

VOS0/ES, DPOS リレーショナル形データベースの開発

Development of Relational Database Facility for VOS0/ES and DPOS

蛭川 壮一* Sôichi Hirukawa
柴 勉** Tsutomu Shiba

小形計算機分野では、処理の分散化やOA化の流れに伴って、多種多量のデータを取り扱うようになってきた。このため、データベースを小形計算機システム上で構築し、ファイルとプログラムの有効活用を図る目的でRDBFの開発を行なった。本論文では、リレーショナルデータモデルの考えに基づいたRDBFでのデータ構造と、その考え方及び仕組みについて述べるとともに、その特長をユーザーの使い方によって解説する。

1 緒言

処理の分散化やOA (Office Automation) 化が近年盛んになってきている。このため、小形計算機分野でも取り扱うデータの種別及び量の増加が著しい。このため、簡単にファイル設計ができ、効率良く外部記憶装置を使用できるファイルシステムが要求されてきた。リレーショナルデータモデルの考えに基づくデータベースが、この要求を満たすものとして最近特に注目されている。

本論文では、小形計算機システム(VOS0/ES:Virtual storage Operating System 0/Extended System, DPOS:Distributed data Processing System)用として開発したリレーショナル形データベースの構造及び特長について述べる。

2 RDBFの開発

小形計算機では、取り扱うデータを部門ごとにファイルとして管理、運用している。このため、各部門間で同じようなデータを持ち、ファイル容量の増大を招いている。また、プログラムがデータ構造に依存しているため、データ構造の変更に対してプログラムも修正しなければならない。RDBF (Relational Database Facility)は小形計算機向けに、これらデータを一元管理することで、ファイルとプログラムの有効活用を図る目的で開発を行なった。

RDBFでは、ファイルの論理的な構造を2次元のテーブル形式(表形式)で表現し、それぞれのユーザー(プログラム)が必要とするとき利用したい形式で、データを取り出すことができる。リレーショナル形データベースファイルを利用すれば、使用する部門の業務内容に依存しないファイルを構築できる。例えば、氏名、経歴、給与から成るファイルを作成し、人事プログラムでは氏名、経歴を、給与計算プログラムでは氏名、給与を利用するという定義を行なうことで、それぞれ同一ファイルから独立した業務を行なうことができる。

RDBFの開発目的を以下に述べる。

- (1) 単純で分かりやすい表形式のデータ構造である。
- (2) データの定義とプログラムが独立している。
- (3) 複数ファイルから必要なデータを、複数ファイルであることを意識せずに取り出せる(論理ファイルの概念の導入)。
- (4) データへのアクセスを従来のファイルの概念と同様にし、コード単位で行なえる。本来のリレーショナルデータベース

では、関連データの集合に対してが一つのアクセス単位となっているが、実用上の観点からRDBFではレコード単位とする。

- (5) COBOLなどの言語でも簡単にプログラムが作成できる。

3 RDBFの概要

RDBFで扱うデータベースは、テーブル形式で表わされたファイルから構成されており、その構造は多種多様な処理の要求に応じられるようになっている。データベースのファイルには、論理的な仕組みから見たファイルと実際のデータなどの格納方式を示す物理的な仕組みから見たファイルがある。図1にデータベースの構成を示す。

3.1 データベースの論理的仕組み

データベースのファイルを論理的な仕組みから見ると、物

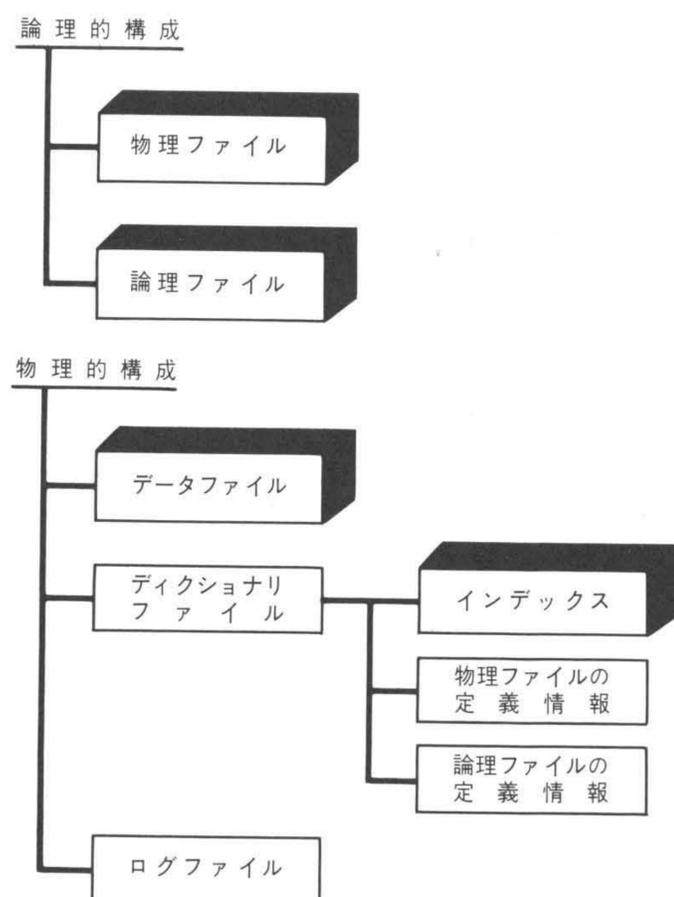


図1 データベースの構成 データベースの構成には、物理的仕組みから見たものと論理的仕組みから見たものがある。

* 日立製作所ソフトウェア工場 ** 日立ソフトウェアエンジニアリング株式会社

理ファイルと論理ファイルの2種類に分けられる。物理ファイルは、現実にデータを格納しているファイルであり、論理ファイルは、物理ファイルに格納されているデータをその物理的表現とは異なるレコード様式で処理するために、物理ファイルから特定のフィールドを取り出したり、フィールドの並び順を変更したり、あるいは複数の物理ファイルを結合したりして構成した仮想的なファイルである。

(1) 物理ファイルの構造

物理ファイルは、実際のデータをもつファイルであり、データをレコード単位で到着順に格納している。レコードは、レコード長、制御情報及び幾つかのフィールドから成り立っており、同一物理ファイル内ではフィールドの並びはすべて等しい。図2にレコードの形式を示す。

(2) 論理ファイルの構造

論理ファイルは、実際のデータをもたない仮想的なファイルであり、物理ファイルを基に作成する。一つの論理ファイルは、最大8個の物理ファイルを基に作成できる。また、一つの物理ファイルからは、最大64個の論理ファイルを作成できる。

論理ファイルには、一つの物理ファイルからデータを取り出すものと、複数の物理ファイルのデータを結合して取り出すものがある。前者を射影による論理ファイル、後者を結合による論理ファイルという。

(a) 射影による論理ファイル

射影による論理ファイルは、物理ファイルの任意のフィールドを取り出し、並び替えたものである。したがって、論理ファイルのレコードと物理ファイルのレコードは、1対1に対応する。図3に射影による論理ファイルを示す。

(b) 結合による論理ファイル

結合による論理ファイルは、異なる物理ファイルの共通なフィールドの値を基に、レコードを結合して取り出したファイルである。結合に用いる共通フィールドを結合フィールドという。

物理ファイルの結合方式は、次に述べるとおりである。

- (i) 結合フィールドの値が一致したレコードが複数あると、そのすべての組合せでレコードを結合して取り出す。
- (ii) 結合フィールドが一致したレコードがないと、そのレコードは無視する。

レコード中のフィールドの並び順や取捨選択は自由であるが、結合フィールドは論理ファイルのレコード中に取り込んでおく。また、結合フィールドには前もってインデックスを作成しておく。図4に結合による論理ファイルを示す。

3.2 データベースの物理的仕組み

データベースは、物理的にはディクショナリファイル、データファイル及びログファイルから成っている。物理ファイル

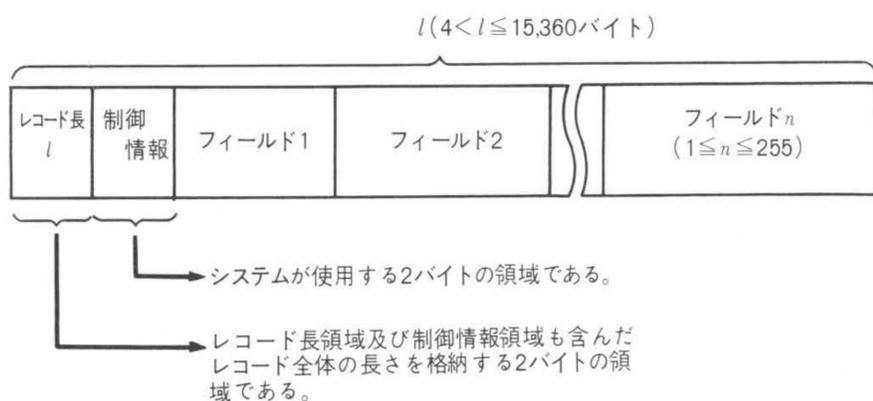


図2 レコードの形式 データベースに格納するレコードの形式は可変長レコード形式である。

(a) 物理ファイル

l_1	-	A1	B1	C1
l_1	-	A2	B2	C2
l'_1	-	A3	B3	
l_1	-	A4	B4	C4

(b) 論理ファイル

l_2	-	C1	A1
l_2	-	C2	A2
l_2	-	(00...00) ₁₆	A3
l_2	-	C4	A4

注：省略フィールドを論理ファイルに取り込むと、そのフィールドの値は(00...00)₁₆として取り扱う。

図3 射影による論理ファイル 基となる物理ファイルから必要なフィールドだけを取り出して、論理ファイルを作成する。

(a) 物理ファイル

l_1	-	A1	B1	C1
l_1	-	A2	B2	C2
l'_1	-	A3	B3	
l_1	-	A4	B4	C4

(b) 物理ファイル

l_2	-	×1	C1
l'_2	-	×2	
l_2	-	×3	C4

(c) 論理ファイル

l_3	-	C1	A1	×1	B1
l_3	-	(00...00) ₁₆	A3	×2	B3
l_3	-	C4	A4	×3	B4

図4 結合による論理ファイル 複数の物理ファイルに対し、結合フィールドを基にレコードを結合して、論理ファイルを作成する。

ルや論理ファイルのデータ定義情報及びインデックスは、ディクショナリファイルに格納し、データはデータファイルに格納する。また、更新履歴情報は、ログファイルに格納する。

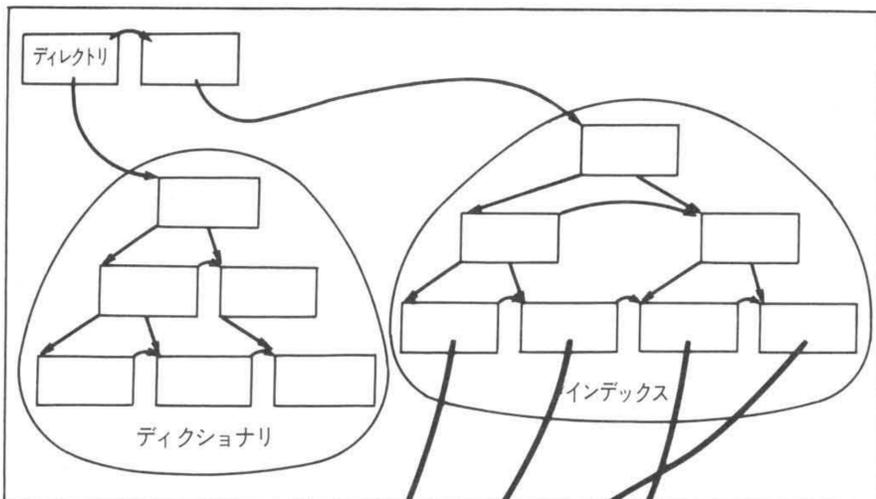
ディクショナリファイルは、システムにただ一つだけ存在するファイルであり、すべての物理ファイルや論理ファイルのデータ定義情報及びインデックスを格納している。データ定義情報は、一つの物理ファイル又は論理ファイルに対するデータ定義情報を1レコードとし、階層形索引構造となっている。インデックスは、キーフィールドの値を指定してデータをアクセスする場合に用いる索引であり、一つの物理ファイルに対して9個作成できる。このうち一つは必要インデックスとして物理ファイル定義時に指定する。これを主インデックスという。これに対し、物理ファイルを定義した後で必要に応じて作成、削除できるインデックスを副インデックスという。主インデックス及び副インデックスとも最大5個のキーフィールドを合成して作成できる。このキーを合成キーという。合成キーは、キーフィールドの値が重複するデータのと看有効となる。論理ファイルのインデックスは、基となる物理ファイルのインデックスのうち、キーのフィールドが

論理ファイル中に取り込まれているインデックスである。

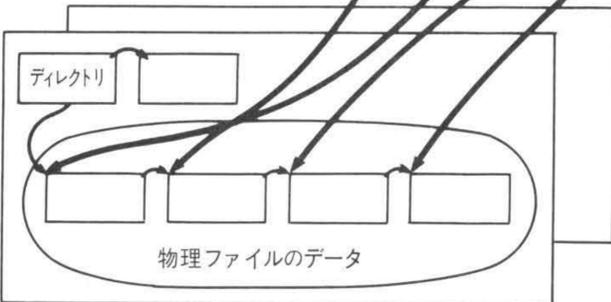
データファイルは、ファイルの先頭からアクセスする順処理と、指定されたインデックスによる乱処理に効果のある相対編成ファイルである。図5にディクショナリファイルとデータファイルの関係を示す。

ログファイルは、データベースに対して出力系のアクセスを行なった場合に、その履歴情報を格納するファイルである。ログファイルは、システムにただ一つだけ存在する。

ディクショナリファイル



データファイル



インデックスには、キーの値とそのキーをもつレコードのデータファイルの相対ポイントが登録されている。

ディクショナリファイル、データファイル共その領域はページというある一定の大きさで細分化し、これをビットマップにより管理している。したがって、ディクショナリやインデックス、物理ファイルのデータなどで領域が必要となったとき、必要な大きさの領域をシステムがダイナミックに割り当てる。

図5 ディクショナリファイルとデータファイルの関係 ディクショナリファイルには、データ定義情報及びインデックス情報を格納し、データファイルにはデータを到着順に格納する。

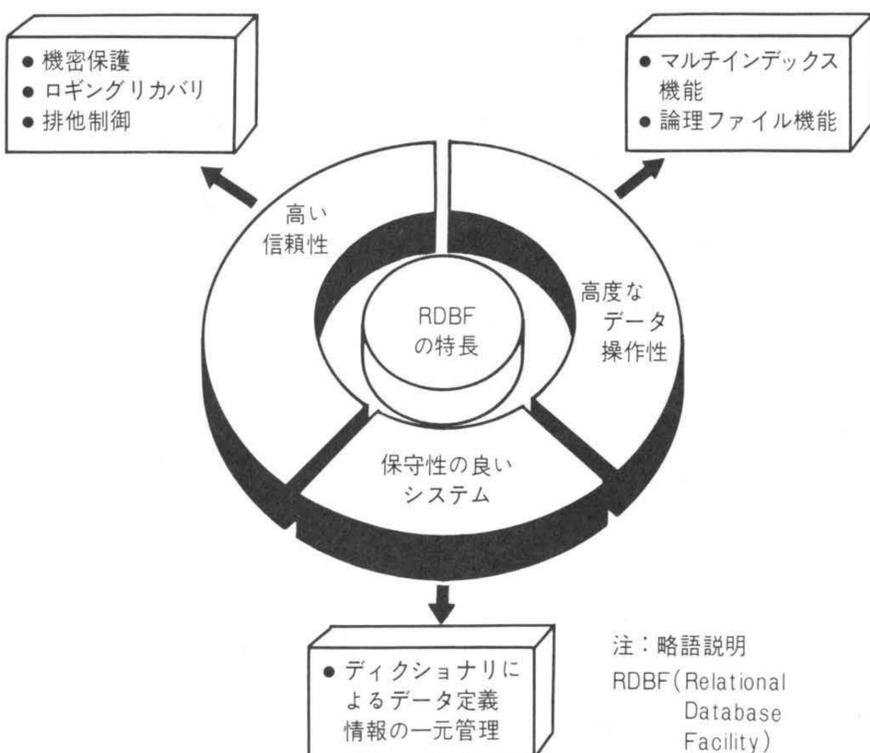


図6 RDBFの特長 RDBFはだれにでも高度なデータ操作ができ、データの独立を保障し、しかも高い信頼性を保証することを特長としている。

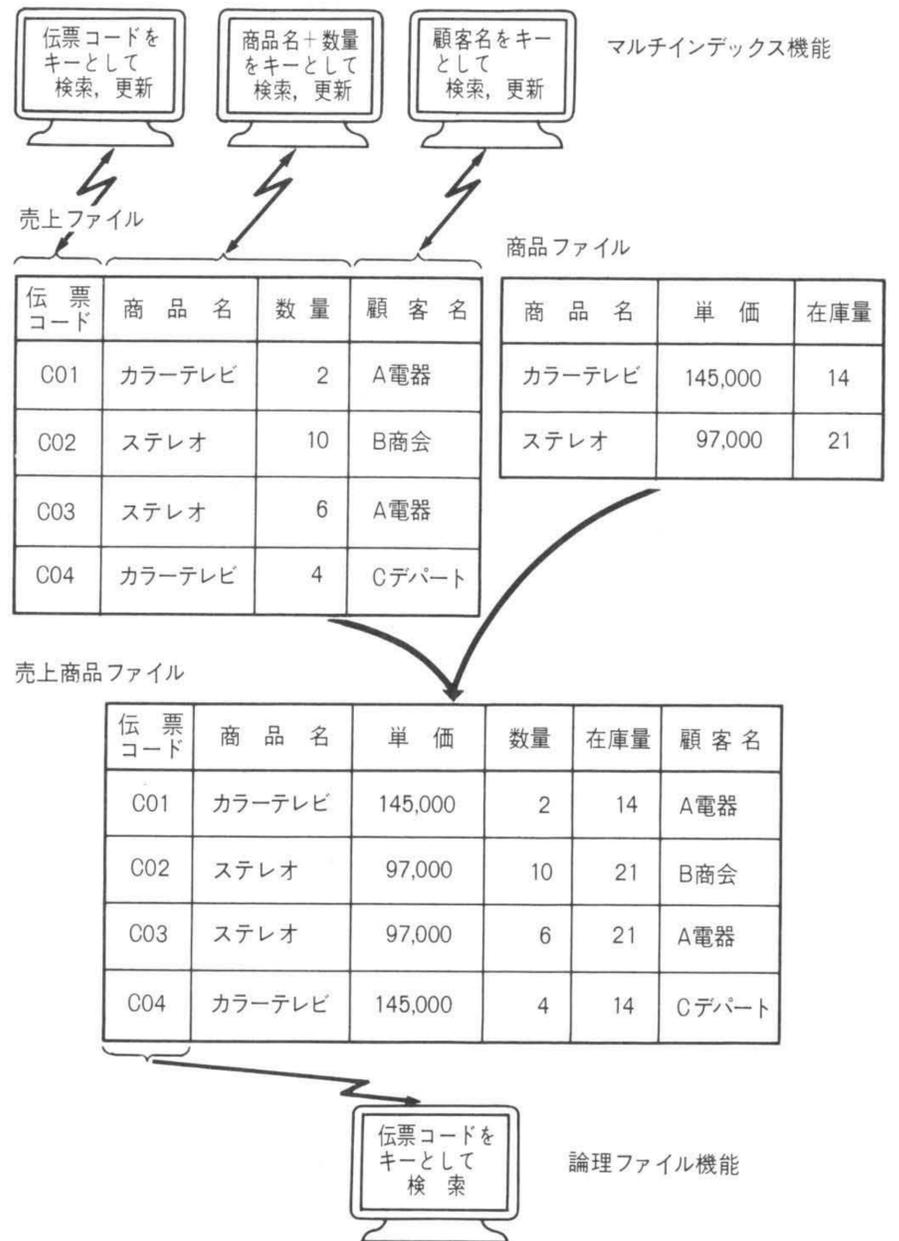


図7 データベースに対する多面的・論理的アクセス 論理ファイル機能, マルチインデックス機能を用いて、高度なデータの操作ができる。

4 RDBFの特長

RDBFは、専門知識をもたないオフィスのユーザーでも簡単に使えるように、特に使いやすさを重視したデータベースシステムであり、ユーザーの作成した業務プログラムやACE3 (Available Command Language for End Users3)を用いて、オフィスの業務で必要な多くのデータを、簡単に蓄積したり検索したりできる。図6にRDBFの特長を示す。

(1) 高度なデータ操作性

論理ファイル機能及びマルチインデックス機能により、多面的・論理的な検索ができるため、ファイルやプログラム数が削減できる。また、プログラムのロジックも簡略化できる。

図7で、売上ファイルには検索、更新目的に応じてインデックスが作成してあるため、分類処理を行なうことなくデータ処理ができる。また、売上商品ファイルの一つのファイルとして作成するのではなく、これを正規化して売上ファイルと商品ファイルの二つに分解し、必要なときに組み合わせて使用することによりデータの冗長性を減らすことができる。

(2) 保守性の良いシステム

データベースを運用中に、業務の拡張などによりデータベースの構造上の変更が必要となることがある。これに伴って、業務プログラムも変更しなければならないとすると、これは大きな負担となる。RDBFは、この問題を射影による論理ファイルとディクショナリファイルにより解決している。この例を図8に示す。この例では、売上ファイルから商品名、顧客名のフィールドを取り出した論理ファイルを定義し、この

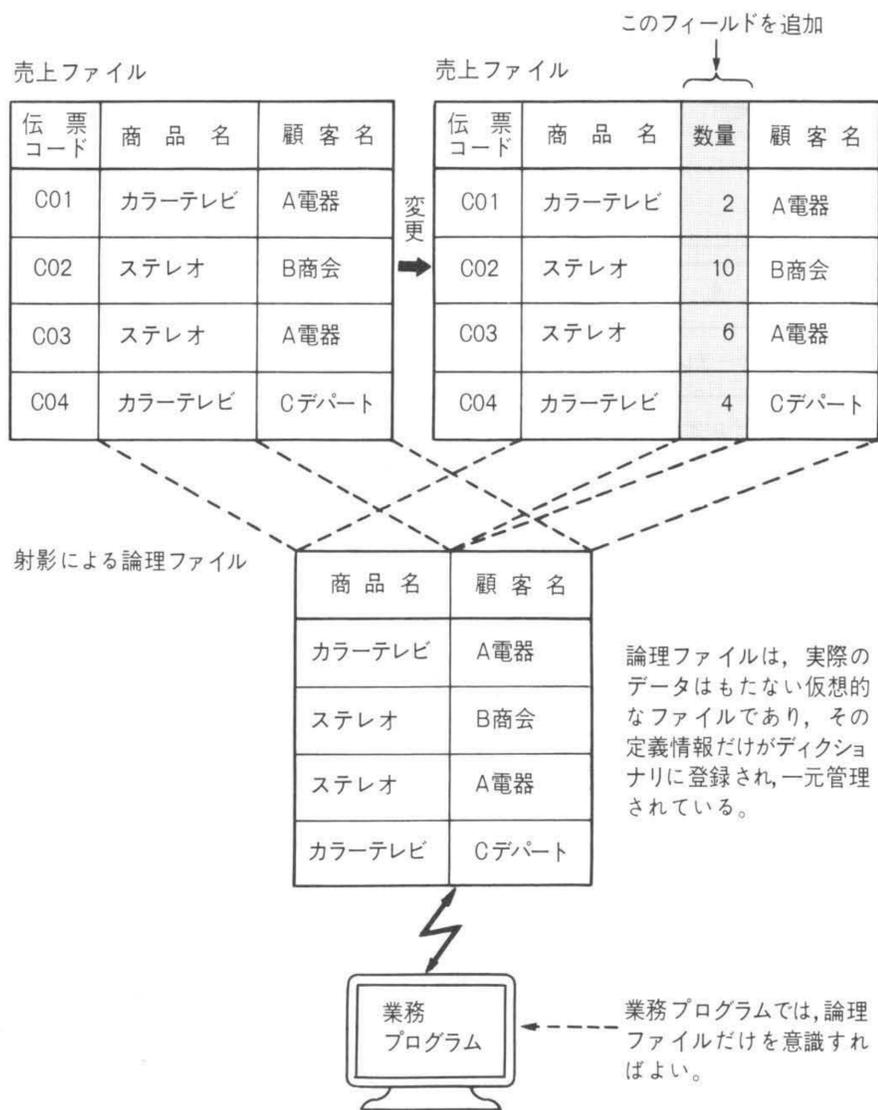


図8 論理ファイル機能による保守性の向上 物理ファイル(売上ファイル)に変更があっても、論理ファイルに変更がなければ業務プログラムを変更する必要はない。

論理ファイルを業務プログラムでアクセスしていたとする。業務変更により売上ファイルに売り上げた商品の数量に関する情報が必要になった場合、論理ファイルをアクセスしている業務プログラムで取り扱う情報に変更はないため、プログラムを変更する必要はない。ディクショナリファイルに登録している論理ファイルの定義情報だけを変更すればよい。

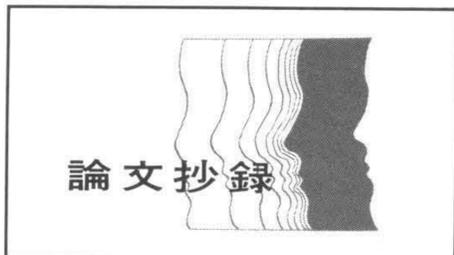
(3) 高い信頼性

- (a) ログ情報を取得する機能を備えているため、万一障害が発生しても、容易に復旧処理ができる。
- (b) 共通のデータベースに対する複数のユーザーからの検索、更新の処理を制御する機能を備えているため、誤って特定の情報を同時に更新したりすることのないように排他制御ができる。
- (c) ファイル単位の機密保護だけでなく、フィールド単位にも機密保護ができるため、データベースをきめ細かく管理できる。

5 結 言

今回開発したRDBFは、2次元のテーブル形式で表現する分かりやすい構造をもつリレーショナル形データベースである。また、データ操作が容易で簡単にデータの蓄積、検索が行なえる特長をもつ。この開発で、小形計算機分野でのデータベースを利用し、ファイルとプログラムの有効活用という所期の目的は達成できた。これにより、エンドユーザーは、多種多量のデータを効率良く、タイムリーに取り扱うことが可能となった。

今後は更に使い勝手を良くするため、手段を随時提供してゆき、各種のユーザーニーズに応じてゆきたいと考える。



音声パケット交換システムにおける遅延適応制御方式の提案

日立製作所 鈴木三知男・檜尾次郎・他2名
電子通信学会論文誌(B) J66-B7, 893~899(昭58-7)

音声パケット交換システムは、音声通信をパケット交換網を用いて実現しようとするものである。一般に、電話などの会話の場合、一人の話者が発声している時間は、通話時間の約40%程度以下であることから、有音部分であるトークスパート(Talkspurt)だけをパケット化することで、通信回線の有効利用が可能となる。しかし、パケット交換方式は原理的に待ち行列系であり、従来の電話交換にはない種々な問題がある。

筆者らは、音声パケット交換システムを実現する場合の基本的な問題として、次の二つを取り上げて考察した。すなわち、パケット交換網内のトラヒック量の増減によるパケットの網内伝送遅延時間の平均、及び分散に対して再生音声品質をどのように保証するかという問題、並びに受信側が各トークスパートの再生をいつ始めるかという再生開始時刻の決定問題である。

これらの問題に対して、本論文ではパケ

ット交換網での伝送遅延時間の分布が、負の指数分布で近似できることをARPAネットワークの実験などから示し、これを前提として考察している。

パケットの伝送遅延時間が負の指数分布に従う場合、各パケットの到着時間間隔のばらつきを吸収し、滑らかな音声を再生するために必要なジッタ吸収時間 T_j は、 $T_j = (-\log ek) \cdot T_{mv}$ となる。ここで、 T_{mv} は平均伝送遅延時間であり、 k はパケット紛失率(伝送遅延が大きく再生時刻より遅れ、棄却されるパケットの割合)である。これから、パケット交換網内のトラヒック量が変動し、平均伝送遅延時間 T_{mv} が変化する場合、パケット紛失率を一定として音声品質を保障するためには、ジッタ吸収時間を変化しなければならないことになる。平均伝送遅延時間 T_{mv} を求める方法として、本論文では、受信側で音声パケットの送信間隔と同じになる時間を発生し、各到着パケットをその

時間と比較することで求め得る方法が示される。

これらの考察結果を踏まえ、遅延適応制御方式が提案されている。本方式は、基本的な問題に対応して、内容的に二つから構成されている。一つは、パケット交換網内のトラヒック量の時間的な変動に対応するために、各トークスパートごとに求められるジッタ吸収時間を、次のトークスパートのジッタ吸収時間として適用していく方式である。この方式により、パケット網内のトラヒック量が変動しても、パケット紛失率を一定とすることが可能となり、音声品質が保証できる。他の一つは、各トークスパートの再生開始時刻の決定を、複数のパケットの到着時刻から求める方法であり、再生開始が著しく遅れる確率を減少する。最後にこれらの効果について、シミュレーション及び数値計算から各々確認されている。