

木簡研究支援システムにおける視点操作と分類機構

森下淳也, 大月一弘 (神戸大学国際文化学部),

杉山武司 (姫路獨協大学情報科学センター), 上島紳一 (関西大学総合情報学部)

階層構造グラフを用いた木簡研究支援システムにおける視点操作と分類機構について考察した。木簡などのデータベースはデータが収集された時点で、データに対して大まかな構造しか与えられていない。言わば半構造化の状態のまま、構築されている。このような半構造化データに対して、複数の観点から、自在に属性付けを行うため、属性集合をグラフのノードと枝にリンクさせて、視点を表すノードからのサブグラフについて、グラフの上位から下位へと属性集合を伝播させるビューを定義した。このグラフを研究者が使用する際には、自分の視点に基づいた透過的なデータ操作が行なえることが重要である。これを支援するスコープの概念とカテゴリによって実現される分類機構を論じた。

Viewpoint Operations and Classification of Hierarchical Structure Graph for Wooden Slips Researches

J. Morishita, K. Ohtsuki (Faculty of Cross-Cultural Studies, Kobe University),

T. Sugiyama (Information Science Center, Himeji Dokkyo University),

S. Ueshima (Faculty of Informatics, Kansai University)

We discuss viewpoint operations and classification functionalities of Hierarchical Structure Graph for wooden slips researches. Databases such as wooden slips have loosely constraint structure, therefore, they construct as so called semistructured data. To give attributes depend on multiple viewpoints, we give a data model which represents as a directed graph. Each node and edge has a link to object that contains a set of attributes. Handling its subgraph from the node which represents viewpoint to another node which has elementary data and propagating all the attributes on every node and edge to the leaf node, the subgraph can give a specific view based on the viewpoint. It is important that users can manipulate their data in a transparent manner even with hierarchical structure graph. Therefore we introduce a scope for handling viewpoints and discuss our classification mechanism with a category.

1 はじめに

近年、文献情報データや環境情報データなどの科学技術データベースが注目されてきている [1, 2]。これらはデータが収集された時点で、データに対して大雑把な構造しか与えられていない場合が多い。言わば半構造化の状態のまま、構築されているという特徴がある。また、確定したスキーマを持つ場合でも、そのデータを利用する段階で、視点の違いや仮説などを盛り込んだ思考実験のような自由度を与えたいと考えられる。

木簡研究支援システムは、中国敦煌遺跡から出土した木簡一千本、中国居延遺跡の木簡一万本を対象にしたシステムである [3]。これらの木簡 (図 1) は、発掘された時点で基本的なデータを収集し、その後、木簡の釋読が進むにつれて文字データが収集される。さらに、釋読文から個々の意味が汲み取られるという過程を経て、データベースに格納される。このようなデータは、

- 文字も分からない状態のまま、長い期間保存される。
- データの解釈が後から発見される。
- データの意味が一般に確定するには、長い期間の研究が必要である。

など、特異な振舞いが見られる。



図 1. 木簡

我々はこのような大雑把なスキーマしか持たない半構造化状態のデータに対して、元のデータベースのデータを保持しつつ、利用者の視点に応じて、属性付けを行うことで様々な構造化を同時に表現するデータモデル、階層構造グラフを考案した [4]。

このグラフでは半構造化状態のデータを基本オブジェクトとして格納し、その仮想化のため、階層構造グラフ上で始点となるノードと端点となるノードを指定し、上下関係に沿ったノード間の属性を継承することで仮想オブジェクトを定義する。利用者は基本オブジェクト

ならびにカテゴリも含めて、仮想オブジェクトとして扱う。階層構造グラフに基づく仮想オブジェクトを扱うことで、対象となるオブジェクトへの多様な属性の付与、複数の分類構造の実現(多重分類)、オブジェクトの多様性の表現(多重属性)を実行時に表現できる。

本稿では、階層構造グラフを用いて利用者が透過的にデータ操作を行なえるように、スコープの概念を導入する。スコープは階層構造上の部分グラフである。利用者がスコープに基づいて仮想オブジェクトを操作することで、他の利用者の操作結果の参照や複数の利用者間の操作について独立性を保ちながら段階的構造化を行うことができる。

また、我々のカテゴリは視点とともに分類関係をも表現している。分類関係の表現に階層構造を用いた研究としては、山崎ら [5]、Shoen [6] による研究などがある。山崎らは、予め性質の与えられたデータの管理ではなく、解析過程のデータを分類し、階層化する目的で、データの階層的な集合の扱いを関係データベース上で扱う方法について考察している。Shoen らの扱う分類構造は予め確定されており、利用者が新たな分類を行うことはない。

2節では、階層構造グラフを説明し、スコープを3節で、階層構造グラフの段階的形成を4節で述べる。

2 階層構造グラフ

階層構造グラフは、ノードと枝からなるサイクルのない有向グラフである [4]。各ノードと枝には属性集合を持つオブジェクトへのリンクが張られている。リンクされるオブジェクトはオブジェクト・アイデンティティと属性集合からなる。次のような形式のものである。

$$O = \langle \text{oid}, \{a_1 : v_1, \dots, a_n : v_n\} \rangle.$$

ここで、oid はオブジェクト・アイデンティティであり、 a_i は属性、 v_i は値である。この属性集合の a_i と v_i はそれぞれ次の様に (attribute) と (values) で定義されるデータである。

```
(attribute) ::= symbol,
(values) ::= <value>
            | <value>, <values>,
(value) ::= int|string|...|oid.
```

オブジェクトは3種類あり、それぞれ基本オブジェクト、カテゴリ、関係オブジェクトと呼ばれる。基本オブジェクトは元々の半構造化状態のデータを格納するもので、グラフの最下位のリーフに置かれる。カテゴリはノードに置かれ、視点を表す。関係オブジェクトは枝に置かれ、視点とデータの間に固有の情報が書き込まれる。図2に階層構造グラフを示す。ノードと枝は、リンクされたオブジェクトの属性集合 $\alpha_i, \beta_j, \rho_{kl}$ を各々持つ(四角で表される)。

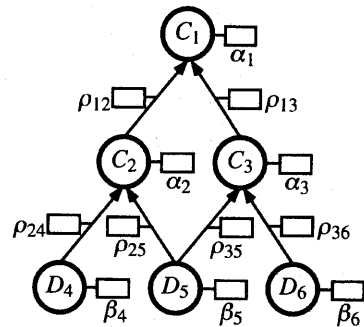


図2. 階層構造グラフ

このグラフに基づいた木簡研究支援システムを図3に掲げる。

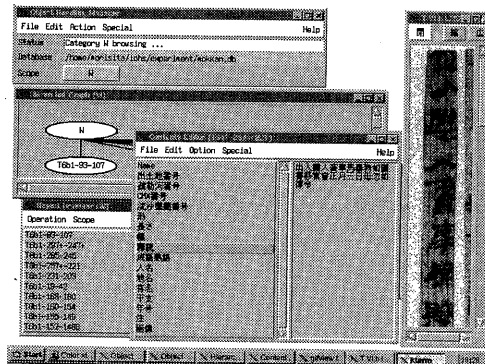


図3. 木簡研究支援システム

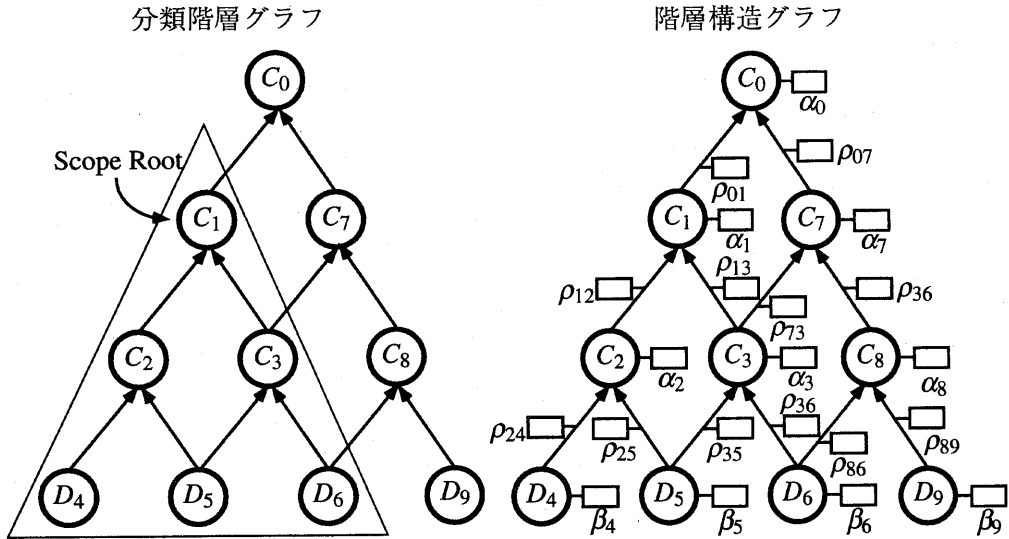


図5. 分類階層グラフ

仮想オブジェクト

オブジェクトの仮想化は属性集合の演算であるが、利用者はこの仮想化されたものを自分のオブジェクトとして扱いたい。そこで、分類項目のノードから下位に分類されているノードまでの部分有向グラフを仮想オブジェクトと定義する。図2のグラフにおいて、 C_2 から見た D_4 の仮想オブジェクトを $\tilde{D}_4[C_2]$ と表記する。これに対してノード自身のオブジェクトを実オブジェクトと呼んで区別する。

仮想オブジェクトの属性は、選んだ実オブジェクトの属性に、分類項目のノードからそのオブジェクトに至る総てのパスからの属性を加えて、膨らんだものである。仮想オブジェクトをみることで利用者のオブジェクトに付与した属性があたかも実オブジェクトに付与されたかのように見える。

例えば、図2での C_1 から見た D_5 の仮想オブジェクト、 $\tilde{D}_5[C_1]$ ではノード、 C_2 と C_3 が間にあるため、その属性は $\alpha_2 + \rho_{25} + \alpha_3 + \rho_{35} + \beta_5$ となる。

3 スコープ

オブジェクト階層上で一つのカテゴリから下位方向へデータオブジェクトまで至るパスの存在するオブジェクト階層の部分階層をスコープと定義する。スコープを決める最上位にあるカテゴリをスコープ・ルートと呼ぶ。スコープを指定するためにスコープ・ルートでそれを明記する事もできる。スコープ・ルート C_2 のス

コープをスコープ C_2 と呼ぶ。

スコープではスコープ・ルートとはつながらない経路は隠蔽され、スコープ自身が閉じたオブジェクト階層とみなされる。オブジェクト階層、図2とスコープの関係を図4に示す。三角で囲まれたものがスコープである。階層の有向枝である矢印がとぎれているのはスコープ内ではそれが見えないことを表している。

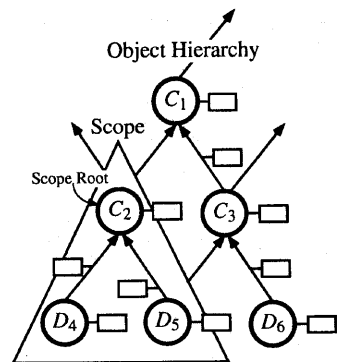


図4. スコープ

スコープは利用者の視点を反映するものであり、利用者は原則としてこのスコープを決めて、オブジェクト階層の操作を行うものとする。利用者が現在指定しているカテゴリをカレント・ルート、対象となる作業範囲をカレント・スコープ、カレント・スコープに含まれる全オブジェクトの集合を対象オブジェクトという。

スコープはオブジェクトの集合を扱う一つの外延としても機能する。例えば、スコープ C_1 の中から属性、 $a_1 = v_1$ であるような基本オブジェクトを検索する場合、

select * from C_1 where $a_1 = v_1$,

という問い合わせをすることで基本オブジェクトの集合が得られるが、この時の C_1 は集合の外延を意味する。

スコープによる利用環境

図5の階層構造グラフにおいて、任意の基本オブジェクトに対して複数個の仮想オブジェクトが定義されるが、これらはグラフ上には、陽には表れない。また、同グラフにおいて、どこまでが分類構造を表し、どこまでが仮想化を表すという区別はなく、利用者の意図や見方によって、分類構造にも仮想化ともみなすことができる。このため、利用者が階層構造グラフを直接参照しながら作業を行うことは困難であると考えられる。そこで、スコープの概念を用いて利用者の利用環境を規定し、操作の簡便性を与える。

仮想オブジェクトを通してみた分類階層は実オブジェクトの分類階層と同一である。スコープの中で仮想オブジェクトを通して新しいカテゴリに分類する場合でも、カテゴリの下位に位置づけられるものは実オブジェクトである。実オブジェクトが分類され、属性継承の働きによって、それがまた、仮想オブジェクトとなる。仮想オブジェクト自身が分類されているのではないことに注意が必要である。

Virtual Objects on Scope C_1

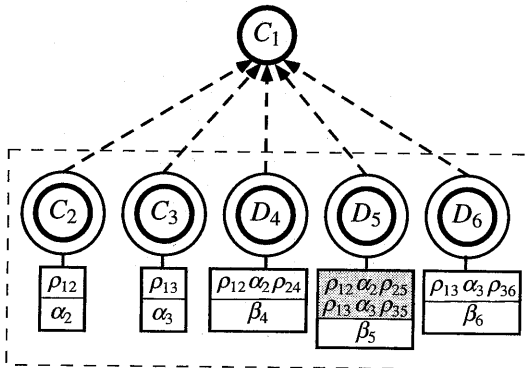


図6. C_1 による仮想オブジェクト

利用者には、各自の欲する仮想オブジェクトと、オブジェクトの分類構造を分離して見せることにする。利用者には、まず、図5に示すような分類階層グラフが与えられる。これは、実オブジェクト間の分類関係を表すもの、即ち、階層構造グラフから属性を取り除

いたものである。この分類階層グラフを用いてカレント・スコープを指定すると、そのスコープの対象オブジェクトがカレント・ルートの仮想オブジェクトとして表示されるものとする(図6)。つまり、カレント・スコープを指定すれば、そのスコープ内に分類されている実オブジェクトの数、ならびに、仮想オブジェクトの種類と数がすべて一意に決定する。

図6において、カテゴリも基本オブジェクトと同様に仮想化されており、区別なく表示される。また、実オブジェクトがカレント・カテゴリに対して直接的/間接的に分類されているかどうかは陽には表れないが、間接的に分類されているものに関しては、上位分類に値する仮想オブジェクトの属性はすべて継承される。

カレント・スコープが変更されれば、同一の実オブジェクトに対しても、その仮想オブジェクトは変更される。図7に、カレント・スコープを C_2 に移動させた場合の仮想オブジェクトを示す。

Virtual Objects on Scope C_2

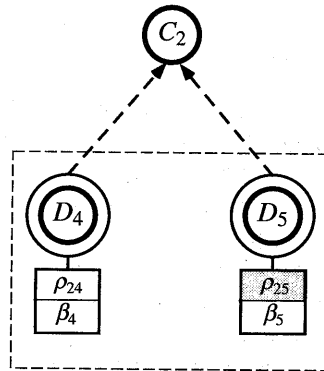


図7. C_2 による仮想オブジェクト

スコープの概念を用いることにより、利用者は、仮想オブジェクトを強く意識しなくてもよいことになる。つまり、分類階層と実オブジェクトのみを意識しておけば、自動的に必要な仮想オブジェクトを利用することができる。

4 階層構造グラフの段階的形成

利用者は、前節で示したスコープ操作をもとに段階的な構造化作業を行うことになる。これは、利用者に対して、

- 実オブジェクトと仮想オブジェクトの区別を強く意識しないでもよい。
- 仮想オブジェクトを操作する場合には、階層を意識しなくてよい。

という便宜を図るためである。

つまり、前者は、仮想化されているという意識をシステム側がサポートしようというものである。後者は、カテゴリ(分類集合)、基本オブジェクトの区別なく、図6に示されるような仮想オブジェクトの中からオブジェクトを選び出し、それを自動的に分類できるということの意味する。

しかしながら、スコープの概念を用いた仮想オブジェクトの操作環境は、操作を簡易性を計ったために、階層構造グラフを忠実に表現しているとはいえない。このため、利用者が行う操作を間接的に制限したり、追加操作を促すような付加機能が必要となる。

4.1 基本ルール

スコープに基づく仮想オブジェクトを操作するために、重要なことは階層構造グラフの正しい場所に情報が格納されるようにしなければならない。そのため、常にスコープの直下に情報を格納することを約束する。属性を付与する場合は、スコープとオブジェクトの関係オブジェクトに格納される。カテゴリを生成する場合には、スコープのサブオブジェクトとして生成する(但し、ルート・カテゴリを除いて)。こう約束することで、情報の所在に関する不定性をなくすることができる。

4.2 仮想オブジェクトへの属性付与

利用者は、仮想オブジェクトを通して属性を付与しようとする。実際の階層構造グラフの形状により、付与しようとした属性の位置が確定できなかったり、また付与できない場合があるため、「スコープ・ルートの直下にあるオブジェクトのみに対して、属性がスコープ・ルートとオブジェクトの関係オブジェクトに付与される」と約束する。即ち、実オブジェクトが基本オブジェクトの場合は、直上の枝に付与する。カテゴリの場合は、直上の枝に付与できるが、配下にも継承されることを利用者に警告する必要がある。実オブジェクトがスコープ・ルートに直接分類されていない場合は付与することはできない。

4.3 カテゴリの新規生成とオブジェクトの登録

スコープはカテゴリであるため、より上位のスコープから見れば集合として扱われる。スコープ内で行われる上記の操作は、より上位のスコープから見た分類階層と仮想オブジェクトのみを変更するものとする。

また、階層構造グラフでは属性の多重継承が許されるため、スコープの下に有るカテゴリが、他のスコープに含まれる場合がある。スコープ内で行われる上記の操作が、他のスコープから見た仮想オブジェクトと分類階層に影響を与えないためには、そのスコープにおける上位/下位の参照関係をくずすことはできない。

図8に仮想化されていないデータオブジェクトに対して、テンポラリカテゴリ C_l を用いてルート・カテゴリ C_n を新規生成する場合の過程を示す。左図は階層構造グラフの変遷、中図と右図はそれぞれスコープ・ルート C_l 、 C_n から見た仮想オブジェクトの変遷を示す。利用者は、中図、右図の仮想オブジェクトを見る。

図8の1で、カテゴリ C_n を生成する。ここではルートカテゴリである為、データオブジェクトと同様に扱われる。2で、登録すべきオブジェクトをグループ化する。ここでのグループ化は必ずしも、新規カテゴリ C_n を直接、使うわけではなく、一時的なカテゴリ C_l を用いている。このグループ化は、検索などの助けを借りる他、利用者が独自の判断で、オブジェクトを取り入れる場合も有る。この時点でも、属性付与による仮想化が行われる。

図8の3で、オブジェクトの登録が C_n になされる。 C_l に含まれるオブジェクトは一時的なものである。この登録がなされたオブジェクト D_3, D_4 がカテゴリ C_n のサブオブジェクトとなる。

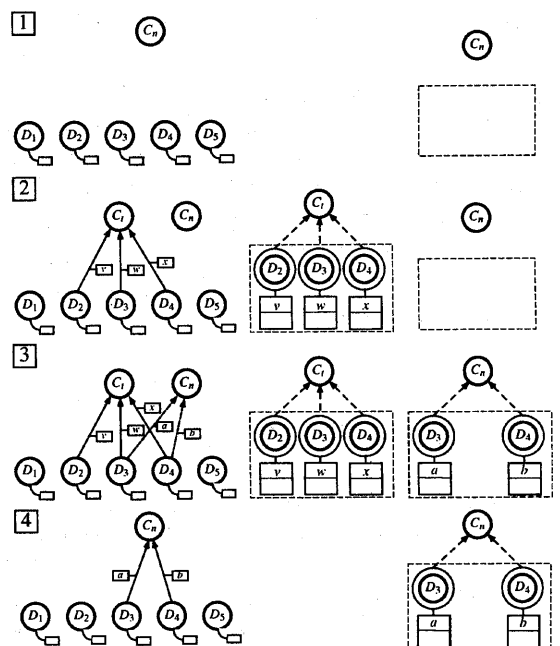


図8. テンポラリカテゴリを用いたルートカテゴリの生成過程

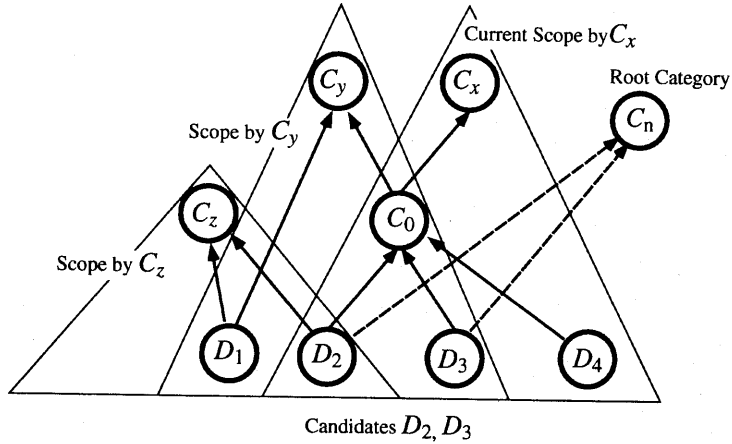


図9. 複数のスコープを用いたルート・カテゴリの生成

また、図9は、カレント・スコープを C_x とする時にルート・カテゴリ C_n を生成している。この図では、利用者がカレント・スコープ内の D_2, D_3 をスコープ C_y から仮想化し、更に D_2 をスコープ C_z から仮想化した結果、候補オブジェクトとして D_2, D_3 を選択したことを意味する。ルート・カテゴリ C_n により新しいスコープが生成されている。

階層構造グラフでは属性の多重継承が許されるため、1つのスコープの部分スコープがカレント・スコープの他のスコープに含まれる場合がある。カレント・スコープ内で親オブジェクト C_p をカレント・ルート以外のカテゴリを選択 ($C_x \neq C_p$) すれば、他のスコープで見た仮想オブジェクトと分類が変更される場合がある。従って、スコープのサブオブジェクト以外にカテゴリを生成することは禁止されている。これによって、上位以外の他のスコープに影響することなく、カテゴリを生成できる。これによりスコープの相互独立性が保たれる。つまり、 C_n が親カテゴリ C_p を持つ場合は、上記で $C_p = C_x$ とすればよい。

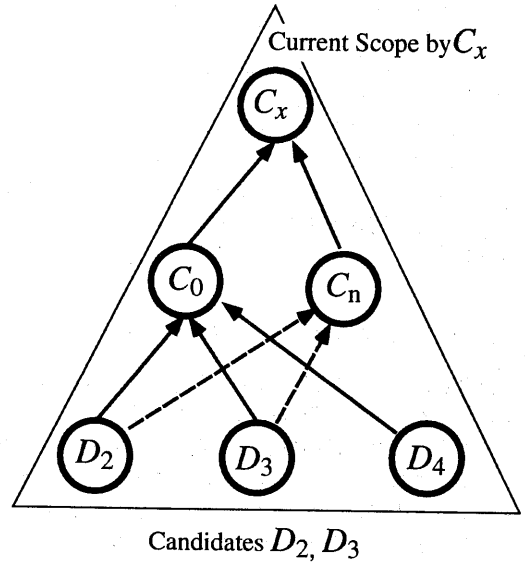


図10. 精細化

4.4 階層構造グラフの詳細化

候補オブジェクトをカレント・スコープ内のオブジェクトのみを対象オブジェクトとする場合を特に精細化という。精細化は、スコープ内のオブジェクトをまず大きな意図で分類して (= 広い視点からのオブジェクトの仮想化)、次に、更に細かく分類する (= 狭い視点からのオブジェクトの仮想化) 場合に相当する (図10参照)。

図11に木簡のデータによる精細化の例を示す。木簡の訳読文より、干支の情報を含むものをカテゴリ「干支」にまず、分類したものである。この「干支」をスコープとしてその中を「日付」と「刻限」に分けることで「干支」の表す意味への精細化を行おうとするものである。ちなみに双方に含まれるデータには二つの干支が含まれている。

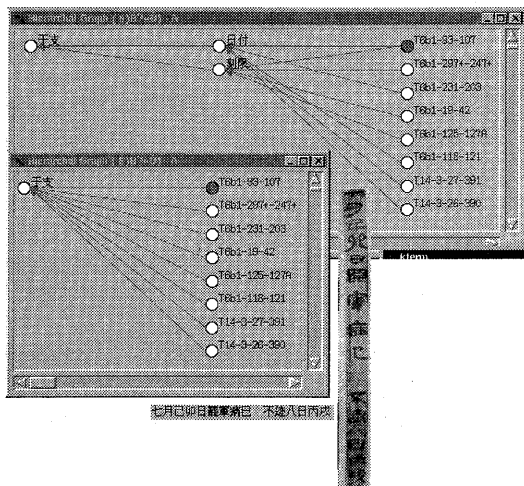


図 11. 木簡研究支援システムにおける精細化

4.5 属性付与による仮想化

カテゴリ C_n に対して登録したオブジェクトを C_n により仮想化して属性を付与する。付与された属性を階層構造グラフ上に格納する場所は複数が考えられる。 C_n がルート・オブジェクトの場合は、 C_n 、 C_n が親オブジェクトを持つ場合は、更に、親オブジェクトの属性、 C_n と親オブジェクトの関係オブジェクトの属性である。この中から位置を決定するには、付与される属性の意味を考慮する必要がある。基本ルール通り、カテゴリ C_n をスコープとして、関係オブジェクトに付与する他、カテゴリ C_n 自身に属性が付与される。

一般に階層構造の中で、付け加えられた情報は、その構造が進化するにつれて、その階層の成り立ちに応じて、変化していく。より下位に位置付けられた属性がより一般的なものとしてより上位の属性として扱われるように属性の移動が起こる。従って、カテゴリ生成の結果、影響を受ける上下の階層を更に吟味することで仮想化が進行する。ここでは、関係オブジェクトの共通の属性を意味に応じて、カテゴリ C_n や更にその上位の関係オブジェクトの属性に移動する操作などがある。

また、検索を用いて C_n に登録するオブジェクトを選択した場合は、検索に用いた属性を C_n の属性としたり、検索自身を C_n の属性とすることも考えられる。

4.6 階層構造グラフへの影響

前節の段階的形成の結果、階層構造グラフが受ける影響は以下の通りである。

C_n をルート・カテゴリとして作った場合は、他のス

コープに基づく分類構造も仮想オブジェクトも全く影響を受けない。

C_n を親オブジェクト C_p のサブオブジェクトとして生成した場合は、 C_p の上位カテゴリで定義されるスコープ内の仮想オブジェクトの数が増加する。また、各仮想オブジェクトは膨らむ。 C_p の上位カテゴリ以外のカテゴリから見た仮想オブジェクトは全く影響を受けない。即ちスコープの独立性が保たれる。

特に精細化の場合、 C_p の上位カテゴリから分類として見れば、分類階層は影響を受けない。カテゴリが 1 つだけ増加する。また、 C_p の上位カテゴリをスコープ・ルートとすれば、スコープ内の仮想オブジェクトは膨らむ。 C_p の上位カテゴリ以外のカテゴリから見れば、仮想オブジェクトは全く影響を受けない。

また、いずれの場合も、新規にカテゴリを生成しても仮想オブジェクトの数が増加するだけで、減少しない。 C_n が親オブジェクト C_p を持つことで、既存の階層構造グラフの属性を継承できる。また、属性付与の際に、既存の属性をそのまま利用できる。

このようにスコープを用いることで、利用者は作業範囲のオブジェクトを指定しながら、同時に他の利用者の作業を参照・利用したり、また、他の利用者が仮想オブジェクトに対して付与した属性を残すことができる。つまり段階的構築において既存の階層構造グラフの意味を保全することができる。

5 分類と仮想化の関係

本来、基本オブジェクトの分類とその仮想化は、別次元のものであるようにも思えるが、我々は、これらは同次元のものであると考える。統一的な観点からオブジェクトを仮想化すれば、それは、その観点のもとに分類されていることを意味する。分類された基本オブジェクトに対してその分類のもとに属性付けを行うということは、分類による仮想化を表す。

分類と仮想化は、対になって行われると考えられる。例えば、ある分類項目を設定し基本オブジェクトを分類していく作業過程を考える。まず、分類の選択対象となるオブジェクトを選択する。これは、「作業領域」という仮項目に対して分類が行われたことを意味する。次に、選択したオブジェクトに対して、(必要ならば) 選択作業に応じた属性を付与する(仮想化)。仮想化されたオブジェクトを操作することにより目的とする分類項目への登録するオブジェクトを決定する(分類)。分類されたオブジェクトに対し、その分類の観点に基づいた属性を付与する(仮想化)。

このように仮想化を繰り返しながら分類作業(基本オブジェクトに対する統一的な意味付け)を繰り返す

うちに、徐々に個々の分類階層ごとに統一的な枠組みが見いだされていき、データベース化されていくことが期待できる。

我々のカテゴリを使って、検索結果を一つのカテゴリとする考え方は分類機構であるが、Rundensteiner からは検索結果を virtual class として扱うという構造化の方法として捉えられている研究もある。[7]

6 終わりに

階層構造グラフを用いた木簡研究支援システムにおけるスコープによる視点操作と分類階層について考察した。木簡システムのようなデータの特徴(大まかに格納されたデータが認識された属性として取り出されるような場合)は、木簡独自のものではなく、科学技術データを利用する立場で考えた場合、より広い範囲に適應されるであろうと思われる。グラフ構造とデータをリンクすることで、グラフからの伝播ビューによって自在なデータの描像が得られることから、データの再構築のインターフェースとしてこのデータモデルが活用できるものと期待される。

参考文献

- [1] IEEE Computer Society, "Special Issue on Scientific Databases," Bulletin of the Technical Committee on Data Engineering, vol.16, no.1, Mar. 1993.
- [2] Zdonik, S., "Incremental Database Systems: Databases from the Ground Up," Proc. of the 1993 ACM SIGMOD International Conference on Management of Data, Washington DC, USA, pp.408-412, May 1993.
- [3] Ueshima, S., Ohtsuki, K., Morishita, J., Qian, Q., Oiso, H. and Tanaka, K., "Incremental Data Organization for Ancient Document Databases," Proc. of the Fourth International Conference on Database Systems for Advanced Applications(DASF AA'95), pp.457-466, Singapore, Apr. 1995.
- [4] 上島紳一, 森下淳也, 大月一弘, 杉山武司, "階層構造グラフを用いた半構造化データの構造化手法," 情報処理学会論文誌, Vol.39, No.4, pp.857-867, Apr. 1998.
- [5] 山崎康雄, 古川哲也, 島崎真昭, "データベースを用いたデータ解析のための集合操作," 情処研報, DBS94-5, pp.39-48, Jul.1993.
- [6] Shoens, K. Luniewski, A., Schwarz, P., Stamos, J., Thomas, J., "The Rufus System: Information Organization for Semi-structured Data," Proc. of the 19th VLDB Conference, Dublin, pp.97-107, Ireland, 1993.
- [7] Rundensteiner, E. A., "Object-Oriented View Technology: Challenges and Promises," Proc. of the International Symposium on Cooperative Database Systems for Advanced Applications, pp.299-308, Kyoto, Dec. 1996.