表示方法を考えている。

- 結果オブジェクトのみを表示する。

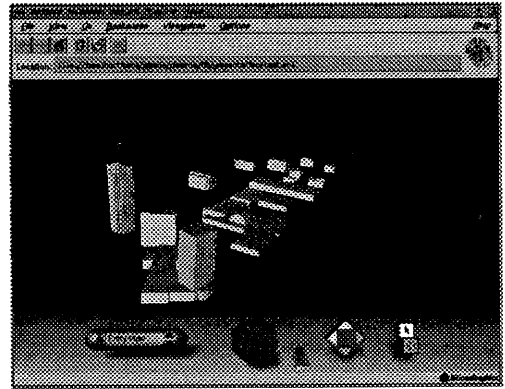


図 2: 部分抽出されたオブジェクト群

- 結果オブジェクトを含む一角を表示する。これは周囲の景観が必要な場合、結果オブジェクトからの距離を指定し、周囲も 3 次元化するものである。
- ウォークスルーを用いた表示。仮想空間での迷子現象を避けるために、結果オブジェクトまでの道筋をウォークスルーで提供し、半自動的にナビゲートするものである。

なお検索は Netscape から CGI と FORM の利用により行うことを想定し、バックエンドのエンジンとして OODBMS O<sub>2</sub> を用いている (図 3 参照)。

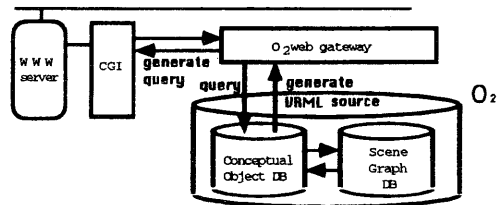


図 3: 実現機構

リンク機構には動的リンク [4][5] を用いる。動的リンクとは静的リンクがあらかじめアンカーやリンク先が決まっているのに対して、参照時にデータベースへの問い合わせによってアンカーやリンク先が決定するものである。リンク情報はアンカーとリンク先の双方の条件を記した質問対の集合で、データとは別に一括して格納される。

具体的には、まず 3 次元オブジェクトに属性値として付加されている文字情報を AsciiText ノードを用いて画面上に表示する (図 4 参照)。その際この表示される文字の内容がリンク情報においてアンカー候補であるならば、このノードに対して

アンカーを埋め込み、クリックした時点でリンク先探索プログラムが働き、リンク情報の条件をもとにデータベースへ問い合わせ、リンク先が決定され、クライアントに返される。

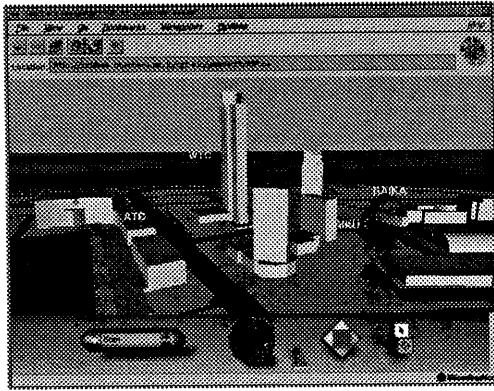


図 4: 動的リンク付与の例

### 3 データモデル

前節で、VRML で記述されている各構造物の意味を記述するために、概念オブジェクトを導入した。この節では、このような 3 次元空間の構造物を記述するためのデータモデルについて考察する。

#### 3.1 データ構造

既に述べたように、3 次元空間の構造物をあらゆるデータにおいては part-of 関係や detail-of 関係が重要な役割を持つ。そこで、これら二つの関係をもつデータモデルとして以下のようなデータ構造を考える。

実体 = { オブジェクト }  
 オブジェクト = (オブジェクト識別子,  
                   [dof = 参照,  
                   parts = { 参照 }, ...])  
 参照 = オブジェクト識別子

実体は実世界の各実体に対応し、その実体を異なる詳細度で表現している複数のオブジェクトの集合で表される。オブジェクトは、オブジェクト識別子と、属性の並びの対である。また全てのオブジェクトに対し、そのオブジェクトの detail-of 関係の親への参照を表す属性 *dof* と、そのオブジェクトが部分として持つオブジェクト、すなわち part-of 関係の子供への参照の集合である属性 *parts* が定

義される。各参照の中身はオブジェクト識別子である。すなわち part-of 関係は実体間の関係ではなく、個々のオブジェクト間の関係として与えるものとする。

#### 3.2 part-of 関係と detail-of 関係の構造

次に、上で定義されたデータ構造のインスタンスが満たすべき条件について考える。

##### detail-of 関係の構造

オブジェクト  $o_1$  が  $o_2$  と同じ実体をより詳細に表現したものであることを表す二項関係  $o_1 \geq o_2$  はオブジェクト間の *dof* 属性の推移的閉包として定義される。この時サイクルがあってはいけないものとする。さらに、各実体には詳細度が最も低く、情報量が 0 のオブジェクト  $o_0$  が定義されていると仮定すると、各実体に関する  $\geq$  関係は  $o_0$  を根とする木を構成する。

##### part-of 関係の構造

オブジェクト  $o_1$  が  $o_2$  の部分であることを表す二項関係  $o_1 \sqsubset o_2$  は *parts* 属性によって定義される。この時、 $\sqsubset$  は森を構成するものとする。すなわち part-of 関係はサイクルを持つてはならず、また排他的な参照である。

##### detail-of 関係と part-of 関係の間の一貫性

全オブジェクトの *dof* 属性と *parts* 属性の値が与えられた時、以下の性質が成り立てば、これらの属性値は一貫性があると考えことにする。

$$\forall o_1, o'_1, o_2 \text{ s.t. } o'_1 \geq o_1, o_2 \sqsubset o_1, \\ \exists o'_2. o'_2 \sqsubset o'_1, o'_2 \geq o_2$$

すなわち、 $o'_1$  が  $o_1$  の詳細化であり、 $o_1$  がその部分として  $o_2$  を持つ時、 $o'_1$  は必ず  $o_2$  の詳細化であるような  $o'_2$  を部分として持たなければならない。これは、detail-of 関係と part-of 関係の意味から考えて合理的な制約であると考えられる。

#### 3.3 データ記述の例

例として、東京タワーとその部分である展望台部分の記述例を以下に示す。

東京タワー = { $\langle t_1, [dof=nil, parts=\{\}, \dots] \rangle,$   
                    $\langle t_2, [dof=nil, parts=\{p_1\}, \dots] \rangle,$   
                    $\langle t_3, [dof=t_2, parts=\{p_3\}, \dots] \rangle$ }  
 展望台 = { $\langle p_1, [dof=nil, parts=\{\}, \dots] \rangle,$   
                    $\langle p_2, [dof=p_1, parts=\{\}, \dots] \rangle,$   
                    $\langle p_3, [dof=p_1, parts=\{\}, \dots] \rangle$ }

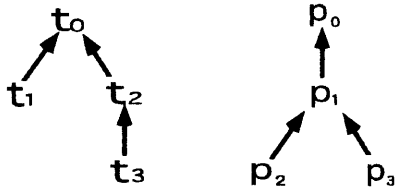


図 5: details\_of 関係の木構造

この例では東京タワーと展望台はそれぞれ  $t_1, t_2, t_3$  と  $p_1, p_2, p_3$  の三種類の詳細度のオブジェクトで表現されている。 $t_2$  は展望台部分として  $p_1$  を参照しているので、 $t_2$  を表示する際には  $p_1$  が用いられ、同様にして  $t_3$  を表示する際には  $p_3$  が用いられる。 $p_2$  は展望台部分のみを指定して表示する時にしか用いられないことになる。また、 $t_1$  は展望台オブジェクトを何も参照していないが、これはこの詳細度では展望台部分は省略されていることを意味する。

また、東京タワーと展望台のオブジェクト間の  $\geq$  は、自動的に与えられている最も詳細度の低いオブジェクト  $t_0, p_0$  を含めて、図 5 のような木を構成するように定義されている。また、この記述は上で述べた detail-of 関係と part-of 関係の間の一貫性も満たしている。

#### 4 アクセス制御

この節では、前節のデータモデルに基づくアクセス制御機構について説明する。特に実体が複数の詳細度を持つ点を生かし、ユーザがある構造物にアクセスした時に、単純にアクセスを許可または拒否するのではなく、その構造物に関してそのユーザが見ることを許される詳細度を決定し、その範囲内で表示する枠組を提案する。その際、その枠組が自然で合理的なものであるためには、次の二点が満たされるべきであると考えられる。

- 各ユーザに対して表示すべき詳細度が一意に決まること。
- あるユーザがあるオブジェクト  $o_1$  を表示不可能ならば、 $o_2 \geq o_1$  なる任意のオブジェクト  $o_2$  も表示不可能であること。

後者は、この枠組があるユーザに対し、ある詳細度の表示は許すが、それより低いある詳細度の表示は許さないような一貫性の無い結果を生じないことを意味している。

#### 4.1 アクセス権の指定

アクセス権の指定は、各ユーザ毎にそのユーザが表示してよいオブジェクトの集合を指定する事で行う。またこの時、あるユーザのアクセス可能な集合にオブジェクト  $o$  が含まれる時、そのユーザには任意の  $o \geq o'$  である  $o'$  のアクセス権が与えられるものとする。例えば、前節の東京タワーの例において、あるユーザに対しアクセス可能なオブジェクトの集合として  $\{t_3, p_1\}$  が与えられたとする。この時、このユーザがアクセス可能なオブジェクトは、 $\{t_0, t_2, t_3, p_0, p_1\}$  となる。

#### 4.2 表示詳細度の決定

ユーザがあるオブジェクトを表示しようとした場合、そのオブジェクト、およびそのオブジェクトが part-of 関係を通して参照している全てのオブジェクトについて、そのユーザがアクセス権を持っているならば表示を許す。例えば前述のユーザが  $t_2$  を表示しようとした場合、このユーザは  $t_2$  およびその部分である  $p_1$  ともにアクセス可能なので、 $t_2$  は表示可能である。一方、このユーザが  $t_3$  を表示しようとした場合、 $t_3$  自身はアクセス可能だが、その部分である  $p_3$  はアクセス可能ではない。この場合、二通りの対応が考えられる。

- アクセスを拒否し何も表示しない。
- $p_3$  はアクセス不可能だが、それと detail-of 関係にあり少し詳細度が落ちる  $p_1$  はアクセス可能なので、 $t_3$  と  $p_1$  を組み合わせる表示する。

すでに述べたように、ここでは、複数の詳細度を生かした柔軟なアクセス制御を実現するために後者の方針をとることにする。

しかし、あるオブジェクトを、本来の指定とは異なる詳細度を持つ部分オブジェクトと組み合わせる表示するのは、常に可能とは限らない。組合せによっては不自然な表示になってしまう場合もあると考えられる。そこで、前述のデータモデルを以下のように変更し、parts 属性で部分オブジェクトを参照する際に、許される詳細度の範囲を指定できることにする。

オブジェクト = (オブジェクト識別子,  
 [dof = 参照,  
 parts = { 範囲指定参照 }, ...])  
 範囲指定参照 = (オブジェクト識別子,  
 オブジェクト識別子)

範囲指定参照  $\langle o_1, o_2 \rangle$  は、標準として参照するオブジェクト  $o_1$  と、最低限必要とされる詳細度を持つ  $o_2$  の対である。この時、 $o_1 \geq o_2$  でなければならない。 $o_1$  と  $o_2$  に同じ値を指定すると特定の詳細度のみを参照している事になる。

これに合わせて、detail-of 関係と part-of 関係の間の一貫性も次のように修正する。ただし、定義中  $\langle o_2, o_3 \rangle \sqsubset o_1$  は  $\langle o_2, o_3 \rangle$  という範囲指定参照が、 $o_1$  の part-of 関係の子供として現われることを表す。

$$\forall o_1, o'_1, o_2, o_3 \text{ s.t. } o'_1 \geq o_1, \langle o_2, o_3 \rangle \sqsubset o_1. \\ \exists o'_2, o'_3. \langle o'_2, o'_3 \rangle \sqsubset o'_1, o'_3 \geq o_3$$

すなわち、範囲指定参照の下限を指定する側のオブジェクトについて従来の一貫性と同一ことが成り立つことを意味する。

このような範囲指定参照を使って記述されているオブジェクトを表示する際の表示詳細度の決定手順を例を使って説明する。今、東京タワー、その部分である展望台、さらにその部分である展望台につけられた照明器が以下のような形で記述されていたとする。

東京タワー =  $\{ \langle t_1, [dof=nil, parts=\{ \langle p_1, p_1 \rangle \}, \dots] \rangle, \\ \langle t_2, [dof=t_1, parts=\{ \langle p_2, p_1 \rangle \}, \dots] \rangle \}$   
 展望台 =  $\{ \langle p_1, [dof=nil, parts=\{ \langle l_1, l_1 \rangle \}, \dots] \rangle, \\ \langle p_2, [dof=p_1, parts=\{ \langle l_3, l_2 \rangle \}, \dots] \rangle \}$   
 照明器 =  $\{ \langle l_1, [dof=nil, parts=\{ \}, \dots] \rangle, \\ \langle l_2, [dof=l_1, parts=\{ \}, \dots] \rangle, \\ \langle l_3, [dof=l_2, parts=\{ \}, \dots] \rangle \}$

そして、あるユーザのアクセス可能な集合が  $\{ t_2, p_2, l_1 \}$  として与えられたとする。この時、このユーザに対する表示は以下の手順で決められる。

1.  $t_2$  はアクセス可能。その部分である  $p_2$  を調べる。
2.  $p_2$  はアクセス可能。その部分である  $l_3$  を調べる。
3.  $l_3$  はアクセス不可能。しかし  $p_2$  からの参照は  $l_2$  まで許容範囲なので、 $l_3$  から  $dof$  属性をたどって  $l_2$  に達するまで調べる。
4.  $l_3$  の  $dof$  をたどると  $l_2$  だが、アクセス不可能。既に  $l_2$  に達しているため、 $p_2$  は表示不可能。

5.  $p_2$  は表示不可能だが、 $t_2$  からの参照は下限が  $p_1$  なので、 $p_1$  を調べる。
6.  $p_1$  はアクセス可能。その部分である  $l_1$  を調べる。
7.  $l_1$  はアクセス可能。
8.  $t_2, p_1, l_1$  を使って表示。

以上の手順で、このユーザに対しては  $t_2, p_1, l_1$  を使って表示を行なうことが決定される。

このような手順で表示方法を決定する場合、この節の最初に述べた二点が満たされる。詳しい証明は省略するが、これは、アクセス権が detail-of 関係にそって子から親へ継承されることと、前述の part-of 関係と detail-of 関係の間の一貫性に関する制約を設けていることによる。

## 5 あとがき

本稿では、まず VRML オブジェクトデータベースの概要に関して述べた。次に part\_of、details\_of 関係に基づいたアクセス管理を行う機構を持つ、インスタンススペースのデータモデルの提案を行った。そして現在、VRML を用いてこのモデルに基づいたデータベースの構築を行っている。

## 謝辞

なお、本研究は、一部、文部省科学研究費重点領域研究（課題番号 08244103）による。

## 参考文献

- [1] 山本 淳、田中 克己「大規模 VRML オブジェクトデータベースシステムの設計」、情報処理学会 ADBS'95 論文集, pp121-129, 1995 年 12 月
- [2] 笹田剛史、川崎寧史、平田倫巳、竹中由秀:「都市のイメージシミュレーション (その 1) - ネットワークによる都市データの相互的利用について-」、日本建築学会第 18 回情報システム利用技術シンポジウム論文集, 1995 年 12 月
- [3] <http://vrml.wired.com/>
- [4] K.Tanaka et.al, Query Pair As Hypertext Links, In Proc.of the 7th IEEE Data Engineering Conference, pp456-463, Kobe, Japan, April 1991
- [5] 上浦真樹、森下淳也、上島紳一、大月一弘、田中克己「ハイパーメディアデータベースの段階的構造化と多重ビュー」、情報処理学会研究報告, 95-DBS-104, pp225-232, 1995 年 7 月