

## データベースオペレーティングシステム OPT-R の 設計目標とアーキテクチャ†

大久保 英嗣<sup>††</sup> 津田 孝夫<sup>††</sup>

本論文では、データベース処理にのみ目的を限定したオペレーティングシステム OPT-R (Relational database operating systems for One-Partition multi-user and multi-Transaction) の設計目標とアーキテクチャについて述べる。OPT-R の目的は、ホストオペレーティングシステム上のデータベース管理システムにおける性能の問題を、データベース処理に必要な機能のみをオペレーティングシステムの基本的機能として実現することにより、解決することである。OPT-R のおもな特徴としては、(1)ソフトウェアによるファイルのセグメンテーション、(2)複数ユーザによる複数トランザクション処理、(3)統一化されたユーザインタフェース、(4)階層的データディクショナリ/ディレクトリ、があげられる。本論文では、これらの特徴を実現するための OPT-R の具体的なアーキテクチャについて述べる。

### 1. はじめに

今日、データベース管理システム (以下 DBMS と記す) は、オペレーティングシステム (以下 OS と記す) 上のアプリケーションパッケージ、あるいはサブシステムとして構築されることが普通である。たとえば、UNIX 上に設計された INGRES<sup>1)</sup>、DOS/VSE 上に設計された SQL/DS<sup>2)</sup> などがある。これは、DBMS が、ファイルとその入出力の管理、ディスクとメモリ空間の管理、マルチタスキング等の機能のサービスをホスト OS より得ることが可能であることによる。しかし、この設計アプローチは、システムの効率の面で種々の問題を生じさせている<sup>3)-6)</sup>。

このような背景のもとで、われわれは、DBMS 側からの要求を満足させるようなオペレーティングシステム OPT-R を設計開発し、その有効性を検証したいと考えた。OPT-R は、従来のホスト OS 上の DBMS と異なり、データベース処理に目的を限定したデータベース専用の OS である。データモデルとしては関係モデルを採用し、Codd のいう完全な関係性 (fully relational)<sup>7)-9)</sup> を実現しようとするものである。OPT-R の特徴を以下に要約する。

#### (1) software segmented virtual memory<sup>10)</sup>

2次記憶媒体をすべて、ページ付けされた仮想空間に置き、プログラムおよびデータ等のシステム対象を区別することなく、セグメント単位で管理している。

#### (2) multi-user and multi-transaction

1パーティション (区画) でマルチタスク制御を行うことにより、複数ユーザおよび各ユーザによる複数トランザクション機構をサポートしている。

#### (3) two-level transaction scheduling

トランザクション単位の大域的なスケジューリング (global scheduling) を行う一方、1トランザクション内で局所的スケジューリング (local scheduling) を行うことにより、システムの処理多重度を高めている。

#### (4) unified user interface

ユーザインタフェースは、若干のコマンドと SQL 風のトランザクション記述用言語をサポートしている。トランザクション記述用言語に関しては、コンパイル手法をとっているが<sup>11),12)</sup>、コマンドを使用する際の手続きと別の手続きを踏まなくともよいように、一つの文が入力されるとただちにその構文解析を行っている。

#### (5) hierarchical transposed file

OPT-R におけるデータベースは、メモリ管理と一貫性をもたせるために、ページ単位に分割されアクセスされる。各ページには、属性の実現値をビットマップに変換しインバートしたビット列とインデックスが格納されている。さらに、ビットマップは階層化されており、高速アクセスが可能である。

#### (6) hierarchical DD/D

データベースの定義情報、一貫性および安全性制御のために、多層のカタログを構成している。

#### (7) modularity scheme

システムに適應性 (adaptability) をもたせるために、複数のサブシステム構成をとっている。

† The Design Objectives and Architecture of Database Operating Systems OPT-R by EIJI OKUBO and TAKAO TSUDA (Department of Information Science, Faculty of Engineering, Kyoto University).

†† 京都大学工学部情報工学科

以下、本論文では、OPT-R の設計目標と、上記の特徴を実現するためのソフトウェアアーキテクチャについてまとめる。

## 2. OPT-R の設計目標

われわれの目的は、計算機の構造からはなるべく独立した、データベース向きの OS を設計することにある。したがって、その性能もさることながら、システムの適応性、すなわちシステム変更の容易さに重点を置きながら設計を行うこととした。OPT-R の設計目標としては、要約すれば、

- (1) 性能 (performance)
- (2) 適応性 (adaptability)
- (3) 移行性 (portability)
- (4) ユーザ親和性 (user friendly interface)

の四つの向上があげられる。さらに、関係型の DBMS の特徴として、大量データあるいは高トラフィックの更新トランザクションに関して設計されているのではなく、むしろ preplanning が困難なアプリケーションに対して、その開発ツールを提供するといった傾向がある<sup>17)</sup>。そこでわれわれは、会話型指向の強い OS を実現するために、ジョブ制御言語や複雑なコマンド体系を排除し、真にエンドユーザをサポートするためのユーザインタフェースの実現に重点を置いている。また、現在、仮想メモリ方式の計算機が主流をなしているが、これにも対処するために、なるべくハードウェアのアーキテクチャから独立した設計思想が取れるように、ソフトウェアによるメモリ管理を行っている(4.1 節参照)。

OS の設計は、その対象とするユーザ属性によっても制約される。ユーザ分類のパラメータとしては、サポートするデータベースの容量、単位時間当りのトランザクション数、データベースアクセスの形態(定型的/非定型的)があげられる。われわれは、まず 100 MB 程度のデータベース容量、1 分間当りのトランザクション数が 10 内外で、しかもあまり定型的なアクセスパターンが存在しないユーザを対象としたシステムを構築することに目標を置いた。それは、現在の DBMS が、大容量ファイルを扱い、しかもオンラインリアルタイム処理指向となっており、定型的なアクセスパターンが存在するからで、それらユーザに関しては、考慮の対象外とした。すなわち、われわれは、小型計算機システム(主メモリ容量が数百 kB 程度のシステム)におけるマルチユーザ DBMS 構築を当面

の目標としている。現在のところ、これらユーザカテゴリに入る DBMS はあまりなく、実際、DBMS はその複雑さゆえに高価であり、小型計算機システムではあまりよく働かないと考えるからである。

## 3. OPT-R のソフトウェア構成

OPT-R のソフトウェアは、制御プログラム(CP: Control Program)とユーティリティプログラム(UP: Utility Program)に大別される。CP および UP は、**図 1**に示すモジュールより構成されている。

CP は、主メモリに常駐するモジュールと動的にローディングされるモジュールとからなり、システムの管理を行うプログラムであって、狭義のデータベースオペレーティングシステムである。CP の設計は、セマフォア(semaphore)を実現するためのシステムプリミティブである POST(事象の発生の通知)および WAIT(事象の発生を待つ)に基づいて行われている。すなわち、CP 内の各モジュールは、いくつかのタスクから構成され、それらタスク間の同期処理によって互いに通信を行いマルチプログラミングを可能としている。したがって、新しい機能を追加する場合は、一つのタスクを追加する形をとり、システム変更は他のモジュールに影響を及ぼさない。CP の基本的モジュールとしては、以下のものがある。

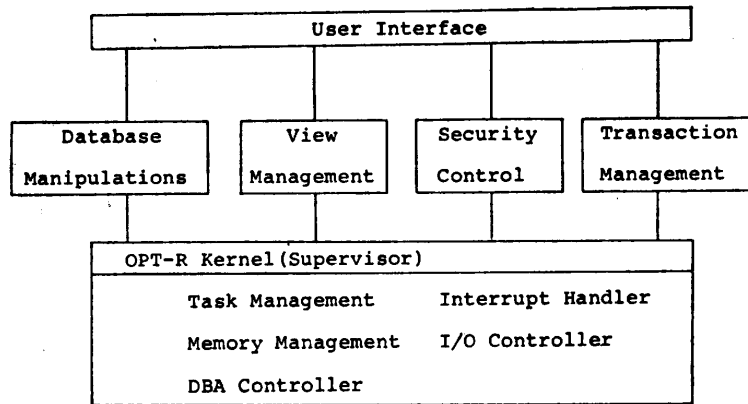
- (1) 各種データベース操作
- (2) ビュー管理
- (3) 安全性制御
- (4) トランザクション管理
- (5) ユーザインタフェース

UP は、データベースの生成、削除およびシステムジェネレーション等の機能を実現するための CP に従属的なプログラムで、計画的にオーバーレイされたり、動的にローディングされるモジュールから構成されている。OPT-R では、データディクショナリ機能の強化を目指しており、UP では、それら生成のための各モジュールが用意されている。

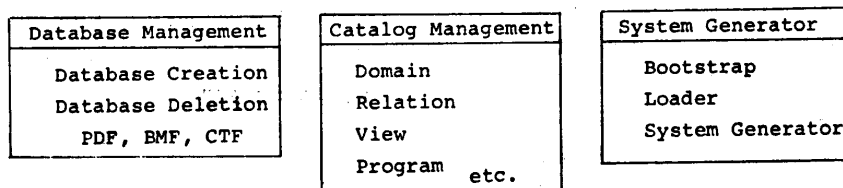
## 4. OPT-R Kernel

OPT-R の kernel は、

- (1) 割り込みハンドラ (interrupt handler)
- (2) 入出力制御 (I/O controller)
- (3) DBA 制御 (DBA controller)
- (4) メモリ管理 (memory management)
- (5) タスク管理 (task management)



(a) 制御プログラム



(b) ユーティリティプログラム

図 1 OPT-R のソフトウェア構成  
Fig. 1 Software configuration of the OPT-R.

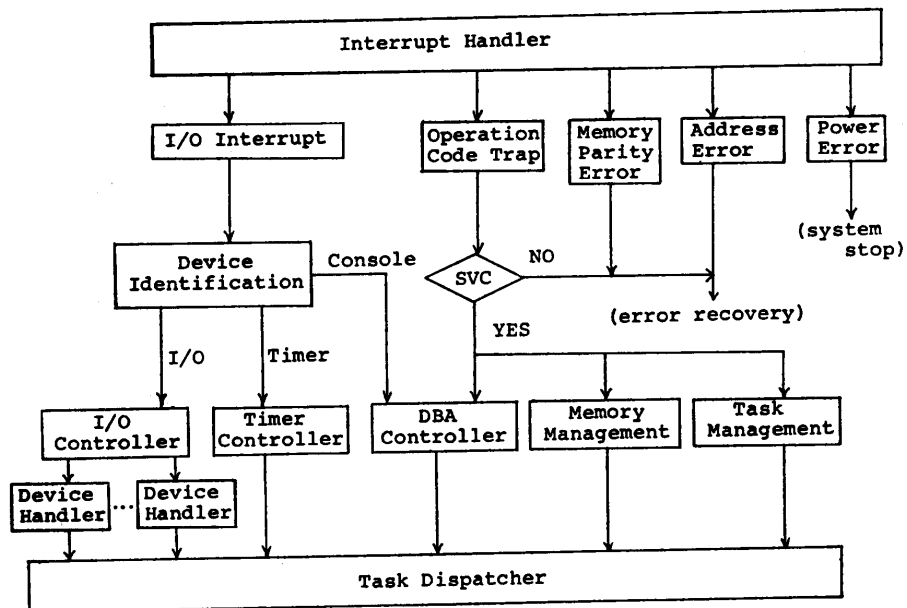


図 2 Kernel モジュールの関連  
Fig. 2 Relationships of Kernel modules.

よりなる。各モジュールの関連を図2に示す。

割込みハンドラは、システムに発生する種々の割込みを受け付け、割込み要因の解析および各処理モジュールへの分岐制御を行う。割込み要因としては、入出力割込み、オペレーションコードトラップ、メモリパリティエラー、アドレスエラー、電源異常がある。このうち、OPT-Rでは、オペコードトラップを利用してスーパーバイザ処理を行っている。

入出力制御は、入出力装置対応にあるデバイスハンドラとそれを管理する制御モジュールおよびタイマ制御から構成される。入出力装置は、PDN (Physical Device Number) と呼ばれる2バイトのコードによって識別される。したがって、新たな装置を接続する場合、それに対するPDNとデバイスハンドラを追加するだけでよく、システムの適応性を向上させている。

DBA制御は、DBAとOPT-Rの間の通信を制御する。DBAは、DBA用のコンソール装置を介して、システムに対して特権的に指令を与えることができる(各権限については、8章参照)。

以下、OPT-R kernelにおける特徴的機能であるメモリ管理およびタスク管理について述べる。詳細は、本論文以降の論文に譲る<sup>22), 23)</sup>。

#### 4.1 メモリ管理

OPT-Rの主メモリは、システムパーティションとユーザパーティションと呼ばれる二つのパーティションに大きく分かれる(OPT-Rの主メモリ構成を図3に示す)。メモリ管理は、システムパーティション内に常駐し、ユーザパーティションおよび2次記憶上の領域を管理している。

ユーザパーティションは、以下に述べる四つのシステムテーブルを用いて管理される。

System partition (80 kB)	System table area
	System queue pool
	CP (Control Program)
	System overlay area
	System stack
User partition (176 kB)	• UP (Utility Program)
	• Catalogs
	• Databases
	• Work areas
	• User programs

図3 主メモリ構成

Fig. 3 Main memory layout.

- (1) 主メモリ管理テーブル (MMCT)
- (2) 内部ページ管理テーブル (IPCT)
- (3) 外部ページ管理テーブル (EPCT)
- (4) ストレージマップ (SMAP)

MMCTは、EPCTとIPCTに関する情報および常駐カタログ(7章参照)に関する情報を保持している。IPCTは、ユーザパーティションのページ(512 B/ページ)対応の一つ存在し、当該ページの使用状況を管理するために使用される。EPCTは、主メモリに存在するセグメント(プログラムおよびデータ等)対応の一つ割り当てられ、ページング処理に使用される。2次記憶上の領域は、SMAPと呼ばれるビットマップによって管理される。2次記憶上の領域は、記録媒体を意識せずに、PBN (Physical Block Number) と呼ばれる4Bのコードによって番号づけられており、SMAPの各ビットは、このPBN対応に存在し、ビット'1'が使用中を、ビット'0'が空きを表している。

#### 4.2 タスク管理

OPT-Rでは、ユーザ管理、トランザクション処理、データ操作の各処理単位に一つのタスクを割り当て、1パーティション(システムパーティション)でマルチタスク制御を行っている。OPT-Rにおけるタスクには、それぞれ優先順位(priority)が与えられており、実行待ち状態になると各タスクは、優先順位に応じてCPUが割り当てられる。OPT-Rのタスクは、システム管理用のタスクとユーザ要求管理用のタスクに大別される。これらのタスクは、上述したような従来のOSで採用されているスケジューリングの対象となるが、ユーザ要求管理用のタスクは、さらに、6章で述べるスケジューリングの対象となる。OPT-Rでは、前者をglobal scheduling、後者をlocal schedulingと呼んでいる。これら二つのスケジューリングにより、従来のDBMSにおけるような頻繁に発生する2次記憶への入出力要求に起因するタスクスイッチオーバーヘッドが問題にならないよう工夫している。

### 5. ユーザインタフェース

OPT-Rでは、ユーザ管理をセッション単位で行い、ユーザ用に若干のコマンドを用意している。また、トランザクションの処理は、通常のコマンド処理のためのreadyモードから、トランザクション記述用言語の入力を可能とするtransactionモードへ移行して行っている。readyモードでは、表1(a)に示すコマンドが使用可能である。ユーザの入力は、80文字

表 1 ユーザインタフェースのコマンド  
Table 1 Commands of user interface.

(a) ready モードのコマンド (a) Commands for ready mode		
LOGON		セッションを開始する
LOGOFF		セッションを終了する
BEGIN		トランザクション処理を開始する
FETCH		トランザクションの結果を得る
CANCEL		トランザクション処理の中止を要求する
DISPLAY		カタログ等の各種情報を表示する
HELP		コマンドに関する情報を表示する
CHANGE		名称, パスワード等の変更を行う
(b) transaction モードのコマンド (b) Commands for transaction mode		
	DECLARE	ドメインの定義
	CREATE	関係の創成
データベ ス定義文	EXPAND	関係に属性を追加
	DEFINE	ビューの定義
	DROP	ドメイン, 関係, ビュー等の消去
	COMMENT	関係, 属性等へコメントを付加
制 御 文	ASSERT	一貫性制約の定義 (assertion)
	TRIGGER	一貫性制約の定義 (trigger)
	GRANT	権限の付与
	REVOKE	権限の削除
データベ ス操作文	INSERT	タブルの追加
	DELETE	タブルの削除
	UPDATE	タブルの更新
検 索 文	SELECT	データベースの検索
	END	トランザクション処理を終了する
	X	ready モードのコマンドを実行する

に制限されており即時処理される。transaction モードでは、表 1 (b) のような SQL 風のトランザクション記述用言語およびコマンドを使用することができる。ユーザの入力は、1 トランザクションにつき 1,024 文字 (2 ページ分) までの入力が可能で、システムでは END コマンドまでを一つのトランザクションとして処理する (トランザクションの概念については、たとえば文献 18) 参照)。

システムは、END コマンドが投入されると、ユーザに対してユニークなトランザクション番号を返却する。ユーザは、そのトランザクションの終了を待たずに、次のトランザクションの処理を要求することができる。トランザクションの処理は、ユーザから離れて (すなわち別のタスクとして) 独自に進められる。処理が終了すると、その旨を端末へ表示する。ユーザは、その後の任意の時点で、FETCH コマンドによって結果を得ることができる。また、すでに生成したトランザクション要求の取消しは、CANCEL コマンドで可能である。以上のように、BEGIN→トランザク

ション入力→END→FETCH/CANCEL といった簡単な手続きにより、データベースをアクセスすることが可能となっている。

## 6. トランザクション管理

トランザクション管理は、前章でも述べたように、BEGIN コマンドから END コマンドまでを一つのトランザクションとして処理する。トランザクションは、一貫性 (integrity) 保証の単位である。すなわち、その前と後では、データの一貫性を保証しなければならない。システムは、トランザクションを以下に述べるような流れで処理する。

### (1) トランザクション開始処理

BEGIN コマンドの処理を行う。すなわち、トランザクション用の作業領域を確保し、トランザクション番号等の各種情報を設定する。

### (2) 構文解析

ユーザの入力したトランザクション記述用言語を文単位に構文解析する。文法的に誤りがあれば再入力を促し、正しければ、それを中間コードに変換してトランザクション開始処理で確保した作業領域へ格納する。

### (3) 意味解析

トランザクション中にビューが出現していれば、中間コードの形でビューカタログに格納されているビューがマージされる (ビュー管理の方式に関しては、とくに新しいことがないので、これ以上触れない)。生成された中間言語系列により、演算順序の変更によるユーザ要求の最適化を行う。その後、それをもとに基本的な関係操作の木を生成する (OPT-R では、関係代数を基本としている<sup>13)</sup>)。

### (4) 局所的スケジューリング

関係操作の木を並行処理可能なタスクの識別アルゴリズム<sup>14)</sup>によって、データベース操作モジュールへの POST-WAIT 系列へ変換する。

### (5) タスク生成

POST-WAIT 系列を一つのタスクとして生成し、大域的スケジューリングの対象とする。

### (6) データベース操作

データベース操作の各モジュールは、(5)で生成されたタスクからのパラメータをもとに、データベースへのアクセス権限をチェックし、さらに、演算単位への各資源に対するロッキングを行い、データベースへのアクセスおよび関係操作を実行して、トランザクシ

ンの結果を生成する。

(7) 一貫性チェック

トランザクション単位の一貫性 (7.2 節参照) をチェックする。一貫性が損われている場合は、もとの状態にバックアウトする。

7. データベースおよびデータディクショナリ

本章では、データベースの物理構造として採用している HTF (Hierarchical Transposed File) と、データの一貫性および安全性制御に使用される各種のカタログについて説明する。

7.1 データベースの物理構造

論理的には表の形で与えられる関係を物理構造に写像する方法として、各列 (属性) 単位に格納する方法

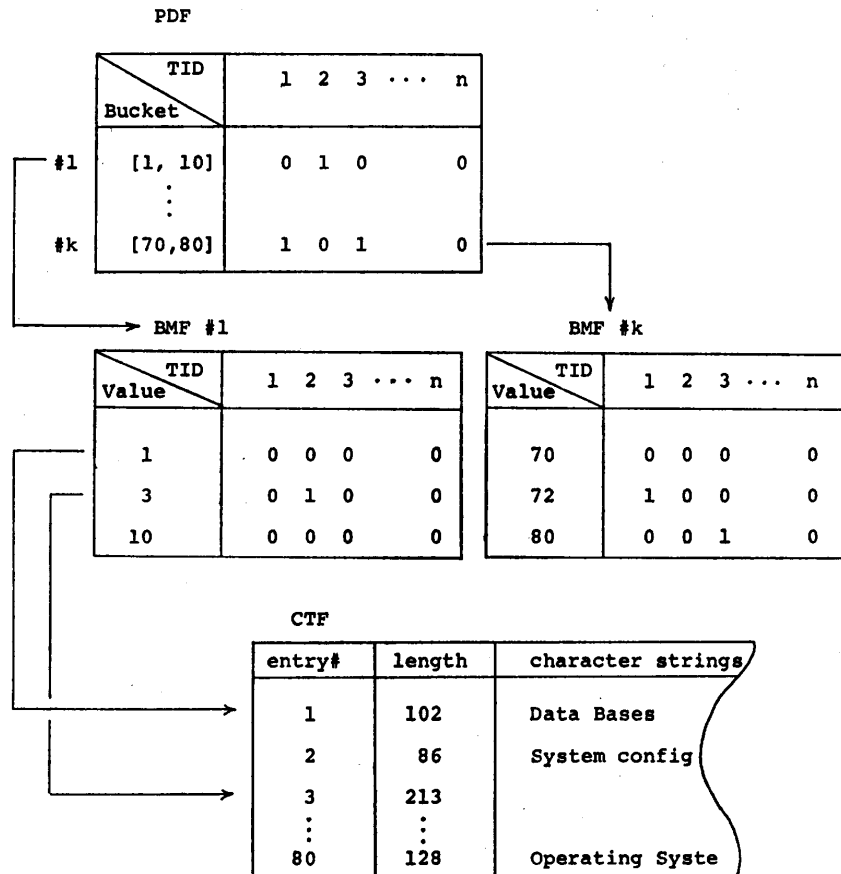
があり、これは、転置型ファイル (transposed file) と呼ばれている<sup>19),20)</sup>。OPT-R では転置型ファイルを採用しているが、転置型ファイルに対するアクセスの高速化とデータベース操作の高速化のために、ビットマップを階層化した階層転置型ファイル (HTF) を実現している。HTF は、以下の三つのファイルより構成される (図 4 参照)。

(1) 分布ファイル (PDF)

属性値の分布状態に着目したビットマップ形式のファイルである。

(2) 2 進行列ファイル (BMF)

行にタプル識別子 (TID), 列に属性値の異なった値をコード化した索引をもつビットマップ行列のファイルである。ただし、データが整数および実数型の場合は、値そのものを索引としている。



PDF: Piecewise Distribution File  
 BMF: Binary Matrix File  
 CTF: Compressed fully Transposed File

図 4 OPT-R におけるデータベースの論理構造  
 Fig. 4 Logical structure of databases in the OPT-R.

## (3) 圧縮完全転置型ファイル (CTF)

実際の属性値の異なった値だけを格納したファイルである。

本ファイル構造は、たんにデータの実現値のみを格納する転置型ファイルに比較して、ストレージコストが非常に大きくなる (PDF および BMF のビット列は、タプル数を  $n$  とすると、 $O(n^2)$  ビットのストレージ量を必要とする)。したがって、PDF および BMF を DIS (Double Indexing Scheme) と呼ばれる方法<sup>15)</sup> によって圧縮している。本圧縮法は、疎行列 (sparse matrix) の圧縮格納法の一つである。DIS は、行ベクトル、列ベクトル、値ベクトルの三つのベクトルから構成される (詳細は省略する)。このように変換された HTF は、それが所属する属性のカタログ内にある 2 次記憶上のアドレス (4.1 節で述べた PBN) によってアクセスされる。

CTF は、文字列等属性値が長い場合に必要となるものであり、PDF は BMF の副次索引である。したがって、HTF の階層構造として以下の 4 通りが考えられる。

- (1) PDF-BMF-CTF
- (2) BMF-CTF
- (3) PDF-BMF
- (4) BMF

更新処理に関して、PDF および BMF をアクセスする回数は、検索処理に比較すると多くなることが予想される。したがって、更新があまり行われなような属性 (たとえば主キーのような属性) には、PDF を

付加する。PDF の目的は、元来、一様分布データ (異なる値の個数がタプル数にほぼ等しいという意味) に関するアクセス回数を減少させる目的で考案されたものである。この考え方は妥当であると思われる。

## 7.2 データディクショナリ/ディレクトリ

本節では、OS 本来の機能で用いられるシステムテーブル以外に、データベース管理機能を実現するために当たって必要となるカタログの集合、すなわちデータディクショナリについて説明する。図 5 に、OPT-R におけるカタログの階層を示す。各マスタカタログのうち、関係、ビュー、ドメインマスタカタログは、システム初期化時にローディングされ、以後主メモリに常駐する。

ドメインマスタカタログは、各ドメイン対応にドメイン名とドメインカタログへのポインタ (PBN) を保持しており、ドメイン一貫性の制御を行うものである。関係マスタカタログは、各関係対応に関係名と関係カタログへのポインタを保持している。さらに、関係一貫性の制御を行うためのアサーションおよびトリガカタログへのポインタと、権限付与を制御するためのグラントカタログへのポインタも保持している。ビューマスタカタログは、各ビュー対応にビュー名とビューカタログおよびグラントカタログへのポインタを保持している。

これらカタログと一貫性制御との関連を以下に要約する (なお、一貫性制御に関しては、文献 16) を参考にした)。

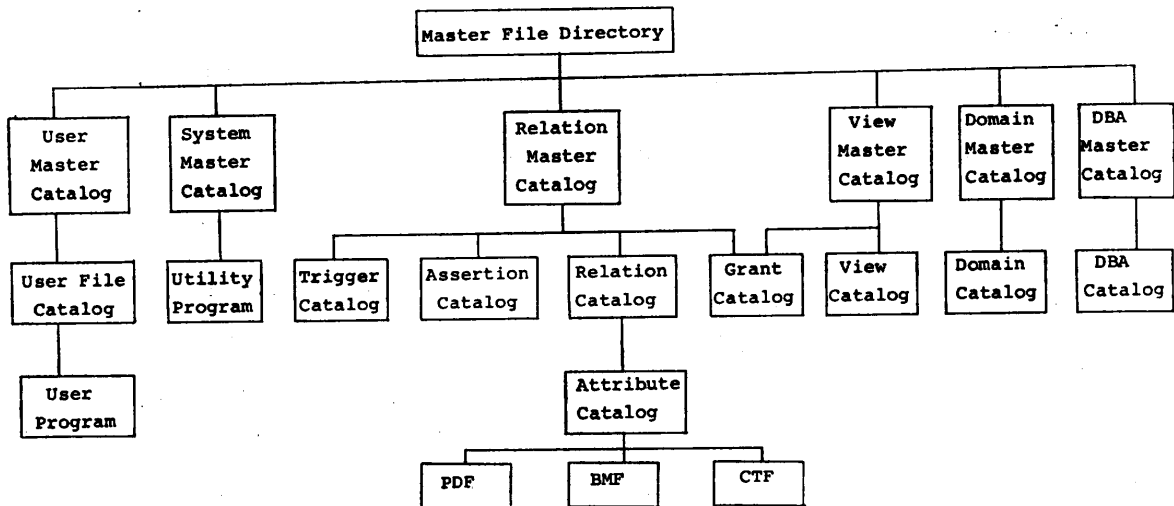


図 5 OPT-R におけるカタログの階層

Fig. 5 Catalog hierarchy in the OPT-R.

### (1) ドメイン一貫性の制御

トランザクション記述用言語の DECLARE 文によって定義されるドメインのデータ型 (バージョン1では、整数型, 実数型および文字型しかサポートしていない) および取りうる値の範囲と長さが, ドメインカタログに格納される. 各データベース操作モジュールは, 演算に先立って, その対象となるデータ間のドメインの競合性 (compatibility) を, このドメインカタログをもとにチェックする.

### (2) 関係一貫性の制御

関係一貫性に関する制約は, ASSERT 文と TRIGGER 文により定義され, おのおのアサーション, トリガカタログに格納される. ASSERT 文は, 関係に対する変更(タプルの挿入, 削除および更新)を行う前の制約を定義するものである. TRIGGER 文は, 関係に対する実際の参照および変更を契機として, その動作が満足すべき制約を定義するものである. ASSERT 文および TRIGGER 文中の制約には, 当該関係内の属性に関する論理式が書ける. とくに, OPT-R では, 変更前の属性の値と変更後の属性の値に関する式を記述することができる. しかし, 二つ以上の関係にまたがる制約 (参照一貫性制約<sup>16)</sup>) は, バージョン1ではサポートされていない.

以上に述べた各カタログに対する参照, 新エントリの追加, 更新, 削除は, すべて PBN を介してメモリ管理が行う. したがって, 各モジュールは, アクセスしたいカタログ名を指定して, メモリ管理を呼び出せばよいだけである.

## 8. 安全性制御

OPT-R では, 権限を付与された適切なユーザだけに, データおよびプログラムに対する操作を許す機構をサポートしている. これは, capability-based addressing の概念<sup>21)</sup>に基づくもので, オブジェクトへのアクセスは必ず C-list (Capability-list) を介して行われる. OPT-R の C-list は, 各カタログに分散して格納されており, 権限チェックのオーバーヘッドが小さくなるよう工夫している. さらに, 権限付与に関する権限もその機構の中に取り入れている. これは, グラントカタログのエントリで実現される.

OPT-R では, さらに, 特定のユーザの集合に権限を付与できるように, グループ単位の管理を可能にしている. OPT-R のユーザは, UIC と呼ばれる内部コードによって識別される. UIC は, グループ識別

コード GIC およびメンバー識別コード MIC の各 2B から構成される. この UIC によって, 各カタログ内のエントリの一つである所有者の UIC と比較して, 以下のようにユーザを区分化する.

- (1) DBA: GIC = (0100)<sub>16</sub> ~ (01FF)<sub>16</sub>
- (2) OWNER: UIC が一致
- (3) MEMBER: GIC が一致
- (4) STRANGER: 上記以外

ユーザが区分化されると, これも各カタログ内のエントリの一つであるアクセス権限エントリによって, アクセス権限 (read, write, delete, execute, grant, revoke, create) が決定される.

## 9. おわりに

現在, OPT-R バージョン1は, HITAC E-600/3 (主メモリ容量 256 kB, ファイル容量 40 MB) 上に構築されており, システム記述用言語としては, アセンブリ言語と PASCAL を使用している.

当初の目的であった, システムの適応性, 移行性およびユーザ親和性は, ある程度達成されたと考えている. しかし, 第1の目標である性能向上に関しては, 詳細な性能解析を行っていないので, 現段階では, 定量的なことを述べることはできない. これに関しては, 他論文で発表する予定である.

最後に, バージョン1で実現されていないが, 今後必要となる機能を列挙しておく.

- (1) システムの多重度 (タスク数) に関する管理方式.
- (2) トランザクションのネスト処理.
- (3) 会計および統計情報を取得するような長期間実行されるトランザクションの管理機能.
- (4) 二つ以上の関係にまたがる一貫性制約 (参照一貫性<sup>16)</sup>) のチェック処理.
- (5) 各システム対象 (関係, 属性等) の値の範囲等による詳細なレベルでの安全制御.

## 参考文献

- 1) Stonebraker, M. et al.: The Design and Implementation of INGRES, *ACM TODS*, Vol. 1, No. 3, pp. 189-222 (1976).
- 2) Chamberlin, D. D. et al.: A History of System R and SQL/Data System, *Proc. Very Large Data Bases*, pp. 456-464 (1981).
- 3) Stonebraker, M.: Operating System Support for Database Management, *Comm. ACM*, Vol. 24, No. 7, pp. 412-418 (1981).



- 4) Stonebraker, M.: Retrospection on a Database System, *ACM TODS*, Vol. 5, No. 2, pp. 225-240 (1980).
- 5) Spooner, D. et al.: The Logical Object Model for Data Base Operating Systems, *COMPSAC 80*, pp. 769-775 (1980).
- 6) Gray, J.: Note on Data Base Operating Systems, in Goods, G. and Hartmanis, J. (eds.), *Lecture Notes in Computer Science*, pp. 393-481, Springer-Verlag, New York (1978).
- 7) Codd, E. F.: Extending the Database Relational Model to Capture More Meaning, *ACM TODS*, Vol. 4, No. 4, pp. 397-434 (1979).
- 8) Codd, E. F.: Relational Databases: A Practical Foundation for Productivity, The 1981 ACM Turing Award Lecture, *Comm. ACM*, Vol. 25, No. 2, pp. 109-117 (1982).
- 9) Codd, E. F.: The Capabilities of Relational Database Management Systems, IBM Research Report, RJ 3132 (1981).
- 10) Baxter, A. Q. and Hart, J. M.: Software Segmented Virtual Memory, *Softw. Pract. Exper.*, Vol. 12, No. 2, pp. 185-194 (1982).
- 11) Blasgen, M. W. et al.: System R: An Architectural Overview, *IBM Syst. J.*, Vol. 20, No. 1, pp. 41-62 (1981).
- 12) Astrahan, M. M. et al.: System R: Relational Approach to Database Management, *ACM TODS*, Vol. 1, No. 2, pp. 97-137 (1976).
- 13) Date, C. J.: *An Introduction to Database Systems*, 2nd ed., Addison-Wesley, Reading (1977).
- 14) Ramamoorthy C. V. et al.: Scheduling Parallel Processable Tasks for a Uniprocessor, *IEEE Trans. Comput.*, Vol. C-25, No. 5, pp. 485-495 (1976).
- 15) Ashany, R.: Application of Sparse Matrix Technique to Search, Retrieval, Classification and Relationship Analysis in Large Data Base Systems—SPARCOM, *Proc. Very Large Data Bases*, pp. 499-516 (1978).
- 16) Date, C. J.: *An Introduction to Database Systems*, Vol. II, Addison-Wesley, Reading (1983).
- 17) Dieckmann, E. M.: Three Relational DBMS, *Datamation*, Vol. 27, No. 10, pp. 137-139 (1981).
- 18) Gray, J.: The Transaction Concept: Virtues and Limitations, *Proc. Very Large Data Bases*, pp. 145-154 (1981).
- 19) Batry, D. S.: On Searching Transposed Files, *ACM TODS*, Vol. 4, No. 4, pp. 531-544 (1979).
- 20) Svensson, P.: On Search Performance for Conjunctive Queries in Compressed, Fully Transposed Ordered Files, *Proc. Very Large Data Bases*, pp. 155-163 (1979).
- 21) Fabry, R.: Capability-based Addressing, *Comm. ACM*, Vol. 17, No. 7, pp. 403-412 (1974).
- 22) 大久保英嗣, 津田孝夫: データベースオペレーティングシステム OPT-R のタスク管理とトランザクションのスケジューリング技法, 情報処理学会論文誌, Vol. 25, No. 4, pp. 544-551 (1984).
- 23) 大久保英嗣, 津田孝夫: データベースオペレーティングシステム OPT-R のメモリ管理方式, 情報処理学会論文誌, Vol. 25, No. 4, pp. 552-559 (1984).

(昭和 58 年 5 月 20 日受付)

(昭和 59 年 1 月 17 日採録)