オブジェクト指向データベースにおけるクラス階層構成法†

2R−5

端山　毅
NTTデータ通信(株)　開発本部

はじめに

クラスの継承はオブジェクト指向の主要な特長のひとつである。基本的なクラスを元に、種々の属性、メソッドを徐々に追加したサブクラスを作成することは、人間の直感にあい、また思考範囲の自然な限定である。

しかし、オブジェクト指向データベースを利用する場合、現実には、クラス定義をしてからデータを生成するのではなく、データが先に存在してこれに合ったクラス階層を定義しなければならない。多くのデータが与えられ、これらに対するクラス階層を構成してデータベースを構築することは、必ずしも容易な作業ではない。

本論文では、クラス階層構成の困難さ、好ましいクラス階層の指針、クラス階層構成の手続きについて述べる。

1　クラス階層構成の困難さ

本論文では、クラスを以下のように集合で表現する。

定義 1 クラス C が属性 $a_1, \ldots, a_n$ $(n \leq 0)$ を有するとき、

$$C = \{a_1, \ldots, a_n\}$$

と表す。

定義 2 インスタンスを有するクラスを**目的クラス** (target class) という。インスタンスを有しないクラスを**中間クラス** (intermediate class) という。

中間クラスはクラス階層定義上の利便性から用いられるクラスである。本論文では、メソッドも一種の属性であると見なす。

目的クラスの集合 $C_1, \ldots, C_m$ を考える。A を全属性の集合、すなわち、

$$A = \{a \mid \exists C_i\, (1 \leq i \leq m)\, a \in C_i\}$$

とする。属性の種類の数 $|A| = n$ 個とする。各クラスは A の部分集合であり、2^n 種類のクラスが存在し得る。A の部分集合の包含関係を $\subset$ で表せば、$L = \langle 2^{|A|}, \subset \rangle$ は束を構成する。図1に束の例を示す。多重継承を許すとすると、実際のクラス階層は L の部分束 L' である。ただし、仮想的に最大元 A と最小元 ϕ を加えて、束構造を保つとする。L から L' を構成する手続きは、L の 2^n 個のノードのうち、目的クラスの他、便宜上必要な中間クラスに相当するノードを選択し、これらのノードを接続する枝の集合を求めることに相当する。

多重継承を許さないと、クラス階層は木になるが、必要なノードと枝の集合を抽出することに変わりはない。

†Class Hierarchy Construction for Object Oriented Databases
Takeshi Hayama (NTT Data Communications Systems Corp.)

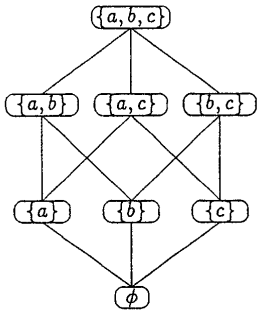

図 1: 束の例

このようにクラス階層を束として形式化する方法はすでに提案されている [2][1] が、適切なクラス階層を得る基盤としては、実際的に必ずしも有効でない。

n 種の属性が存在するとき、前記の束のノードの総数は 2^n 個であり、枝の総数は

$$\sum_{k=1}^{n} \frac{n!}{(k-1)! \cdot (n-k)!}$$

であり、

$$2^n < \sum_{k=1}^{n} \frac{n!}{(k-1)! \cdot (n-k)!} < n \cdot 2^n$$

である。

このような束構造を操作することは、計算機の記憶容量、処理速度からして限界がある。ワークステーションを用い数 MB の記憶領域で処理可能な範囲は、属性の種類が高々 20 種程度までである。

2　望ましいクラス階層

どのようなクラス階層が得られればよいのか、指針として以下の 3 つが挙げられる。

1. 中間クラスが少ない。
2. 追加される属性の総数が少ない。
3. 継承関係が少ない。

$A = \{a\}$, $B = \{a, b, c\}$, $C = \{a, b, d\}$ が目的クラスであるとき、図2のような2通りのクラス階層が考えられる。枝に付した数字は追加された属性数である。図2-(1)には中間クラスが存在せず、追加された属性数は合計4である。図2-(2)では $\{a, b\}$ という中間クラスが存在し、追加された属性数は合計3である。

多重継承を許す場合、クラス C_1, C_2 が $C_1 \subset C_2$ であっても、C_1, C_2 の間に継承関係を定義する必要は必ずしもないので、3. も考慮の対象となる。

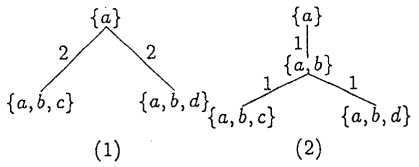

図 2: クラス階層の例

3 クラス階層構成手続き

本章では、目的クラスの集合を $T = \{C_1, \ldots, C_m\}$ とし、全属性の集合を A とする。

3.1 単一継承のみの場合

目的クラスの集合をひとつの属性に注目し、その属性を含むものと含まないものに分ける。注目する属性は、最も多くのクラスに共通して出現する属性とする。二つに分割されたクラスの集合を、同様にして再帰的に分割して、各属性を含むか含まないかによって2分木を構成する。出来上がった2分木から冗長なノードを削除し、n 分木にすればクラス階層が得られる。図 3 にこの手続

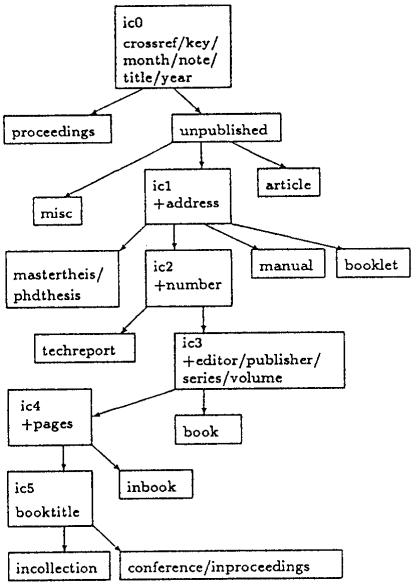

図 3: BIBTEX の文献クラス

きに基づいて、文献データベース BIBTEX [3] [5] [4] 用のクラス階層を構成した例を示す。ic[0-5] は中間クラスを表す。+address 等は追加される属性を示す。

3.2 多重継承を利用する場合

目的クラスの部分集合であるクラスの集合 $sub(2^{|A|})$ を、

$$sub(2^{|A|}) = \{C' \mid C \in T,\ C' \subset C\}$$

と定義する。$sub(2^{|A|})$ は中間クラスの候補となる集合(目的クラスも含む)であり、これらのクラスを中間クラスにしても属性定義の追加は発生しない。束 $\langle 2^{|A|}, \subset \rangle$ の代わりに $\langle sub(2^{|A|}), \subset \rangle$[1] からクラス階層を考え始めた方が現実的である。

$\langle sub(2^{|A|}), \subset \rangle$ から不要なクラスを削除する方法としては、以下の方法が考えられる。

1. 中間クラスを使用しない。
2. 中間クラスとして単一属性のクラス $\{a_1\}, \{a_2\}, \ldots, \{a_n\}$ のみを採用する。目的クラスは上記の n 個のクラスから多重継承を利用し、2段階の継承ですべて実現できる。各属性の定義は1回ずつしか行なわれず、中間クラスの数も属性の種類で抑えられている。
3. $sub(2^{|A|})$ のうち k 個以上の目的クラスの部分集合であるクラスのみを中間クラスとして採用する。例えば $k = 2$ または $k = 3$ とする。
4. 1. と 2. の方法を組み合わせ、単一属性のクラスと複数の目的クラスの部分集合であるクラスのみを中間クラスとして採用する。

属性数が少ない場合は、3. や 4. も使えるが、属性数が多くなると、このような制約を用いても、なお中間クラスや継承関係が相当な数にのぼる。

4 応用分野

データベースのユーザにとって同種のデータでありながら、その内部構造が異なる場合、それらを格納するためのクラス階層は、異なるデータがユーザーからは1種類のデータに見えることが望ましい。種々の構造を持ったデータを一様に操作できることは OODB の特長であり、このような場面では機械的にクラス階層を構築して構わない。ここで例に用いた文献データベースの他、複数の組織が有する同種のデータを統合する場合に、本論文で述べた自動的なクラス階層構築が有用である。

参考文献

[1] Bläsius, K. H. and Hedstück, U. Resolution with Feature Unification, *LNCS*, 329 (1987), 17–26, 1st Workshop on Computer Science Logic.

[2] Cardelli, L. and Wegner, P. On Understanding Types, Data Abstraction, and Polymorphism, *Computing Surveys*, 17, 4 (1985), 471–522.

[3] Lamport, L. LATEX: *A Document Preparation System*, Addison-Wesley (1986).

[4] 松井正一 日本語 BIBTEX:JBIBTEX, JBIBTEX C Version キット (ver.0.20 用, 1989 年) (1991).

[5] Patashnik, O. BIBTEXing, Documentation for general BIBTEX users (Jan. 1988).

[1] 束ではない。