

## データベース管理システム「沙羅」の機能拡張

2R-1

石丸 知之 植村 俊亮

東京農工大学 工学部 電子情報工学科

### 1. はじめに

文書処理、CAD、マルチメディア処理、そしてハイパーテキストで代表される最近の応用プログラムのためのデータベース管理システムを目指して、われわれは、データベース管理システム「沙羅」を試作中である[1]。

沙羅のデータモデルは、オブジェクト指向言語の考え方を取り入れている。これは、オブジェクト指向データモデル[2]に分類できる。

最初の試作では、基本と考えられる機能だけを取り入れたが、明らかに機能不足であった。本稿では、試作の経験をもとにした沙羅の機能の拡張について述べる。

### 2. データモデルに追加する機能

現在の沙羅のデータモデルでは以下のような問題がある。

- (1) 一つの実体に対する異なる表現をうまく取り扱えない。
- (2) 継承の種類のひとつの「制約 (constraint)」がうまく扱えない。
- (3) 参照が単純すぎる。Cのポインタと同じレベルである。

沙羅の第2版でのこれらの問題を解消するためにデータモデルを拡張した。その拡張点について述べる。

#### 2.1 マルチクラスインスタンス

通常の、オブジェクト指向データモデルでは、インスタンスはただひとつのクラスに属する。

オブジェクトは、実世界の存在のデータベース内での表現である。増永[3]が主張するように、実世界の存在をデータベースに写像する方法は、唯一ではない。たとえば、実世界の人物の表現には、写真のようなイメージによるものと、履歴書のような文字データによるものが考えられる。これらのデータは、ひとつの実世界の実体の異なる表現である。オブジェクトの存在について一貫性をとらなければならないこともある。

たとえば、その人物の情報がデータベースから取り除かれるときには、イメージによる表現と文字による表現を同時にデータベースから取り除くことがある。

このように、実世界の存在に対するオブジェクトの表現を複数使用するために、沙羅では異なるクラスのオブジェクトをひとつのオブジェクトに結び付ける機構を導入する(図1参照)。ひとつに結び付けられたオブジェクトは参照時に使用するクラスによって見え方が異なる。たとえば、イメージによる表現と文字による表現のふたつのインスタンスをひとつに結び付けたインスタンスは、参照時の指定によってイメージとして呼び出されたり、文字データとして呼び出されたりできる。

このようなインスタンスは、属するクラスが保持する属性をすべてもつ。ただし、それらクラスが共通のクラスから属性を継承してるときには、インスタンスでは、ひとつの属性を共有する(図2参照)。

マルチクラスインスタンスは、多重継承と異なっている。多重継承では、クラス定義で複数のクラスの性質を引用している。また、必要に応じてスーパークラスの属性やメソッドを利用した新しい性質を定義できる。それに対してマルチクラスインスタンスでは、インスタンスの構造をひとつのオブジェクト内に格納するだけで、クラスが異なる属性の融合はできない。

#### 2.2 制約継承

オブジェクト指向データベースで提案されている継承の方式は、(1)代用(substitution)継承、(2)包含(inclusion)継承、(3)特殊化(specialization)継承、そして(4)制約(constraint)継承の4種の継承に分類できる。[1]

これらの継承の方式の最初の3つは、オブジェクト指向データベースでは、これまで自然に取り入れられている。

残った制約継承をデータモデルに取り入れるために、スーパークラスのインスタンスのうちで制約または条件を満たすものをインスタンスとしてもつクラスを作成できるようにデータモデルを拡張する。制約継承す

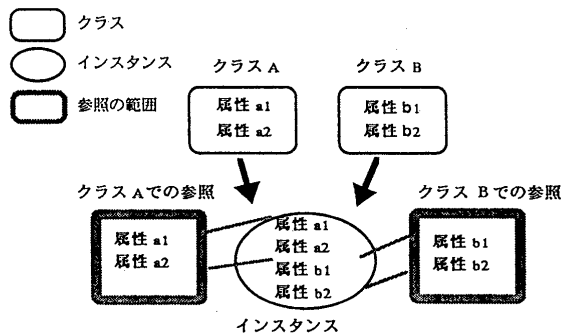


図 1. 複数のクラスに属するインスタンス

るクラス（「制約クラス」と呼ぶ）は、指定したスーパークラスと比較して、メソッドや属性の追加や変更がない。しかし、属性の値に対する条件によって、そのクラスのインスタンスの集合は、スーパークラスのインスタンスの集合の部分集合となる。

このような方式では、インスタンスをスーパークラスと共有するので、スーパークラスと制約クラスでのインスタンスの一貫性は保証されている。

制約クラスのインスタンスは、スーパークラスと制約クラスの複数のクラスに所属する。これは、2.1 で述べたマルチクラスインスタンスの機構を利用している。

### 2.3 属性

オブジェクト指向データベースでの属性の指定は、属性となるオブジェクトのクラスの指定や inverse attributes のような参照の指定程度しかできなかった。

親子集合をもとにした CODASYL 方式のデータベースでは親子属性の種類を、挿入属性については、手動 (MANUAL) と自動 (AUTOMATIC)、保留属性については、一時 (OPTIONAL)、永続 (MANDATORY)、そして固定 (FIXED) と分類している。[4]

オブジェクト指向データベースでも、このような属性の代入の制御が必要である。

人物のデータを格納した人事データベースをオブジェクト指向データベースでマルチメディア化した場合も、データ管理の基本はこれまでのものと変わらない。たとえば、写真データはある人物に結び付けられているときに意味をもつ (保留属性が固定) という情報が必要である。

このような属性に関する制約を沙羅のデータモデルに追加する。CODASYL 方式の属性の制約は親子構造に関したものであったが、オブジェクト指向データベースではオブジェクトを構成する要素としての属性に関する制約となる。

つまり、あるクラスのインスタンスを作成したとき

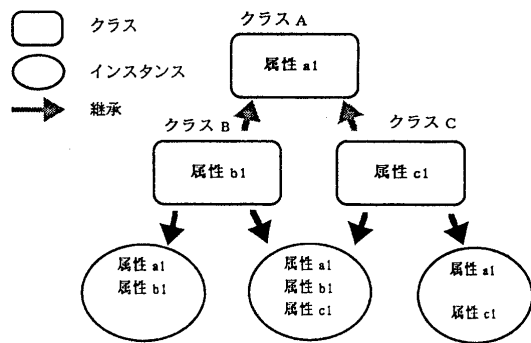


図 2. クラス階層とインスタンスの属性

に、そのインスタンスが属性となるインスタンスを設定する機能や、あるクラスのインスタンスはそのインスタンスを属性としてもつインスタンスの生存期間に依存する機能をもつという機能を提供する。

このような属性の制約は、inverse attributes とは異なっている。inverse attribute は、ふたつのインスタンスでの参照の方式を、片方から参照しているときには必ず他方からも参照するように指定するものである。

それに対して、属性の制約は、インスタンスを作成したときの参照の作成や、参照の解消の制約を規定している。したがって、たとえば、保留属性が固定でありときの参照は、inverse attribute であっても、そうでなくてもかまわない。

### 3. おわりに

沙羅の第2版の設計について述べた。現在、この設計に基づいて、実装を進めている。

実行系は仮想機械方式を採用している。仮想機械の水準で、データモデルを実現している。

まだ、データベースと呼ぶためには機能が不足している部分は、サーバ機能、仮想機械のマルチプロセス化 (ライトウェイトプロセス)、トランザクション処理などである。

### 参考文献

- [1] 石丸他, "データベース管理システム「沙羅」の設計と試作", 電情学会研究報告, DE92-1, 1992
- [2] Atkinson, M., et al, "The Object-Oriented Database System Manifesto", Deductive and Object-Oriented Database, North-Holland, 1990, 223-240
- [3] Masunaga, Y., "Design Issues of OMEGA: An Object-Oriented Multimedia Database Management System", J. IPS Japan, 14, 1, (1991 Jan), 60-74
- [4] 植村, "データベースシステムの基礎", 1979, オーム社