

## 管理実体型概念について

穂鷹良介  
筑波大学 社会工学系

従来型のデータモデルに対比してオブジェクト指向データモデルの一つの著しい特徴として、オブジェクト識別性が挙げられている。従来型のデータモデルにおける類似概念は「値」を基礎とした識別キーによる対象の識別であるが、それに対し様々な不適切さがオブジェクト指向データベースの立場から主張されている。

一方実務家の立場からは「値」に基礎を置く識別キーには本質的な必要性がある。本報告はこれらの異なった考え方を再考し、オブジェクト指向データモデル提唱者の指摘をどのように考えれば従来型のデータベースで行われている考え方と調和するかを考え、識別キーの取扱いに関して「管理実体型」概念を提唱する。

## On the concept of Controlled Entity Type

Ryosuke HOTAKA  
Inst. of Socio-Economic Planning, Univ. of Tsukuba

As one of the significant characteristics of object-oriented database compared to those of traditional data model, object identity is proposed. The equivalent concept observed in the traditional data model is the identifier key concept based on its "value", but advocates of object-oriented database pointed out several drawbacks of the value-based identifier.

On the other hand, value-based identifiers seem to be indispensable from the points of view of practitioners. This paper reconsiders these different opinions and proposes a concept "controlled entity type" in trying to integrate both traditional and object-oriented databases.

## 1. はじめに

従来型のデータモデルに対比してオブジェクト指向データモデルの一つの著しい特徴として、オブジェクト識別性 (object identity) が挙げられている ([Atkinson 89])。

従来型のデータモデルにおける類似概念は識別キーによる対象の識別であるが、[Copeland 84], [Kaehler 83], [Kent 91], [Khoshafian 86] 等は様々な理由を挙げてその不適切さを主張している。

一方多くの実務では従来型のシステムを利用している都合上、対象の表現に識別キーを利用せざるを得ないのが現状であり、もしもオブジェクト指向データモデル提唱者のいうように識別キーの利用が欠点を有しているならば従来型のデータベースシステムを早晩破棄し、オブジェクト指向システム等に移行しなければならぬこれは実務家にとって大問題となる。

本報告は上記論文等で指摘された識別キーの問題点を再考した結果

- (1) オブジェクト指向データモデル提唱者の指摘する通り識別キーの使用に関しては誤った設計に至る可能性がある。
- (2) しかし、そのような誤った識別キーの使い方は設計の段階で避けることができる。
- (3) それだけではなく識別キーの使用なしには複数の利用者の中でデータの共有を図ることは実質上不可能である。
- (4) したがって識別キーの使用は未来永劫避けることはできない。

ことを主張し、従来型のデータベースシステムを用いても使い方がえ誤らなければ別段問題はないことを示す。また識別キーの誤った使用を未然に防ぐために管理実体型というデータモデリングのための概念を導入し、その使い方を説明するのを目的とする。

なお管理実体型概念はデータモデル機能の標準として日本規格協会で検討が進められている JDMF ([情報資源スキーマ調査研究委員会 92]) にデータモデル機能の仕様として取り入れられているが、これはデータの共有を目的とするデータベース設計のベテランならば無意識のうちに考慮している考え方である。

## 2. 識別キーに対する批判とオブジェクト指向データベースでの提案

[Khoshafian 86] は関係データモデルで用いられている利用者が定義する識別キーに対して次の4点を問題として挙げる。

- (1) 識別キーは利用者が定義したものであるにもかかわらずそれを変更することはできない。
- (2) 識別キーは関係データモデル内のすべての対象を識別できない。
- (3) 識別キーとしてどの属性を選ぶかに恣意性がある。
- (4) 識別キーを利用しているときには検索時にオブジェクト指向データベースで容易に利用できる経路表現ではなく結合 (join) を利用しなくてはならない。

[Copeland 84] はどの対象を表現したデータも1単位として参照できる機能を実体識別性 (対象識別性 (object identity) と同義と考えられる) と呼び、例えば

- (5) 二人の人が同一の子供の集合を有しているということを関係データモデルでは一つの単位として参照することはできない。

ことを主張し、[Copeland 84] と [Kaehler 83] は

- (6) 関係データモデルにおいては一つの実体が他の実体を参照するためには他の実体を一意に識別するフィールドが存在しなければならぬとし、例えば「部名」を用いて関係「部」の組を識別するのは「部」の名前が変更になるまでは支障がない

が「部名」が変更になったときにはこの識別方法は不適である。  
という意味のことを述べている。[Kent 91] は対象の識別は認識できる事象の生起に遡るとし

(7) オブジェクト指向データベースでは異なる事象 (event) の生起に異なるオブジェクト識別子を与えるのに対し、関係データベースのような値に基礎を置くシステムでは対象の識別と存在に関して対象の生起に明確な事象を有しないから、識別に関してオブジェクト指向システムに比較して十分良いモデルを提供していない。と述べた。[Kent 78], [Kent 79] は以上のほかに例えば一つの実事「車をある使用者に使用させる」ということを関係データモデルで「車」, 「使用者」の二つの属性を用いて表現するとき

(8) 使用者が「個人」であったり、「部署」であったりするとき「使用者」の属性値として「個人」, 「部署」の両者を区別する等質な (homogeneous) 識別キーを考へることができない。  
と批判した。

オブジェクト指向データモデルでは、識別キーに替わってオブジェクト指向データベースが発行するオブジェクト識別子を用いることによってこれらの問題を解決できると主張しているように思われる。

### 3. 識別キーに対する批判点の再考

2. に挙げた識別キーに関する批判点を再考してみる。

識別キーの目的は対象を正確かつ容易に識別することでその目的に照らして不適当な使用のされ方がもしあるとすればたとえそれが可能でも一定の制約を課さなくてはならない。関係データベースの場合、識別キーは属性として実現されることになるが、属性として可能なことをすべて許すと結果として不適当な使い方もできてしまうことに気がつくなくてはならない。

2. (6) で指摘された批判点はこの一例である。「部名」は「部」を識別するという目的を持っているが、誤り易いことにこれを単なる属性として考えてしまうと「部」の「名前」であるから、「部」に与えられた「名前」が変更になったときは当然一度与えたある部の名前を変更するべきであると期待されてしまう。これは「部名」に

(a) 対象を識別する  
という目的と

(b) 対象のある属性を表現する

という相容れない二つの目的を持たせたことに原因がある。識別キーの目的は対象を識別することで対象自身は新しく応用データモデルに表現の対象として登場するか、あるいは表現の対象でなくなるかの何れかであって変更されるということはない。識別キーが表現している対象に変更が無い以上、識別キーを変更するということは許すべきではない。この点2. (1) で識別キーの値を変更できないことを批判の対象としていることは上記 (a), (b) を混乱して問題の本質を理解していない言及である。

2. (3) では識別キーを選択するのに恣意性があると批判しているが、識別キーは対象を識別するという目的のためだけに設けられるべきであって上で指摘した類の混乱を招くような属性を選んで識別キーとすべきではない。対象を識別しようとするならばその目的のためだけに特別な識別キーを設けるべきであって、これは識別しようとした対象が定まったときに同時に定めるべきものである。つまりある対象の識別キーはその目的のため

だけに設けられたもので他に別の意味を持っている属性を用いることはできないという意味でそこに恣意性はない。

[Koshafian 86] は 2. (2) 関係データモデルのすべてのオブジェクトに対して識別キーを提供できないという批判の例として次を挙げている。

「ある従業員オブジェクトが彼の配偶者を記述する属性において彼または彼女のファーストネームを用いていたとする。後日その配偶者がまた従業員になったときに配偶者の識別方法が変更になる不都合がある」

確かに識別を要する対象に対しては上で考察したようなその対象を識別する目的のためだけに導入された識別キーを用いるにしても応用データモデルに取り入れられたすべての対象について識別キーを設けることはできない。なぜならば識別キーは人為的なものであって更に他の属性によって説明される必要があり、そのような属性の値が再び識別キーである場合にはまた他の属性によって説明される必要があつていつまでも記述が終結しないからである。

このことからデータモデルの対象には識別キーを設けるべきものと識別キーを設けるべきでないものとの二種が存在することが分かる。この区別は対象をデータモデルに取り入れるときにされるものでその区別をはっきりさせないでモデリングを行うことはできない。[Koshafian 86] が挙げた上の例で、初め配偶者をファーストネームで記述したときはこれを識別キーを設けて記述しようとする対象とみなしていなかったから単なる属性として記述したのであって、後日配偶者が従業員になったときにはハッキリと識別キーで識別しなければならない対象と認識した訳でその記述方法が異なるのは当然であると考えられる。

では識別キーを設けるべき対象とそうではない対象とはどのように区別されるべきであろうか。識別キーを設けなくては本来の目的を達成できないケースとして複数人によって共有されるデータを扱う場合が考えられる。

たとえば識別キーの典型例として銀行の口座番号を考える。口座番号はそれを頼りに金銭の振込、引き下ろしが行われるからある口座番号を指定した場合それが何を指しているかが関係者に明確に分からなくてはいけない。口座番号の指定誤りは即口座指定の誤りである。このような対象を識別するためには識別キーの付与と識別キーと対応する対象の口座との対応について管理者の厳密な管理がなされていなければならない。関係者に明確に分かるためには人間の五感で識別できる何らかの「記号」が識別キーの値として与えられる必要がある。

まとめると、データの共有を目指すためには識別キーは

- (1) 厳密な管理のもとに
- (2) 人間によって識別可能な値を持たなければならない。

となる。

以上の考察は 2. (7) の批判とも関係する。オブジェクト指向データベースでは異なる事象の生起に異なるオブジェクト識別子を与えるとなつてはいるが、そもそも異なる生起か否かはデータ投入者の判断であると考えられる。時間を置いて窓口に現れた同一従業員を同一と判断するのは、データを管理している人間の意志を反映しているとき以外にはなく、オブジェクト指向データベースあるいは関係データベース等のシステムによって解決できる問題ではない。それよりもむしろシステムを運用している管理体制の問題である。

2. (5) の批判は例えば関係データベースのようなレコードを基本的な表現構造としているデータモデルを用いるときに制約であると多くの論者が考えている ([Kent 79]) 誤解ではないかと思われる。

例えば集合  $S = \{a, b, c\}$  を定義する方法に 2 種類が考えられる。一つは  $S = \{a, b, c\}$

というように  $S$  を陽に表現する方法。もう一つは

$a \in S$

$b \in S$

$c \in S$

のように  $S$  についてのいくつかの記述をすることによって間接的あるいは陰に表現する方法である。関係データベースで最後の例を表現するときには2個の属性を持つ3行の tuple により可能となる。

後者の方法を用いるならば二人の人が同一の子供の集合を有しているという表現は関係データモデルでも一つの単位として表現できる。 $S$  を削除したときなど  $a, b, c$  が  $S$  の要素であるという表現も同時に削除する必要があるが、これは単にデータの更新に伴う整合性条件の問題で、具体的な応用の表現として適切か不適切かという差はあるかもしれないが、関係データモデルが集合を1単位として扱うことができないということではない。適用分野によっては陽表現よりも陰表現の方が適している場合がある。

特に  $S$  について要素の列挙だけではなく別の記述（例えば  $S$  に適用される保険などの契約関係）を複数人が分かる形で行おうとすると  $S$  の陽表現だけでは不十分で、 $S$  に適当な識別キー値を与え、その識別キー値を参照する複数個の陰表現 (tuple) が必要となる。

つまり共有が必要な概念はそれが複雑な構造を持つものが持つまいがとにかく識別キー値で表現する必要がある。対象はそれを識別キー値で表現することにより（表現できたと約束することにより）モデルの共通参照ポイントとしていろいろな記述（例えば tuple）から参照されることにより陰関数的に定義される。識別キーで表現している対象は何でも良いわけで対象の表現能力が陽表現の範囲内であると考えする必要は全然ない。

2. (8) で指摘された点に関しては確かに現状の関係データモデルは改良の余地がある。モデリングの時には車の使用者を「個人」あるいは「部署」という区別をせずに汎化して単に「車使用者」とだけしておくことにより、後日「車使用者」として「個人」でも「部署」でもなく例えば [Kent 79] のいうように「支店」などが「車使用者」になったとしても設計を変更しなくて良いという利点がある。しかしその点が改良されたとしても最終的には実際に車を使用している「者」が誰なのか「車使用者」の具体的な実現値が「個人」なのか「部署」なのかの対応をとる必要があるのも明白である。この対応を考えるとときには「個人」あるいは「部署」ごとにその実現値がどのような識別キーによって表現されているかについて担当者による管理が依然として必要である。

批判点2. (4) は単に関係データベースにおいては等キーを利用しての結合操作によってのみ複数個の関係を関連づけるようになっていく仕様上の問題で、データの表現を識別キーで行ったとしても利用者にオブジェクト指向データベースで行っているような検索経路を見せることは実装の仕方でもどうにでもなる。

#### 4. 管理実体型概念の導入とその使い方

3. でデータモデルの対象には識別キーを必要とするものとそうではないものがあることが分かった。識別キーは3. で紹介した批判にもあるように通常の属性と混乱される可能性があるため、両者をはっきりと区別するために特別な概念を提案する。

識別キーを必要とする実体型を「管理実体型」と呼ぶことにする。

管理実体型は以下に述べるような条件を満たす。

- (1) 管理実体型の実現値は管理者によって厳密に対象と1対1に対応するすることが保証されなければならない。
- (2) 管理実体型の実現値は識別キー値としてデータの共有を許すために人間がそれに

ついて参照が可能な適当な値で表現される。

- (3) 管理実体型に対応する識別キーは人為的なもので、管理実体型の実現値を識別するという目的のためだけに設けられ、それに対しては生成、削除が可能であるが、変更はできない。
- (4) 識別キーそれ自身は他の意味を持たない。
- (5) 識別キー値はそれがどのように複雑な対象を表現していると仮定されようがそれを単位に扱われる。そのように見せかける整合性制約がデータモデルに備わっていると仮定される。
- (6) ある応用データモデルである時点において管理実体型が幾つあるかまた各々の実体実現値としてどのようなものが管理者によって登録されたかということが応用データモデル内で把握されていなくてはならない。

識別キー値は必ずしも原子的な記号である必要はない。たとえば複数の識別キー値の対から構成される場合もあるしもっと複雑な構造をとることもあり得る。ある応用データモデルである時点において把握されている管理実体型とその実現値は何等かの形で記憶されていなければならない。通常これはデータベースで扱う表あるいはファイルを用いてなされる。[情報資源スキーマ調査研究委員会 92] で検討中のJDMFではこのための表を実表として考えている。

## 5. おわりに

従来型のデータベースあるいはデータモデルに対してオブジェクト指向データベースからオブジェクト識別子についての批判点を紹介し、それを従来型のデータベースあるいはデータモデルでどのように考えるべきかを吟味した。

批判点は傾聴すべき点が多々あるが同時に現在「値」を基礎としてデータの共有を行っているシステムに対して若干考慮の足りない点があることを指摘した。従来型の識別キーの扱いでそのまま活きる部分と改良をすべき点を明らかにし、両者の統合として新たなモデル概念「管理実体型」を提案した。管理実体型概念は純粋なデータモデルの概念というよりは応用データモデルの運用と関連した概念である。

## 謝辞

[穂鷹 92] と関連して筑波データベースセミナーで増永良文氏(情報大)、清木 康氏(筑波大)から大変参考になるご意見を拝聴したことを感謝いたします。

## 参考文献

- [Atkinson 89] M. Atkinson et al.: The Object Oriented Database Systems Manifesto, Proc. Int. Conf. on Deductive and Object Oriented Databases, Elsevier, Amsterdam, 1990.
- [Copeland 84] G. Copeland and D. Maier: Making Smalltalk a Database System, SIGMOD '84, June, 1984.
- [情報資源スキーマ調査研究委員会 92] データモデル機能JDMF (v16.2) 日本規格協会, 1992.
- [穂鷹 92] 穂鷹良介: オブジェクト指向vs関係データモデル, 情報処理学会第44回(平成4年前期)全国大会

[戈 92] 戈 宝明: 個人的な通信, 1992.

[Kaehler 83] T. Kaehler and G. Krasner: LOOM-Large Object Oriented Memory for Smalltalk-80 Systems, in Smalltalk-80: Bits of History, Works of Advice, Addison-Wesley Pub. Co., Reading, Mass. 1983.

[Kent 78] W. Kent: Data and Reality, North-Holland, 1978.

[Kent 79] W. Kent: Limitations of Record-Based Information Models, ACM TODS, Vol.4, No.1, March 1979.

[Kent 91] W. Kent: The Breakdown of the Information Model in Multi-Database Systems, SIGMOD RECORD, Vol.20, No.4, December 1991.

[Khoshaflan 86] S.N. Khoshaflan and G.P. Copeland: Object Identity, Proc. Object Oriented Programming Systems, Languages and Applications-86, 1986 ACM.

[Maier 87] D. Maier and J. Stein: Development and Implementation of an Object-Oriented DBMS, in Research Directions in Object-Oriented DBMS, edited by B. Shriver and P. Wegner, MIT Press, 1987.