オブジェクト指向ｖｓ関係データモデル

6H−5

穂鷹良介

筑波大学　社会工学系

1．本報告の目的

従来型のデータモデルの欠点を補うデータモデルとしてオブジェクト指向データベース／データモデル（以下ＯＯＤＢと略す）に大きな期待が寄せられている（[1]）。わが国においても実システムの開発が計画されるなど（[4]）今後のデータベース／データモデルに大きな影響を与えることが予想される。

一方現実のデータ処理の大部分は従来型のデータベースによってなされているのも事実である。もしもＯＯＤＢが本当に優れたものであるならば従来型のデータベースは漸次ＯＯＤＢによって置き換えられなければならないことになってこれは実務家にとって大問題となる。

本稿は両者の考え方の基本を考察することによりその本質を明らかにし，実務家がこの問題をどのように考えればよいか一つの判断基準を与えることを目的とする。

これらの考察を行うのに際してＯＯＤＢというものとして何を考えるのが適切であるかという問題が別途存在するが，ここでは非常に基本的な性質から考察を行おうとしているのでたとえば［3］で提案されているような非常に素朴なモデルを前提とする。

2．抽象化概念サポート

複雑な対象をモデル上で複雑さを厭わずに表現することが許されるならばモデルが複雑な対象を表現する能力について優劣を論ずる必要はない。複雑さをどれだけ容易に扱うことができるかが問題である。

明示的には述べられていないが，ＯＯＤＢが従来型のデータモデルに比較して複雑な対象の表現に適しているとされる理由の一つは抽象化の能力の差にあると考えられる。現在までによく知られている抽象化は次の３種である：

（１）リスト抽象　aggregation とも呼ばれ，いくつかの実体を定められた規則に従って並べたものを再び別の実体とみなす抽象化

すなわち今 $x_1,\ldots,x_n$ を n 個の実体とするときこれらを

$$[a_1{:}x_1,\ldots,a_n{:}x_n]$$

のようにまとめて一つのリスト実体 x とすることである。$a_1,\ldots,a_n$ をインスタンス変数，属性あるいは列などと呼び，$x_1,\ldots,x_n$ をそれらの値と呼ぶ。

（２）集合抽象　いくつかの実体をグループ化したものを再び別の実体とみなす抽象化

すなわち今 $x_1,\ldots,x_n$ を n 個の実体とするときこれらを

$$\{x_1,\ldots,x_n\}$$

のようにまとめて一つの集合実体 x とすることである。

（３）汎化　いくつかの実体型の実現値を互いに別の実体型に属するものと区別しないで，まとめて新たな別の実体型の実現値とみなす抽象化

すなわち今 $X_1,\ldots,X_n$ を n 個の実体型とするときこれら実体型のどの実現値も新しく設ける実体型 X の実現値であるとすることである。

これらのいづれの抽象化もＯＯＤＢは取り扱うことができる。具体的にはインスタンス変数の値としてリスト実体あるいは集合実体の値を認める。従来型のデータモデルの言葉でいうならば属性あるいは列の定義域として原子的な実体型だけでなくリスト実体型あるいは集合実体型を認めている。いわゆる非正規形関係を除いて普通の関係データモデルは定義域に原子的な実体型しか認めていないからこの点に関してＯＯＤＢは複雑な対象を適当に抽象化することによって表現し易くなっている。

汎化に関しては実体型間の汎化，その逆の専化あるいは特化をオブジェクトクラス間の汎化階層として実現している。関係データベースの代表であるＳＱＬにおいても汎化階層はまだ導入されていないから，この点に関してもＯＯＤＢの表現力の方が勝っている。

3．スキーマ記述

列の定義域がすべて原子実体型であるような簡単な応用データモデルのスキーマを記述することは可能でありかつ有効であるが，定義域にリスト実体型または集合実体型を再帰的に不定回数定義を許すたとえばＯＯＤＢの応用データモデルなどでは実務的に有効なスキーマを定義することが今までのところなされていない。

4．実現値表現中の実体型識別情報の存在

ＯＯＤＢの oid の特徴はそれを指定したときどの実体型（ＯＯＤＢではオブジェクトクラス）に属するものであるかが分かっていることである。関係データベースなどの従来型のデータベースにおいては属性の値は単なる値であってそれがどの実体型に属するものであるかは値のほかに実体型の指定を別途行うことを前提としている。汎化を実現するデータモデルにおいては，oid が与えられたときそれが属する実体型が分かるメカニズムがないといけないが，そのためには oid の一部に実体型を指定

Object-oriented vs. Relational Data Models
Ryosuke HOTAKA
Univ. of Tsukuba

する情報を埋め込むことが必要となる。たとえば Smalltalk/V ([2]) においては oid は５バイトでそのうち２バイトがオブジェクトクラスを現す。

実現値からそれが所属する実体型を知ることができるのは利点であるが，同時にこれは複数のＯＯＤＢが互いにデータを共有するとき oid の定め方に別途管理機構を設けなければいけないことを意味する。

実現値表現中の実体型識別情報は実現値がどの実体型に属しているかが判明しないときに処理上必要とされるものであるが，所属実体型が何らかの方法によって別に識別できるときには絶対に oid の中に埋め込んで置かなければならないと考える必要はない。事実従来型のデータベースにおいては実現値に所属実体型情報を入れずに情報の交換を行っているが，複数のデータベースシステム間でデータの共有を考えるときにはこのほうが面倒が少ない。

したがって一つのデータベースシステム内でデータの処理を行っている限りに置いてはＯＯＤＢの oid 方式はそれなりにメリットがあるが共有を目的としたデータの蓄積の場合には従来方式の実現値表現を併用する必要がある。[6] は管理実体型概念の導入によってこれを果たそうとしている。

５．共有記述

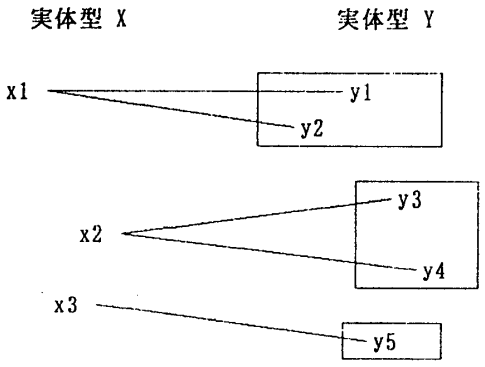
ＯＯＤＢは oid が複雑な対象を表現する能力を持ちかつその値はシステム内部で定められたもので利用者が直接指し示すことができるものではない。一つの対象について様々な角度から複数人が共通の対象を直接参照する目的には不向きでありまたその必要もすくないと考えられる。

ＯＯＤＢのモデル化の第一の目的は対象を忠実に模写することにあってその共有については考慮が不足しているように思われる。これに対して関係データモデルでは直接対象の構造を表現するというよりは対象を利用者が指し示すことができる値を用い，その対象について記述を行うことを目的としている。共有を目的とするデータの表現には対象を値で指し示す必要がある。なおサロゲートを用いてのみ表現される複雑な対象の表現の共有は一度それに「名前」という値を与えた後にそれを用いて記述を行うという形でなされるのではなかろうか。

６．表現の一意性

以上考察してきたように理想的なデータモデルはＯＯＤＢ，関係データモデル両者の機能を統合したものになりそうである。そのとき両者の機能をどのように使用して応用データモデルを決定するかということが問題となる。[5] では表現の一意性が必要とされているときにどのようなモデル化概念を使用すべきかということに関して観察を行った。この問題は応用データモデルの設計問題で全容はまだ分かっていない部分が多いが，集合抽象化を用いた場合表現の一意性が保てなくなることが指摘されている。したがってどのようなときに集合抽象化を用いるかという規準を示すことが問題となる。筆者が提案する規準は次のようなものである。

集合実体はその構成要素に自立性がないときに用いる。換言すれば集合実体の構成要素に自立性があるときには集合実体と構成要素としてモデル化しようとした実体との間の関連の記述として述べる（図１）。



もしも xi が削除されたときその構成要素 yj を無条件に削除することができないときには実体型 Y は自立性を持つと考えて上記のような集合抽象化は行わずに両者の実現値の間には単に関連があるとしてモデル化する。

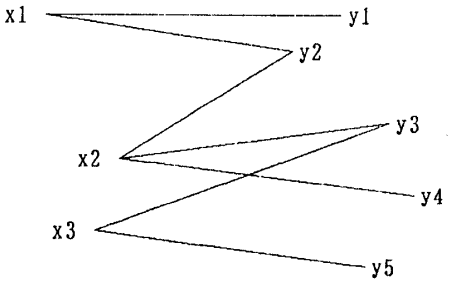


図１　実体型の自立性に依存した集合抽象化の使用

参考文献

[1] Atkinson,M. et al.: The Object-Oriented Database System Manifesto, The first International Conference on Deductive and Object-Oriented Databases (DOOD89), Elsevier, Amsterdam, 1990.
[2] Digitalk Inc.: Smalltalk/V Tutorial and Programming Handbook, August 1987
[3] George Copeland and David Maier: Making Smalltalk a Database System, ACM SIGMOD'84 pp. 316-325.
[4] 田中克己ほか：Ｏｂａｓｅプロジェクト，データベース・システム 84-21，情報処理学会 1991.7.18
[5] 穂鷹良介：標準データモデル機能のモデル化概念，Proc. Advanced Database System Symposium '91, Tokyo, Japan, December 5-6, 1991, pp.71-76.
[6] 穂鷹良介：管理実体型概念について，データベースシステム研究会，情報処理学会 1992.3