

## 解説

# データ管理参照モデル†

穂鷹 良介†



### 1. 歴史的経緯

データ管理参照モデル標準化の活動は1981年ISO(国際標準化機構)TC 97/SC 5(プログラミング言語)の下にWG 5(データベース管理システム)が設けられたときに始まる。

1985年にSC 21(開放型システム間相互接続のための情報検索ならびに管理)が新設されデータベース関連の標準化活動はすべてその中のWG 3(データベース)で扱われることとなり当プロジェクトもWG 3内に移り現在に至っている。

当初当プロジェクト名はデータ管理参照モデルではなくデータベース管理システム参照モデルといわれていたが、SC 21/WG 3にIRDS(情報資源辞書システム)のプロジェクトが新たに設けられたためデータベース管理システム参照モデルという名前はやや狭過ぎることになり、もうすこし一般的なデータ管理参照モデル(Reference Model of Data Management 以下本文では略してDMRMということにする)という名前に途中で変更になった。

参照モデルという考え方は市販ソフトウェアなどにあるものではなく非常に抽象的なものである。そのため同じ概念が提案されているような場合でも無用な摩擦が生じてしまうようなことがしばしばあり、言語の障壁を越えお互いに相手の立場を理解しかつ自分たちの考えを理解してもらえるようになるまでに長年月を必要とした。1987年ブダペストで開かれた専門家会議では基本的な概念、草案の内容について各国は合意に達したがまだはっきりとした文書の形で参照モデルができあがっているわけではない。合意内容に沿って各国の専門家から作業文書を構成するテキストが集められ、それをまとめて1988年3月のワシントン会議で作業文書を提案草案(draft proposal)に登録することを予定している。

### 2. 目的

現在情報システムのデータ管理に関連する分野では非常に多くの考え方が存在し、用語もメーカーによりユーザによりまちまちなものが用いられている。

ISOで現存する規格あるいは将来開発が予定されている規格の中にもデータ管理に関するものが複数個存在する。これらを放置しておくともISOが制定する規格案の中に世の中にみられるような混乱を再び招くことになりかねない。

標準化とは重大な考え方の相違がなければ類似したものを統合して一つの考え方に制約し、また用語についても標準的なものを数少なく認めることによって規格の使用に際して理解の効率を上げさらに誤解の生じる可能性を減らそうとするものである。ISOが制定する規格案の中で概念が混乱し用語が不統一になってしまったのではISOの存在意義が疑われることになる。

したがってDMRMは情報システム内に保持されるすべてのデータに関連する共通の概念、用語を定義し規格同士の調整、相互の関連を明らかにしようとする。したがって個々のデータ管理に関する規格はDMRMの定める概念が利用できる時には別の概念を新たに持ちだしてはいけなく、用語も同じものを用いることが必要とされる。逆にDMRMの定める概念、用語などに不適切なものがあればそれは改訂されねばならない。

しかしDMRMは決して規格案に沿う実装(implementation)方法を定めるものではない。単に規格に則った製品を利用者が利用するときに出会う無用な混乱を少なくしたいと意図するだけである。

### 3. 概念

#### 3.1 データレベル

一般にデータ処理であるデータを処理するときその処理に必要な意味または定義を事前に処理を行う主体は知らなければならない。この意味または定義はそれ

† Data Management Reference Model by Ryosuke HOTAKA  
(Institute of Socio-Economic Planning, University of Tsukuba).

† 筑波大学

基本レベル (レベル4)			基本スキーマ (3, 4)
IRD 定義レベル (レベル3)		IRD スキーマ (2, 3)	IRD 定義 (3, 3)
IRD レベル (レベル2)	応用 スキーマ (1, 2)	IRD (2, 2)	
応用レベル (レベル1)	応用 (1, 1)		

応用レベル対 IRD レベル IRD 定義レベ  
 (レベル対1) 対 (レベル対2) (レベル対3)

図-1 データレベル構造

自体としては抽象的なものであるが、それを代表するものを人為的に考えることは議論を明確にするのに役立つ。これをもとのデータのメタデータといい、この意味でメタデータはデータを「記述する」という。

データとメタデータは記述されるものと記述するものという関係をもつ。たとえば利用者が通常利用する応用データベース（データ）をそのスキーマ（メタデータ）が記述する。データ辞書あるいは情報資源辞書においてはさらに応用データベースのスキーマを記述する必要があり記述の段階がもう1段階増える。

ここにデータレベルという概念を導入する。データレベルには図-1のように番号を振ると上の関係が説明がしやすい。応用スキーマはデータレベル2の概念で図-1の箱(1, 1)のデータを記述し自分は箱(1, 2)に位置する。同じ概念は箱(2, 2)の一部のデータとして今度は箱(2, 3)のIRDスキーマによって記述されている。

一般に箱( $i, i$ )のデータは箱( $i, i+1$ )の概念によって記述され、レベル対 $i$ （一種のデータベース）の実現値として生成される。その一部は（全部ではない）たとえばデータベース定義プロセッサなどによって箱( $i-1, i$ )の概念となるように準備される<sup>9)</sup>。データ管理参照モデルでは一応データレベル4の概念はそれ以上高レベルの概念によって記述されないことを仮定している。データレベルの段数は4に制約される理論的な理由は特にないが、よくでてくるデータ管理参照モデルの応用例はこの範囲に収まるものが多かったので、あえてモデルを複雑にしなかっただけである。

レベル対は利用者がデータベースなどを使用するときのインタフェースに対応し、さまざまなサービスを提供される。

### 3.2 スキーマ分類概念

データレベル構造のスキーマもしくはスキーマを構成する要素を分類する概念を導入する。データモデルはスキーマによって具体的に表現されるから、これらの概念はデータモデルそれ自身を特徴づけるために用いられる。

情報資源辞書の内容は大変膨大なものとなる。情報資源の適切な分類概念が利用できればデータ管理者がそれを管理するときに注意を集中する範囲を限定でき効率的な管理をすることができる。将来、標準規格の情報資源辞書とインタフェースをもつ多様なソフトウェアの登場が予想されるが情報資源の適切な分類はこれらのソフトウェアの開発を促進するためにも効果的である。

#### (1) 対象種類

情報資源辞書は情報資源を記述するものであるが、記述される情報資源にどのような種類のものがあるかをあらかじめ認識しておき、情報資源辞書の内容をこの種類に従って分類することは大規模な情報資源辞書を取り扱うときに便利である。

データレベル構造図(図-1)のIRDまたはIRD定義の内容はそれが記述する対象(情報資源)によって次の3種類に分類される。

- (a) データを記述するもの
- (b) プロセスを記述するもの
- (c) 利用者を記述するもの

情報資源はそれぞれ細分化され細分化されたおのおのが特有な階層をなす。

たとえばデータは小は属性値から大はファイル、データベースに至るまで大きさに関する階層をもつ。あるいは具体概念から始まり抽象概念に至る汎化階層をもつ。

プロセスに分類される情報資源で最下位のものはプログラムあるいは人手による作業で途中で手続き、サブシステムなどがあり最上位にシステムが位置付けられる。

利用者とは情報システムを利用する大小さまざまな組織の総称でこれも個人レベルのものから最後は組織全体に至るものを考えることができる。

情報資源辞書がこのような情報資源の階層を反映して具体的に定められるときにはデータモデルの表現手

段を用いてなされることになる。

#### (2) データ表現ステージ

対象世界がモデリングされる場合、データモデルは大きく分けて論理的表現をとる段階と物理的表現をとる段階と二つの表現形態をとる。

論理的表現形態をとっているデータモデルを論理ステージにあるといい、物理的表現形態をとっているデータモデルを物理ステージにあるという。論理ステージにあるデータモデルは物理的表現手段に依存しない。利用者は論理ステージモデルを利用することにより物理データ独立性を達成することができる。

#### (3) スコープ

スキーマの記述が対象システムを全体的にとらえた大局的な見地に立つものか、あるいは応用分野に固有の局所的な見地に立つものかを区別する\*。

### 3.3 実現値分類概念

実現値を分類する概念かスキーマを分類する概念かは必ずしも明快に区別できないが、制御の対象となるものを分類する概念を、ここでは実現値分類概念として述べる。これには多様なものが考えられるが標準化に関係して重要と思われる分類概念は次のものがある。

#### (1) ライフサイクル

情報資源はその生誕から消滅に至るまでさまざまな状態をとる。情報資源の型ごとにその状態を分類し記述する概念をライフサイクルという。たとえば対象種類が entity type (実体型あるいは主体型) ごとにその実現値は生成、更新、検索、削除のライフサイクルをもつ。また図-1 の応用スキーマはまず IRD スキーマの実現値として生成され、単なるソーステキストとして IRD 内に置かれる。この時点ではまだ応用データを制御する「有効な」ライフサイクル上の状態に至っていない。その後システム管理者の指示によって「有効な」応用スキーマに状態を変える。

#### (2) 位置

情報資源の物理的な蓄積場所を表現する。ファイル

\* ANSI/SPARC の3層スキーマ構成を一種の参照モデルと考えると

- (a) 概念スキーマ
- (b) 外部スキーマ
- (c) 内部スキーマ

はそれぞれ

- (a) 論理ステージで大局スコープ
- (b) 論理ステージで局所スコープ
- (c) 物理ステージで大局もしくは局所スコープ

に位置付けられるスキーマであり、物理ステージで大局と局所のスコープのスキーマを ANSI/SPARC の参照モデルでは区別していない。物理ステージで局所スコープのスキーマは materialized view (具現化されたビュー) といわれているものに近い。

のありか、プログラムの記憶場所、利用者の居場所などを示す\*\*。

### 3.4 データモデル

対象世界の対象を代表する基礎となるデータモデリングの概念を実体 (または主体) という。

対象世界の対象の複雑さに対応してデータモデルではさまざまな実体の複合体を構成する手段が与えられるが、そのようにして構成される実体の複合体も再び実体と見なされる。複合実体はそれを構成する実体に分解されると考える。複合実体ではない実体を原子実体という。

ある実体の集合 (必ずしも有限とは限らない) があってそこに属する実体がある観点から同じ意味をもっていると考えられるとき、その意味は実体型によって代表されると考える。

ある実体型によって意味が代表されていると考えられる実体は、その実体型の実現値といわれる。

実体型は内包的あるいは外延的に指定される。

データベースなどで実体を扱うために実体の有限集合が選択され、保持される。この選択された実体の集合は実体型のファイル (またはポピュレーション) と呼ばれる。ファイルでどのような内容が保持されるかはファイルが期待される目的によって定められる。ファイルは関係データベースでは表といわれる。

ファイルは単に一つの実体型の実体実現値を集めたものであるが1個1個の実体をあらかじめ定められた実体型から定められた役にに応じて集められたものは列を構成する。列は時に文、アグリゲーション、述語、レコード、組、関連などといろいろの名前で呼ばれる。

ある実体型 E の実現値の実体 e が一つの列にある役をもってその構成に参加するとき、その役の意味は属性と呼ばれる別の概念 A によって代表される。E をその属性の定義域と呼び e をその列の属性 A の属性値という。

列はこれを再び実体とみなす。列は複合実体の一種である。

列型のファイルで属性値または属性値の列がファイルの中の列を一意に識別するとき、対応する属性または属性の集合をその列型の識別子またはキーと呼ぶ。

\*\* ライフサイクルはさまざまな情報資源のさまざまなライフサイクルの状態に応じて情報システム設計方法論あるいはソフトウェアパッケージの登場が予想され、その標準化もいづれ話題に登るものとおもわれる。位置は分散データベースを議論するときに考慮される分類概念である。

## 4. 記述方法

データ管理参照モデルの記述方法としては二つのものがある。一つは参考文献3)で提案されたデータフローダイアグラムを用いるもので、これは9)でも紹介された。その後オブジェクトモデリングという記述方法が提案され(5)または1)), テータ管理参照モデルの本文ではこの方法を用い、データフローダイアグラムは別途テクニカルレポートという参考資料でオブジェクトモデリングによる記述を検証する形で成果をまとめる方針が定められた。本文をなるべく短くして早期に成立させるためである。したがってここではオブジェクトモデリングを中心に述べる。

### 4.1 オブジェクト

オブジェクトとは基本的にはプロセス要素の抽象化されたもので、DMRMに直接関係あるインタフェースあるいはサービスの側面だけをモデル化し、これらのサービスがどのように実現されるかの詳細については述べないものである。

モデルはクライアントあるいはサーバとして対話を行うオブジェクトを用いて表現される。

クライアントの役を果たすオブジェクトはサーバの役を果たすオブジェクトに対してサービスを要求する。

各オブジェクトはそれが他のオブジェクトに対して提示する外部インタフェースによって定義される。ここではオブジェクトが提供できるサービスとサービスが適用されるデータの型が指定される。

各オブジェクトは対応するオブジェクト型の実現値である。オブジェクト型はその型の実現値全体に共通のサービスを決定する。ある特別のオブジェクト型のオブジェクトはそのサービスが適用されるデータの型を初めから固定している。しかし汎用のオブジェクト型ではオブジェクトの実現値がサービスの適用されるデータの定義をサービスの適用のつど別に要求する。

一つのオブジェクトは同時に複数のサーバオブジェクトのクライアントでありうるし、逆に一つのサーバオブジェクトは同時実行制御を行っているときなど複数のクライアントオブジェクトに同時にサービスを提供する。

アプリケーションプログラムはオブジェクトの一つとみることができるが、その定義方法がオブジェクトコードの形であるかソースコードであるか、実行に際

してそれがコンパイルされるかインタプリトされるかなどはデータ管理参照モデルの立場からは関心はない。

データ管理参照モデルの立場からみると利用者は通常サービスの要求をするがサービスの提供は行わないオブジェクトとして解釈される。

### 4.2 データ管理オブジェクトのためのデータ定義

データ定義はデータ管理の必須の概念であってオブジェクト表現と合わせてデータ管理全体を説明することになる。ここではまず単一の汎用データ管理サーバオブジェクトでサービス提供に際してデータ定義を要するケースに限定して両者の関係をみる。

最も基本的な要素はサーバオブジェクトのインタフェースで利用可能なデータ型の集まり全体の指定でオブジェクトの扱うデータ全体に世界を限定したとき、先に論理ステージで大局スコープのスキーマと述べたものに対応する。これをオブジェクトの全スキーマと呼ぶことにする。

サーバオブジェクトの提供するサービスをクライアントオブジェクトが利用するときには両者の間にサービス要求の表現と解釈に合意がなくてはならない。具体的には受け渡されるデータの定義が必要である。これをサブスキーマという。先に論理ステージで局所スコープのスキーマと述べたものに対応する。

クライアントの要求に正確に答えるためにはクライアントが内部的に仮定している特別なデータ変換のような物理的なデータ定義も必要である。これをクライアントの内部スキーマという\*。先に物理ステージで局所スコープのスキーマと述べたものに対応する。

特定のオブジェクトに対して明示的なデータ定義が必要なとき、それらは別のスキーマサーバオブジェクトを用いて指定され蓄積されなくてはならない。オブジェクトの実現値の一般的な性格は対応するオブジェクト型によって定められているが、その機能の完全な指定はスキーマ内のデータ定義と「接着 (binding)」することによりなされる。接着のタイミングは種々の環境によって異なりうる。クライアントオブジェクトがサービスを要求するとき接着がそのつど行われるか行われなくても環境によって異なりうる。

### 4.3 オブジェクトのグラフィック表現

オブジェクトに関して実現値を示す実現値表現とオブジェクト型を示す型表現の2種を考えることがで

\*ここで述べている内部スキーマの定義はANSI/SPARCのそれと異なっていることに注意されたい。

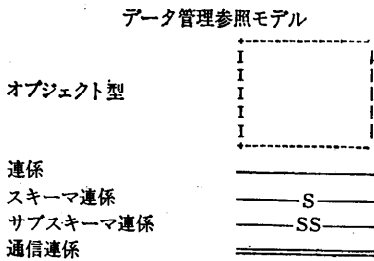


図-2 記号

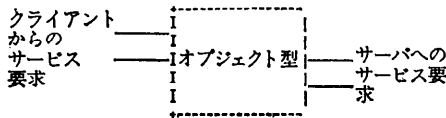


図-3 オブジェクト表現

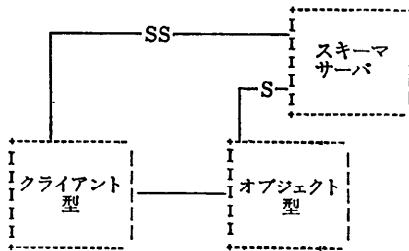


図-4 スキーマを用いるときの表現

きる。

実現値表現は精密であるが実現値ごとにオブジェクトを書かなくてはならず複雑になる。型表現は型に共通する一般的な形を示すため図は簡単になるが、あいまいさをともなう。簡単さのためにデータ管理参照モデルでは以下、型表現を用いる。

図-2 はグラフィック表現の基本記号を示す。なおこれらの基本記号は参考文献1)ではイメージが異なるが論理的には等価である。

図-3 はオブジェクト表現の一般型を示す。オブジェクト型の左側にそのオブジェクト型のクライアントオブジェクト型を配置し、そこからのサービス要求を関係で示す。オブジェクト型のIはインタフェースを示す。他のオブジェクト型から出ている関係はサーバオブジェクト型へのサービス要求を示す。

図-4 はオブジェクト型がスキーマを指定するケースを示す。スキーマ関係、サブスキーマ関係がオブジェクトのスキーマ、サブスキーマ利用の様子を表現している。スキーマの利用はサーバに対する通常のサー

ビス要求と異なって必ずしもサービスの要求ごとにデータが受け渡されるわけではなく、実装の方法によってはたとえば応用プログラムのコンパイル時に接着がなされるときだけデータの受渡しが行われる。

## 5. 参照モデルの記述

参照モデルはさまざまなデータ管理に共通した一般的な形を述べる抽象モデルと特殊なデータ管理を想定した詳細モデルの二通りの提示方法で示される。詳細モデルは想定する場面の数が多いので抽象モデルと若干の詳細モデルを示して参照モデルの概要を伝える。

### 5.1 抽象参照モデル

抽象参照モデルは以下の抽象オブジェクト型から構成される。

#### (1) ブロックサーバ

以下のサービスを提供するオブジェクト型である。

- データのブロックを蓄積検索する
- 内部データの複製を作る
- (回復のために) ジャーナルを保持する
- データを整合性のある状態に復帰させる

#### (2) データベースコントローラ

以下のサービスを提供するオブジェクト型である。

- 指定された利用者、サブスキーマと結合されたクライアントオブジェクトをもつセッションを確立する。
- 処理の意図 (たとえば共有しながらの更新) アクセス権利を認識する
- データベースレコードを選択、挿入、更新、削除する
- トランザクションを起動し、停止する
- データベースの中へあるいは中から大量データを転送する
- 回復手続きを起動する
- データベースの整合性をチェックする
- データベースを再編成する
- データベースを再構成する
- バックアップ手続きを選ぶ

#### (3) スキーマサーバ

以下のサービスを提供するオブジェクト型である。

- スキーマの一部を選択、挿入、更新、削除する
- 接着に必要とされる形でスキーマを提供する

#### (4) 利用者要求プロセッサ

以下のサービスを提供するオブジェクト型である。

- データベースをアクセスする要求を認知する

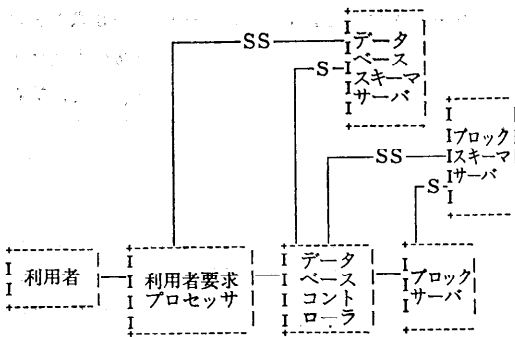


図-5 抽象データ管理参照モデル

- データベースをアクセスする要求を検査する
  - データベースをアクセスする要求を認可する
  - データベースをアクセスする要求を実行する
- (5) 利用者

データ管理サービスはなにも提供しないオブジェクト型である。この型の実現値はそれに許されるデータとサービスを決定し識別子をもつ。

抽象モデルは図-5 のように図示される。

利用者オブジェクト型の実現値は1個以上の利用者要求プロセッサに対してサービスを要求し、各プロセッサは1個以上のデータベースコントローラにサービスを要求する。データベースコントローラの実現値は複数のブロックサーバのサービスを利用するかもしれないが、それらのデータは全体として一つのデータベースを構成する。ブロックサーバの実現値は唯一のデータベースコントローラの実現値にサービスを提供し、データベースコントローラは複数の利用者要求プロセッサにサービスを提供する。利用者要求プロセッサは唯一の利用者オブジェクトだけにサービスを提供する。おのおののインタフェースにおいては利用可能なサービスならびにアクセス可能なデータはサービスを要求した利用者の識別の下に決定される。

データベースコントローラとブロックサーバの両者ともスキーマへの接着を必要とする。図-5 にはこのためにスキーマサーバの二つの役が、データベースコントローラとブロックサーバおのおののクライアントのサブスキーマの接着のための関係とともに示されている。おのおののスキーマサーバはスキーマの蓄積と検索のために再び他のサービスを要求するかもしれないが、図-5 が利用者のデータベースへのアクセスを示すためだけなので省略されている。

図-6 は図-5 の簡略表現である。省略された関係は明示的に書かなくても存在するものと仮定されて

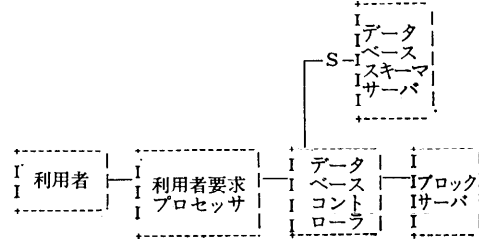


図-6 簡略抽象データ管理参照モデル

いる。

## 5.2 詳細参照モデル

上で述べた抽象参照モデルは、データ管理のどの場面でも適用可能な一般的な表現であると考えられる。実際にはもっと詳細な場面を識別しなくては標準化のために役立つものとはならない。具体的にはデータ管理の活動が生じるさまざまなケースが想定され、おのおのに別々の詳細モデルが導かれる。

場面の識別は主に図-1 のデータのレベル構造にあてはめてなされる。参照モデルの現段階では場面がまだデータモデルを用いるほど詳細化されていない。同様に分散処理の影響も考慮されていない。ともにデータ管理参照モデルの将来の課題である。

詳細参照モデルは以下のオブジェクト型の役を含む

- (1) ブロックサーバ役 以下の役が識別される：
  - (スキーマ) ブロックサーバ
  - (応用) ブロックサーバ
  - (辞書) ブロックサーバ
- (2) データベースコントローラ役 以下の役が識別される：
  - (応用) データベースコントローラ
  - (スキーマ) データベースコントローラ
  - (辞書) データベースコントローラ
- (3) スキーマサーバ役 以下の役が識別される：
  - (データベース) スキーマサーバ
  - (ブロック) スキーマサーバ
  - (応用) スキーマサーバ
  - (変換) スキーマサーバ
  - (データモデル) スキーマサーバ
- (4) 利用者要求プロセッサ役 以下の役が識別される：
  - 応用プログラム
  - 対話的質問プロセッサ
  - スキーマデータベースプロセッサ
  - 対話的辞書プロセッサ
  - スキーマプロセッサ

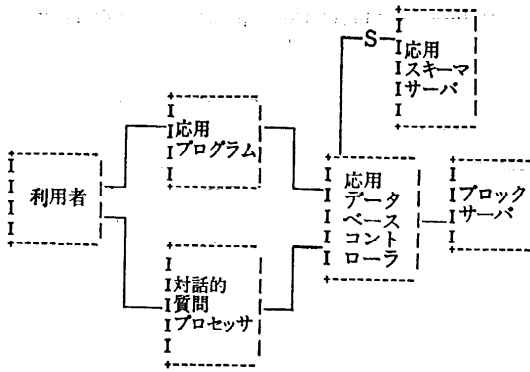


図-7 応用レベル対でのデータベースアクセス

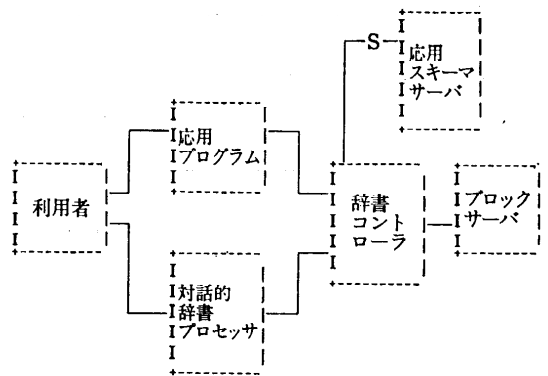


図-9 IRD レベル対でのデータベースアクセス

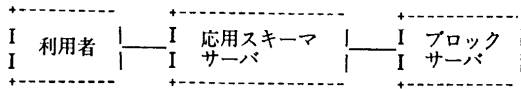


図-8 応用レベルデータのためのスキーマ定義

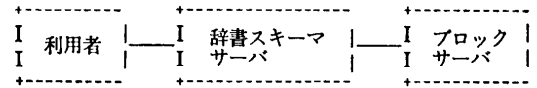


図-10 応用レベルデータのためのスキーマ定義

図-1 の各レベル対ごとにデータ管理のさまざまな場面が想定される。

応用レベル対では利用者オブジェクトのサービス要求に対して応用プログラム、対話的質問プロセッサの役のオブジェクトが利用者に成り代わって(応用)データベースコントローラに対してサービスを要求する場面がありこれは図-7 のように示される。

IRD レベル対では典型的には二つの場面が考えられる。一つはデータベースコントローラ型に対して特定のデータモデルのためのスキーマを定義する場合で、通常データベースのデータ定義がこれに当たる。このときデータ定義を司るアプリケーションスキーマサーバ(通常言い方ではデータベース定義プロセッサ)がレベル2データをどのように扱うかは事前に確定して、データ定義処理に当たって改めてさらにレベル3データからレベル2データに関する記述を補う必要はない。図で示せば図-8 のようになる。

もう一つはこのレベル対で用のあるデータがアプリケーションスキーマだけではなく、ほかに利用者に対して特別のサービスを提供するような仕組みを設けるようなときである。この場合、図-9 に示すように辞書コントローラにアクセスする形となる。

IRD 定義レベル対では、IRD スキーマを変更したりその内容を検索したりする。もしもレベル構造に基本レベル以上のレベルを認めないという制約があればここでのデータの利用の場面は図-10 のようなものだけであろう。

以上、詳細参照モデルとしてもごく簡単な場面につ

いて述べた。遠隔データベースアクセス、データモデルの変換、スキーマサーバの実現法などさらに詳細な参照モデルについては割愛する。分散データベース、データベース設計など、より精緻な参照モデルの今後の開拓が必要とされる。

謝辞 DMRM ならびに IRDS の標準化活動に関して日頃 SC 21/WG 3 でお世話になっている鈴木健司 (NTT)、溝口徹夫 (三菱電機)、堀内一 (口立製作所)、細谷明宏 (日本システミックス) の諸氏に紙面を借りて感謝の意を表したい。

### 参考文献

- 1) ISO/TC 97/SC 21/WG 3 DMRM Rapporteur Group: Information Processing Systems Data Management Systems Basic Reference Model, version 7 (Oct. 1987).
- 2) ISO/TC 97/SC 21/WG 3 DMRM Rapporteur Group: Information Processing Systems Data Management Systems Basic Reference Model, ISO/TC 97/SC 21/WG 3 N 150, version 5 (June 1986).
- 3) ISO/TC 97/SC 21/WG 3 DMRM Rapporteur Group: Technical Outline of the Model of Data Management, ISO/TC 97/SC 21/WG 3 N 241, version 5 (June 1986).
- 4) Reference Model of Data Management, Bud 03, ISO/TC 97/SC 21/WG 3 DMRM Rapporteur Group Budapest Meeting (Oct. 5, 1987).
- 5) Mike Newton: Contribution for the Reference Model of Data Management, ISO/TC 97/SC 21/WG 3 N 402 (Sep. 11, 1987).

- 6) Japanese member body: Dimensions of Various Parts of the Level Structure, ISO/TC 97/SC 21/WG 3 N 417 (Sep. 4, 1987).
- 7) Japanese member body: On Numbering Scheme of Concepts Appearing in Levels Structure, ISO/TC 97/SC 21/WG 3 N 416 (Sep. 4, 1987).
- 8) R. Hotaka: A Proposal on More Detailed Reference Model Concept, Bud 18, ISO/TC 97/SC 21/WG 3 DMRM Rapporteur Group Budapest Meeting (Oct. 16, 1987).
- 9) 穂鷹良介: データ管理参照モデル, 情報処理学会研究会報告データベースシステム 59-4 (1986).
- 10) 溝口徹夫: 情報資源辞書システム, 情報処理, 本特集号, pp. 215-224 (1988).

(昭和62年12月17日受付)