

9G-2

拡張リレーショナルDBMS上でのオブジェクト指向DBMS の実現方式について

藤田一哉† 藤代一成† 池辺八洲彦†
† 筑波大学 † お茶の水女子大学

1 はじめに

近年, CAD/CAM等の高度なデータベース応用が求められるにしたがって, マルチメディアデータのような複雑な構造をもつデータ型(即ちオブジェクト)の必要性が高まってきている. そして, このオブジェクトを扱うオブジェクト指向DBMS(OODBMS)の研究・開発が現在盛んに行われている.

我々はコスト面を重視し, オブジェクト指向データモデル(OODM)を既存システムがサポートするデータモデルへマッピングするという, 既存システムを活用する方法を用いて, OODBMSの実現を試みている. 我々が注目したターゲットデータモデルは, (株)リコーで研究・開発された拡張リレーショナルDBMS "G-base¹"に採用されている, グラフデータモデル(GDM)である. GDMは, 従来のリレーショナルデータモデルに2種類のリンク構造を導入したもので, その結果, 実体間の関連を明示的に記述することができ, 実世界の自然な表現を可能にしている. このリンク構造の存在が, 本システムの実現を非常にシンプルなものにしている要因の1つである.

本システムは,

1. スキーマトランスレータの製作 [Schr85].
2. データ辞書(DD)の定義.
3. 問合せシステム(トランスレータ)の製作.

という流れに沿って開発されてきており, 現在は第2段階まで進み, 基本的なデータ定義をGDMスキーマにマッピングすることが可能である.

本稿では, データモデルのマッピングと, それに関連してサポートしなければならないメタデータの概要について述べる.

Implementation of an object-oriented DBMS on an extended relational DBMS

Kazuya Fujita† Issei Fujishiro† Yasuhiko Ikebe†
† University of Tsukuba † Ochanomizu University

2 オブジェクト指向データモデル

本システムに採用したOODMは, 米国MCC社の開発したOODBMS "ORION"に採用されたCore Model [Kim90, Chap.2]に基づいている. その中では, データモデルの細部の定義に関していくつかの選択候補が提案されており, 以下に我々が選択した結果を示しておく.

- 全継承のサポート.
- 多重継承のサポート.
- クラス合成階層中のサイクルの否認.
- インスタンスの所属を単一クラスに限定.
- クラス属性(共有・欠落)の採用.

3 マッピング

両データモデルはダイアグラムも類似しており, 各データ構成子が自然に対応づけられる. データ構成子の対応の概要を〈表1〉に与える.

表1: データモデル間の対応関係

OODM	GDM
オブジェクト識別子	レコード識別子
クラス	レコード型
クラス階層	仮想リンク型
クラス合成階層	実リンク型

このようなシンプルな対応関係は, システムの開発を容易にし, それだけコストの軽減につながるといえる.

以降, 本節ではマッピングの中核をなす上述の2種類の階層(リンク型)について, その対応の理由と実現法を述べる.

¹G-baseは(株)リコーの登録商標である.

3.1 クラス階層と仮想リンク型

まず、クラス階層と仮想リンク型の対応であるが、これには2つの大きな理由がある。1つ目は、仮想リンク型が一对多(一)の関連を示すものであることである。クラス階層とは、クラス間の特化/汎化関連を示すもので、その利点としては、

1. インスタンス集合が細分化され、効率の良いデータ操作を行うことができる。
2. 継承の概念により、属性の重複定義を避けることができる。

等が考えられる。これらの効果を実現するために、1つのインスタンスが持つ各属性値を、そのインスタンスが属するクラスとそのスーパークラス群に対応するレコード型に分散させた。そして、これらの間はキー値による一对一の参照がなされる。本システムのこのような性質から、データベース内のインスタンスの論理的な存在可能範囲をただ1つのクラスに制限した。

2つ目の理由は、仮想リンク型が暗黙的に持つ、参照の一貫性制約の存在である。このため、クラス階層に存在する継承の概念を、本システムで制約として明示的に定義しなくても取り入れることができる。

3.2 クラス合成階層と実リンク型

次に、クラス合成階層と実リンク型の対応であるが、これは実リンク型が、多対多の関連を表すものであるためである。クラス合成階層は属性の定義域を表すという性質から、全く値は等しいが異なるオブジェクト(つまり同形(isomorphic)なオブジェクト)が複数存在する可能性が高い。この同形なオブジェクトは、それぞれを参照するオブジェクトと一对一の関係を示す。また、1つのインスタンスがある属性に複数の値をとることが考えられ、これは一对多の関係を示す。しかしクラス階層の場合と異なり、クラス合成階層には、参照される定義域クラスのキー値が参照するクラスに存在しなくても、システムの性質には問題がなく、仮想リンク型にみられるような一貫性制約も必要ないといえる。このような理由から、実リンク型を選んだ。

4 データ辞書

OODMとGDMは非常に良いマッチングを示すが、両データモデルは根本的に概念が異なるため、一貫性

制約(規則)などのデータモデル間の差分情報をメタデータとして管理しなければならない。そこで本システムの要求を完全に満たすように、"G-base"のDDを核として、不足点、変更点を取り入れたDD(これをオブジェクト指向DD(OODD)と呼ぶことにする)を定義する必要がある。

スキーマ定義に関して、データベーススキーマ情報の実質的な保持はcoreDD部が行っているため、入力されたDDLファイルがGDMの形式に変換されたときに生じる、GDM側で単純に表現しきれないもののみを改めてOODDに記述する。しかし、現在はシステムを稼働させるのに必要な最低限の定義しかできていない。今後定義する必要があると思われる項目を以下に挙げておく。

- クラス属性の記述。
- メソッドの記述。
- オブジェクトのバージョン管理情報。
- オブジェクト指向スキーマ上での authorization。
- スキーマの動的変更時における一貫性制約。

また、本システムでは属性名の二重化やクラス合成階層のサイクル化など、データベースの正しい状態を保持することを妨げる可能性のある概念や処理の採用を見送ったため、DBMSがこれらの排除を行うようにDDLプロセッサを拡張した。

5 まとめ

本システムでは、現在静的なスキーマのトランスレーションが可能である。今後は、メソッドの採用など、DD部を中心とした未定義項目のサポートと既存部分の最適化、そして開発過程の第3段階である問合せ機能部の製作を行っていく方針である。

参考文献

- [Kim90] W.Kim: *Introduction to Object-Oriented Databases*, The MIT Press, 1990.
- [Schr85] Axel T.Schreiner and H.George Friedman, Jr: *Introduction to Compiler Construction with UNIX*, Prentice-Hall Inc., 1985.