オブジェクト指向データベースにおける制約表現とその管理*

4H−4

大本 英徹, 神谷 誠, 田中 克己†
神戸大学‡

1 はじめに

オブジェクト指向データベース管理システム (OODBMS) が関係データベースシステムと比較して，一般により優位にあるとされる理由として，1) 強力なデータモデリング能力，2) データとその操作 (メソッド) の一体化，3) クラス階層・型階層による属性構造やプログラムの差分的定義，等がある．しかし，現状のOODBが有する問題点もクローズアップされ，その解決へ向けた研究が急務となっている．

本稿では，オブジェクトに課せられた制約を，特にデータベース中のオブジェクトとして格納・管理することで制約管理を行なうことを考え，その基本的検討課題について考察した．

2 OODBの制約表現・管理の要件

現存のOODBMSでは，オブジェクトに課せられる制約を維持・管理する機能をサポートするものは，ほとんど無い．データベースの一貫性管理という点から，制約管理のサポートは早急に解決されるべき問題であるが，その際，以下のような事項を考慮する必要がある．

- オブジェクト識別子 (oid) に関する制約表現とその管理機構が必要．
- 制約の変更に伴うスキーマの再コンパイルは行なうべきではない．
- 制約には様々なレベルが存在する．厳守されるべきもの，条件付きで破ってもよいもの，等が考えられ，これらのレベルを制約自身が表現可能であるべきである．
- OODBの大きな特徴である継承機構をうまく利用できるような機構であることが望ましい．例えば制約の継承やそれによるGenericな表現が行なえるべきと考える．
- 制約自身に対して制約が存在することが考えられる．その記述も出来れば同一の機構で表現できるのが望ましい．

3 OODBの制約の分類

OODBにおいて，オブジェクトに課せられる制約の種類として次のようなものがある．

- 定義域制約：オブジェクトの属性値に関連する制約．例えば，ある住宅地における住宅の建蔽率は80%以下でなければならない等という住宅オブジェクトに関した制約が考えられる．
- 関連制約：オブジェクト間の関連に関する制約であり，さらにいくつかに分類できる．
  - オブジェクト間の関連制約: 例えば，あるクラスのオブジェクトと別のオブジェクトの参照関係は1対多関係にある，等がある．異なる従業員は必ず異なる従業員番号を有する等という，いわゆるキー制約もこの範疇に含められる．
  - オブジェクト名/オブジェクト間の関連制約: オブジェクトに対して名前付けが可能なOODBMSでは，この名前とオブジェクトとの対応関係に関する制約も重要となる．
  - oid/属性値間の関連制約: 既存のOODBではoid管理の制約を表現・維持する機能がほとんどサポートされない．このoid管理制約を表現する概念として経路関数従属性 (PFD)[1] がある．これは関係データベースにおける関数従属性の拡張となっている．例えば，異なった自動車はそれぞれ異なったエンジンを持つはずであり，共有することは有りえない．自動車オブジェクトのEngine属性にエンジンオブジェクトが入っているとすると，この制約をPFDで表現するとEngine ⇒ IDと書ける．このPFDを用いてComposite Objectのexclusive制約 [2] が表現できる．また，キー制約と同等の表現も可能である．

4 制約の表現・管理方法

1. メソッドによる表現 この方法は通常のメソッドコードとして制約を表現するものである．現状ではメソッドの変更がなされるとスキーマの再コンパイルを必要とする．

2. DBMS固有の特殊機能 一部のOODBMSは限定的なオブジェクト間の関連制約を維持する機能を提供する．しかし，ユーザが必要に応じて拡張することが出来ない．

3. ルールによる表現 演繹オブジェクト指向DB(DOOD)やActive DBでは，ルールを用いて制約管理を行なうことが可能である．この方法は既存のOODBに対して適用することが難しい．

4. オブジェクトとしての表現 SciorはAnnotationと呼ばれる属性に付随するオブジェクトを用いて，制約のチェックを行なうことを提案している [3]．オブジェクトとして制約を表現するアプローチにおける利点を次節に示す．

5 オブジェクトによる制約の表現

ここでは，データベース内の任意のオブジェクトの集合sに関する制約cを，次のような対で表現する．$c = < s, p >$ ここで，s：データベース中に定義可能な任意のオブジェク

---

*Representation and Management of Constraints in Object-Oriented Databases

†Eitetsu Oomoto, Makoto Kamitani, and Katsumi Tanaka
‡Kobe University

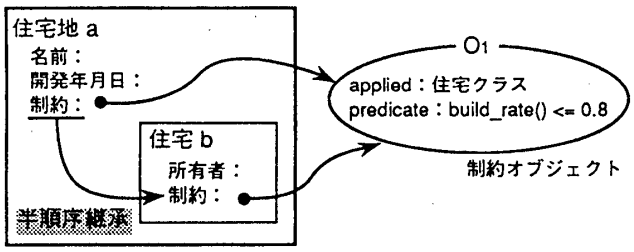

図 1: 半順序継承による制約の継承

ト集合，$p : s$ 中の任意の要素オブジェクトに関する述語である．

例えば，全ての住宅の建蔽率は 0.8 以下であるという制約は，次のように書ける (C++風の表現を用いている)．

$$< \text{extention(住宅クラス)}, \text{x.build\_rate()} <= 0.8 >$$

但し，extention( ) は任意のクラスの外延を返す関数，x は集合 s 中の任意のインスタンスを示す変数，build_rate() は住宅の建蔽率を返すメソッドとする．

また，被制約オブジェクトの集合である s の指定を，例えばSQL 質問で行なうことにより，制約を受けるオブジェクト群の決定を動的に行なうことも可能である．

この対を属性値を用いて表現するオブジェクトをデータベース中のインスタンスとして管理し，それら制約オブジェクトに基づいて制約管理が行なうことができれば，以下のような利点が生じる．

・制約オブジェクトに対する操作が可能
各制約オブジェクトは基本的に DB 中のオブジェクトとなるから，それらに対する種々の操作が可能となる．さらに，制約の変更もオブジェクトに対する通常の操作となるから，変更に伴うスキーマの再コンパイルを必要としない．

・属性値付加により各種付加情報を持たせることが可能
例えば，CAD 等の設計データベースでは制約条件を満たさないオブジェクトは必ずしも存在を許されないわけではない．こういう場合，各制約には様々なレベルが設けられるべきであり，それらのレベルを表現する情報を制約自身に付随させることができる．

・継承機構の恩恵を受けることが可能
制約自身がオブジェクトであれば，クラス階層による属性構造やメソッドの継承が可能となる．また，我々は半順序関係に基づく継承機構についても，すでに検討を行なってきている [4]．これはオブジェクト間の属性値の半順序関係 (例：地理データベースにおける 2 次元的包含関係) に基づくインスタンス間の継承機構であり，これとの組み合わせにより，制約が属性値として表現できるならば，従来，困難であった制約の表現が可能となる．図 1に例を示す．

住宅の建蔽率に関する制限は各地域によって異なり，例えば住宅地クラス定義中に，その中に存在する住宅の建蔽率制約を記述することは困難であり，住宅クラスに於ても同様である．ある住宅地オブジェクト a 内では各住宅の建蔽率が 0.8 未満に制限されているとする．この建蔽率に関する制約は "制約" 属性の制約オブジェクト o を入れることで表現されており，この建蔽率制約属性とその値が領域の包含関係に基づく半順序継承により，住宅オブジェクト b に継承されているとする．

制約オブジェクトには次のような属性を定義する．applied 属性：制約の適用対象オブジェクトの定義．例えばクラスの外延などが入る．predicate 属性：制約を受けるオブジェクトに適用される述語が入る．各住宅オブジェクトに "制約" 属性が存在する場合，住宅オブジェクトが自分自身に制約を満たしているかどうかを確認するには，自分自身を引数とするチェックメソッドを制約オブジェクトに送る．制約オブジェクト中のチェックメソッドでは，パラメータとして渡されたオブジェクトが applied 属性の値の要素となっているかどうかを判定し，要素であれば predicate を評価して真偽値を返すといったようにしておけば，制約を満たしているかどうかの確認が出来る．

・制約に関する制約の表現が可能
制約自身がオブジェクトとして表現される結果，その制約オブジェクト自身に関する制約が同一の機構で表現できる．

6 今後の検討課題

制約表現モデル
制約を表現するモデルとして，データベース中のオブジェクトとして管理する方法は，利点がいくつか有るが，制約オブジェクトの構造や有すべきメソッドなどの詳細な検討がさらに必要である．これは，実装の可能性とも密接な関係がある．また，オブジェクトの属する "クラス" に課せられる制約を表現可能であるかどうか，可能であるとしてそれは継承可能であるのか，半順序関係継承機構と組み合わせた場合に不都合な状況は発生しないのか，等といった問題がある．

制約オブジェクトの実現
現状の主流となりつつある C++では，データをプログラムとして動的に解釈・実行することが一般に出来ない．すなわち，制約を表現する述語を属性値とし，実行時にその評価を行なうことは一般に困難である．しかし，一部の OODBMS では SQL 質問を文字列として動的に生成・実行できるものがあり，述語を評価する機構として質問処理系を利用することが出来る可能性がある．

7 まとめ

本稿では，オブジェクト指向データベースにおける制約をオブジェクトとして表現する方法について，いくつかの利点と問題点について考察した．今後は表現形式のさらに詳しい検討を行ない，実際に OODBMS を用いて実装することを検討している．

参考文献

[1] Weddel, G. E.: *A Theory of Functional Dependencies for Object-Oriented Data Models*, Elsevier Science Publishers B.V., 1990.

[2] Kim, W, Bertino, E., and Garza, J.F., *Composite Objects Revisited*, Proc. of ACM SIGMOD, pp.337-347, June 1989.

[3] Sciore, E.: *Using Annotations to Support Multiple Kinds of Versioning in an Object-Oriented Database System*, ACM Trans. on Database Syst., Sept. 1991.

[4] 大本, 田中,「空間的・時間的オブジェクトの包含関係に基づく継承機構とその物理的記憶構造」, 情報処理学会研究報告, 91-DBS, 84-26 , 1991.