

リアルタイム用リレーショナルデータベース 管理システム“ADF/RS”

Advanced Data Management Facilities for Realtime Application System “ADF/RS”

大容量ミニコンピュータ、マイクロコンピュータの登場に伴い、リアルタイム制御分野でも、コンピュータシステムで扱う情報量が飛躍的に増大しており、リアルタイムデータベース実現の要求が高い。

日立製作所では、このような状況に対処するため、リアルタイムシステムの要件である高応答性を損なうことなく、データベース処理アプリケーションシステムの飛躍的な開発効率向上を支援するソフトウェアとして、リアルタイム用リレーショナルデータベース管理システム“ADF/RS”を開発した。

ADF/RSは既に実システムに適用されており、この適用を通して、リアルタイム制御分野への性能面での適用性とアプリケーションシステムの開発効率の向上が確認されている。

廣田敦彦* Atsuhiko Hirota
大脇隆志* Takashi Ôwaki
山口康隆** Yasutaka Yamaguchi
林 利弘* Toshihiro Hayashi

1 緒 言

LSI技術の進歩による制御用計算機の大容量化、高性能化、更にはマイクロコンピュータ能力の向上に伴い、計算機制御システムは、近年、制御から管理、情報処理までを含めてトータルシステム化し、また同時に分散システム化するなど、ますます大規模、複合化する傾向にある。これに伴い、システムの中を流れるデータの量は、従来に比べ飛躍的に増大し、また個々のデータの重要性も高まっている。このため、計算機制御分野でも、データを効率よく、かつ信頼性高く一元管理する「データベース化」の考え方が必要なものとなってきている。

しかし一方では、このデータベース化を行なうには、高度なエンジニアリング能力が必要であり、これにこたえられる高級SE(システムエンジニア)の絶対数の不足、更にはSEの能力の限界を越える大規模システムの出現などにより、データベースを良質に設計し、それを円滑に運用・管理してゆくことが困難なものとなる。このため、これを支援するソフトウェアの開発によるソフトウェア生産性の向上が急務となっている。

一方、情報処理分野では、ソフトウェア生産性向上の決め手として、リレーショナルデータベース^{1),2)}が注目され、急速に浸透しつつある。しかし、応答性に対する要求レベルの違いなどにより、従来のリレーショナルデータベースは、計算機制御分野でのデータベース化に際しての問題を解決するものとはなっていない。

このような背景のもとに、計算機制御分野でのデータベース化を実現するため、HIDIC V90シリーズのリアルタイム用データベース管理の標準ソフトウェアとして開発したものがADF/RS(Advanced Data Management Facilities for Realtime Application System)である。

本稿では、ADF/RS開発の基本思想・方針、構成、機能及び適用結果について述べる。

2 ADF/RS開発の基本思想

従来のアプリケーションソフトウェアの開発、特にリアルタイム用オンラインシステムの開発では、データベース化を支援するソフトウェアが不十分であったため、以下に述べるような問題があった。

- (1) システムが複雑化し、SEが性能要求と処理要求を同時に満足するような良質なテーブル、ファイルを設計することが非常に難しかった。
- (2) システム改造・拡張時にテーブル・ファイルの変更・改造を、矛盾なく迅速に行なうことが困難であった。
- (3) 個々のプログラマが、性能を重視して作成された複雑なテーブル・ファイルを意識してプログラムを作成していたため、プログラムが複雑化し、プログラムの生産性が低下していた。
- (4) 処理要求の追加や、性能改善のためのテーブル・ファイルの変更が発生すると、プログラムの大幅な作り直しが発生することが多かった。

一方、リアルタイムシステム実現に当たっては、

- (5) 対象プロセスに追従できる高応答性能の達成が必要な要件である。

ADF/RSは、上記リアルタイムシステムの要件(5)を満足しながら、上記(1)~(4)の問題点を解決するデータベースを実現すべく、以下を開発の基本思想とした。

- (1) データベース最適設計技法の提供と、設計仕様一元管理により、設計・保守の容易化を図ること。
- (2) 扱いやすいデータベース操作インタフェースを実現し、プログラムの製作効率・信頼性の向上を図ること。
- (3) 主記憶の活用を含めて、性能調整可能な高性能データベース処理の実現を図ること。

3 ADF/RSの開発方針

一般のリレーショナルデータベースでは、

- (1) 物理構造を意識させない、単純な2次元の表でデータを表現する。

* 日立製作所大みか工場 ** 日立製作所システム開発研究所

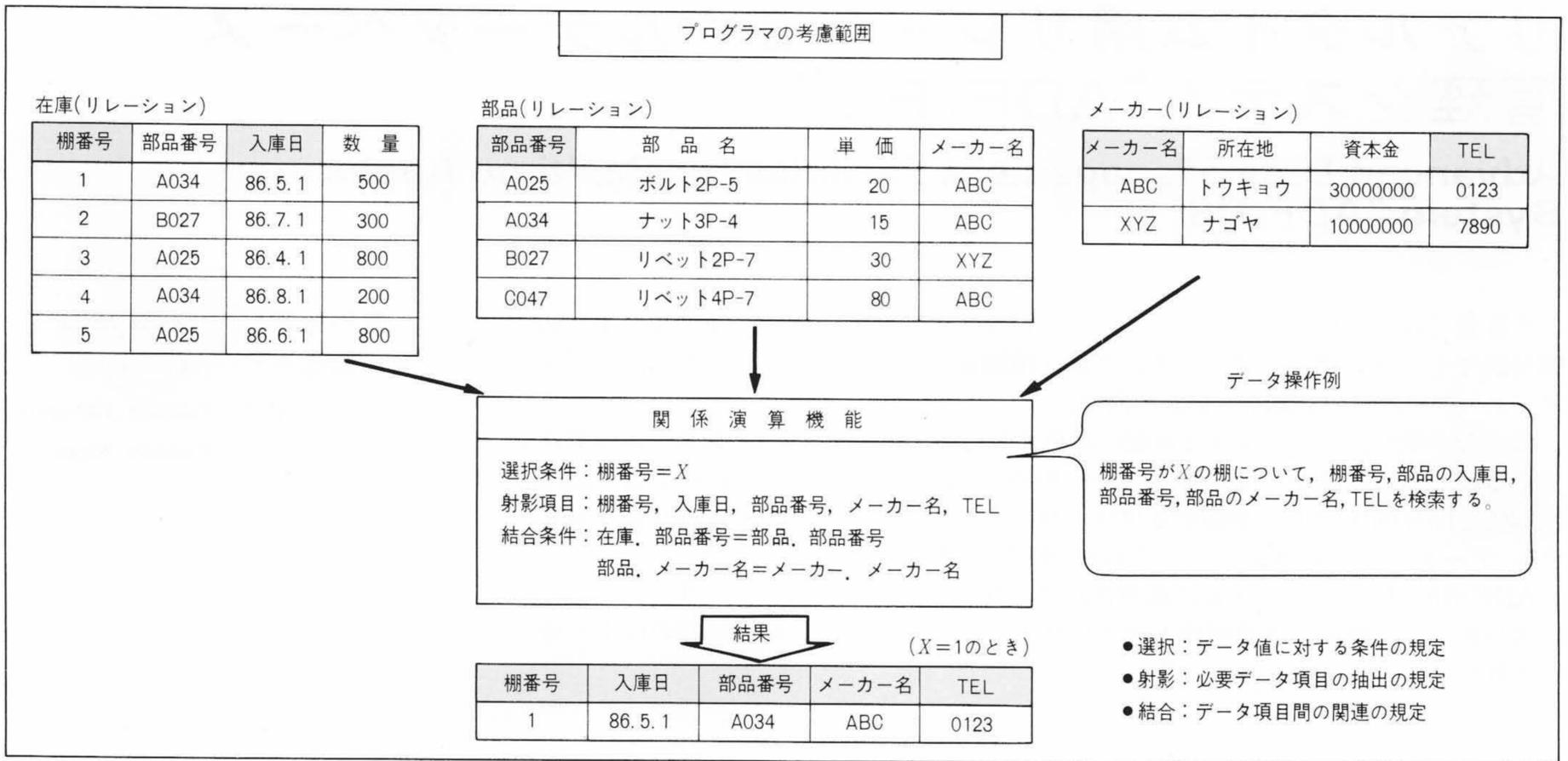


図1 リレーショナル データベースの特徴 リレーショナル データベースは、「2次元の表(リレーションと呼ぶ。)を用いた表現」と「関係演算機能」により、非定型業務向きの柔軟なデータベース インタフェースを実現している。

(2) 関係演算機能により、アクセス手段を意識せずにデータ操作が行なえる。

といった扱いやすいデータベース インタフェースを実現し、非定型業務に向けた高い柔軟性を確保している(図1)。

しかし、計算機制御分野は、定型的業務が主体であり、また高応答処理が要件であるため、一般のリレーショナル データベースを用いると、以下に述べるような問題がでてくる。

(1) 定型的業務では、扱うデータ構造がプログラム実行以前に決められるが、一般のリレーショナル データベースではデータ構造をデータ操作時に確定させるため(関係演算機能)、プログラムが複雑化し、また実行時の応答性も低下する。

(2) 一般のリレーショナル データベースでは、データ間の関連を管理しないことにより、データベース構築を簡易化している。このため、データベース全体の一貫性の保証はユーザーに委ねられ、大規模・複雑なデータベースの設計・保守を矛盾なく行なうことが難しい。

(3) 一般のリレーショナル データベースでは、データベースの物理構造設計をユーザーに解放していないため、性能要求に応じたきめ細かい性能設計ができない。

ADF/RSでは、上記のような一般のリレーショナル データベースの問題点を解決するため、以下を具体的な開発方針とした(図2)。

(1) データベース設計・保守の容易化

(a) 論理設計と物理設計の独立化による最適設計支援

データベースの設計対象を、プログラムが意識するデータベースの論理的なビュー(論理構造)と、記憶装置上でのデータの物理的な配置、構成法(物理構造)に分離する。更に、それぞれ処理要求、あるいは性能要求だけを考慮して独立に最適設計すればよいようにすることによって、最適なデータベース設計を支援する。

(b) データ ディクショナリに基づく設計・製作・保守一貫支援

断片的な設計情報を逐次格納でき、それを統合一元化管

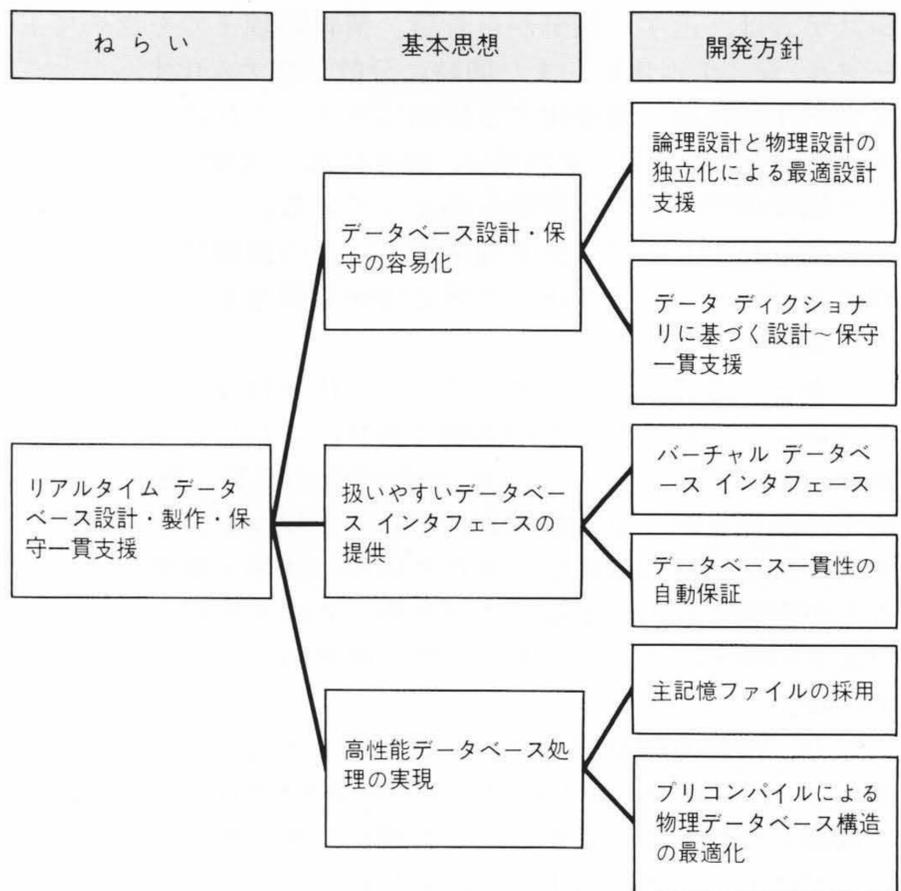


図2 ADF/RS開発の基本思想と方針 ADF/RS(Advanced Data Management Facilities for Realtime Application System)はリアルタイム データベース設計・製作・保守の一貫支援をねらいとし、「データベース設計・保守の容易化」、「扱いやすいデータベース インタフェースの提供」及び「高性能データベース処理の実現」を開発の基本思想とした。

理できるデータ ディクショナリを具備することにより、データ項目レベルからデータベース全体を構築していくボトムアップ設計アプローチを支援する。また、一元管理された情報を基に製作・保守を一貫支援する。

(2) 扱いやすいデータベース インタフェース

(a) バーチャル データベース インタフェース

データベースの物理構造やデータ アクセスの手段を、プログラマに意識させないバーチャルなデータベース インタフェースを実現し、プログラム製作の効率化、高信頼化を図る。

(b) データベース一貫性の自動保証

リアルタイム システムに必要な競合管理・障害対策に対する意識をプログラマから排除し、データベースの一貫性をシステムが自動的に保証することにより、システムとしての信頼性の向上を図る。

(3) 高性能データベース処理

(a) 主記憶ファイルの活用

物理データベースを構成するファイルの一部又は全部を、主記憶上に常駐化できるようにすることによって、特に厳しい応答性の要求されるアプリケーションでも使えるデータベースの構築を可能とする。

(b) プリコンパイルによる物理構造の最適化

関係演算に相当する処理のうち、事前に決められる定義処理(射影, 結合)を、設計仕様情報に基づきプリコンパイルすることにより、あらかじめ最適なアクセス パスを作成する。更に、これに対応する物理データベース構造を、性能要求に合わせて選択可能とすることによって、データベース処理のオーバヘッド低減を図る。

4 ADF/RSの構成

以上に述べた開発方針を具現化するため、ADF/RSは図3に示すシステム構成とした。

ADF/RSは、次の二つのサブシステムから構成される。

(1) データベース アクセス システム

データベース操作実行時に、データベースとユーザー間のデータ授受を統一的に実行するサブシステムであり、2種類のインタフェースを提供する。

(a) データベース操作プログラム インタフェース

アプリケーション プログラムからリンク可能なデータベース操作インタフェースであり、構造化プログラミング向き、かつホスト言語に独立なマクロ命令形式のインタフェースである。表1にマクロ命令一覧表を示す。

(b) データベース操作コマンド インタフェース

ユーザーがプログラムを作成することなく、端末から直

表1 ADF/RSデータベース操作マクロ命令一覧 ADF/RSのアプリケーション プログラムとのインタフェースは、データを操作するための基本マクロ命令群と選択条件の指定を行なうための拡張マクロ命令群から成る。

分類	名称	機能
基本マクロ命令	LCATS/WITHS	ストリーム(レコードの流れ)を発生させる(連続的ストリーム・離散的ストリーム)。
	INSERT	レコードを追加する。
	OBTAIN	レコードを取り出す。
	DELETE	レコードを削除する。
	EOBTAN	レコードの1アイテムを取り出す。
	EMODIFY	レコードの1アイテムを更新する。
拡張マクロ命令	ENDS	ストリームを終了する。
	CND	選択条件を指定する。
	ANDCND/ORCND	選択条件をAND/OR結合する。
	LCATSS/WITHSS	サブストリームを発生させる(1:N繰り返し構造の場合)。

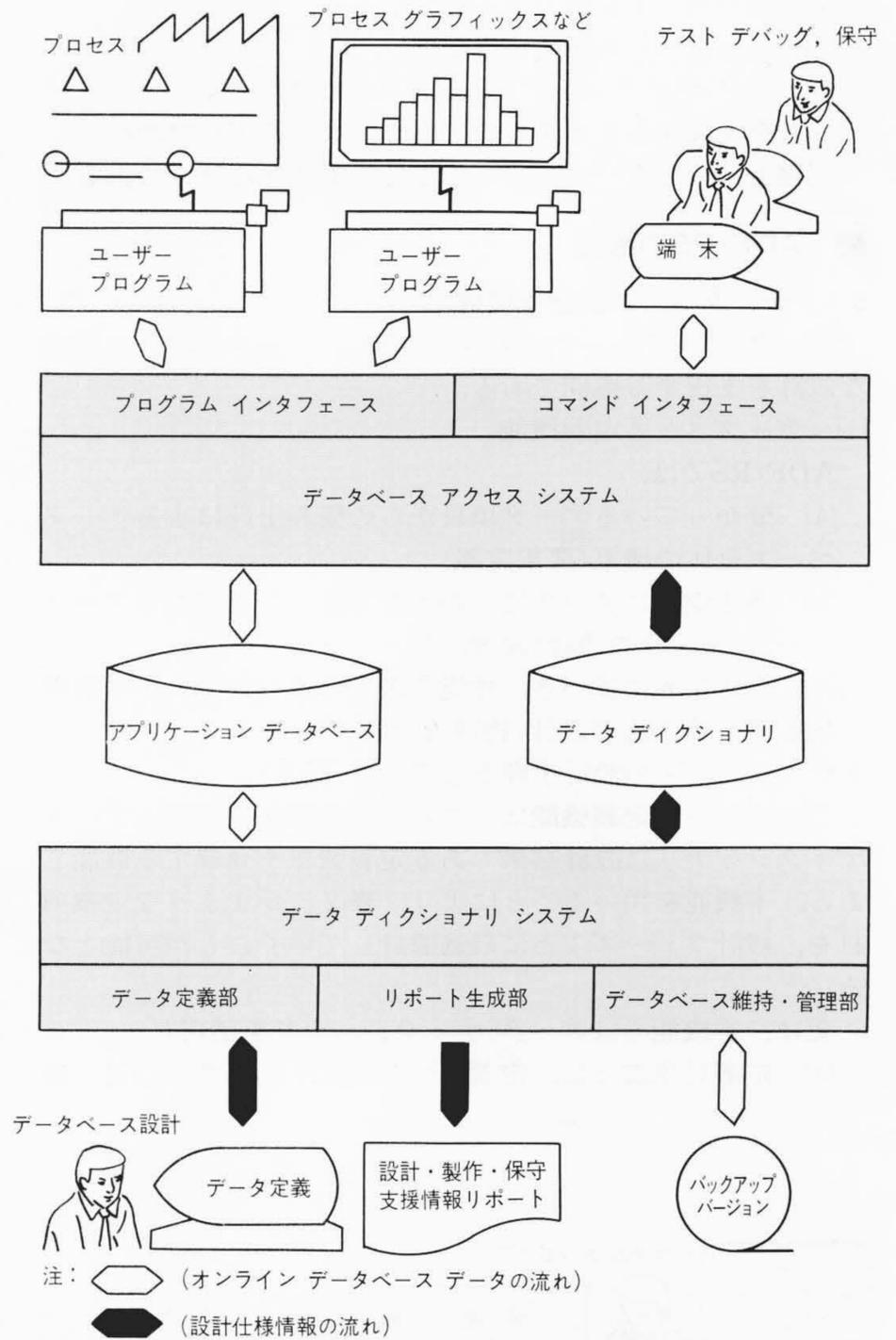


図3 ADF/RSのシステム構成 ADF/RSは、「データベース アクセス システム」及び「データ デクシヨナリシステム」の二つのサブシステムから構成される。

接データベースにアクセスできるコマンド言語形式のインタフェースである。データベースに関するプログラムのテスト デバッグ・保守を、プログラム インタフェースと同様に論理レベルで可能とする。

(2) データ デクシヨナリ システム

データベースに関する仕様情報を一元管理するデータ デクシヨナリを核とし、これに蓄えられた情報をベースに、データベースの設計・製作・保守を一貫して支援するサブシステムである。データ デクシヨナリ システムは、次の三つの構成要素から成る。

(a) データ定義部

データベースに関する仕様情報の定義に基づき、それをデータ デクシヨナリに格納する部分である。ユーザーは、これを行なうための専用のデータベース定義言語を用いることによって、仕様情報を容易にデータ デクシヨナリに登録し、また必要に応じて参照・変更・削除することができる。

(b) レポート生成部

データ デクシヨナリに格納された設計仕様情報から、SEやプログラマのためのデータベース設計・製作・保守支

援情報リポートを出力する部分である。

(c) データベース維持・管理部

データ ディクショナリに蓄えられた情報から、データベースを実現するファイルの生成や、バックアップの取得・回復といったデータベースの維持を実施する部分である。

5 ADF/RSの機能

5.1 データベース設計支援機能

データ ディクショナリをベースとし、データベースの最適な設計を支援する機能である。

(1) データベース定義機能

ADF/RSでは、

- (a) 分かっているデータ項目からの積み上げによるデータベース全体の構築(要求定義)
- (b) 要求定義に基づき処理要求を考慮した論理的なデータベース ビューの設計(論理設計)
- (c) 要求定義に基づき、性能要求を考慮したデータの物理的配置・構成法の設計(物理設計)

をデータベースの設計手順としている(図4)。

データベース定義機能は、この設計手順に従って、データディクショナリに設計結果である定義情報を登録する機能である。本機能を用いることにより、表2に示すような定義項目を、設計フェーズごとに最適設計していくことが可能となる。

更に、本機能ではデータ ディクショナリを活用し、

- (a) 定義対象ごとに、定義すべき項目をユーザーに対し誘

表2 ADF/RSのデータベース定義項目一覧 ADF/RSでは、データベース定義項目を各設計フェーズ最適に設計していくことが可能である。

設計対象	定義対象	定義項目
要求仕様 (要求定義)	データ項目	項目名称・意味 データ型・精度
	データ間の関連	依存関係・対応度数
処理仕様 (論理設計)	論理ビュー	処理に必要なデータ項目・意味
	操作属性	アクセス属性 (追加・更新・削除の可否)
性能仕様 (物理設計)	データの物理構造	ファイル編成法・リンク構成法 総レコード数
	物理アクセス性能パラメータ	索引ファイル仕様 バッファ仕様(面数・サイズ)
	物理デバイス	主記憶/補助記憶

導すること。

- (b) 定義登録・変更時に他の定義項目との整合性チェックを実施し、誤った定義を排除すること。
などにより、データベースの設計を効率よく、また誤り少なく行なえるようにしている。

(2) リポート機能

データ項目からの積み上げといったボトムアップな設計手順では、設計最終段階で個別に積み上げた結果をもとに、逆に大局的な見地から全体を鳥観し、グローバルな最適化や、不整合部分の排除を行なうことが、良質な設計を行なう上で重要である。

このため、ADF/RSでは、個々の情報を整理し、見やすい形での設計支援リポートを出力する。このことにより、設計の一貫性維持、未詳細化部分の抽出が容易に行なえるようになる。

5.2 高レベル データ操作機能

アプリケーション プログラムに対し、論理レベルのデータ操作インタフェースを提供する機能である。

(1) 論理ファイル機能

プログラムごとに、処理に必要なデータ項目を集めた仮想的なファイル(論理ファイルと呼ぶ。)をデータベース論理設計フェーズで物理構造とは独立に定義し、プログラムからは、この論理ファイルだけを意識してデータベースの操作記述が行なえるデータ操作インタフェースを提供する(図5)。本機能により、複数の物理的に異なるファイルに格納されている互いに関連するデータを、単純な表イメージで扱えるようになる。

また、論理ファイルに対するデータの検索は、必要とするデータに対する条件式を指定することにより行なえるようにし、どのようなアクセス方法を用いてデータを取り出すかというような「How(手段)」レベルの意識を不要にしている。図6に具体的なプログラミング例を示す。

これらにより、プログラムの複雑さを解消し、信頼性の高い、簡潔なプログラムを作成できるようになる。

(2) データ間関連管理機能

計算機制御システムでは、処理の大半は定型的業務であり、対象となるデータ構造やデータに対する処理構造が動的に変化することは少ない。このため、一般のリレーショナル データベースのようにデータ間の関連づけを操作時に行なう高い柔軟性をもったデータ操作を実現するよりも、あらかじめデータの関連を定義しておき、それに基づいて関連の制約に反

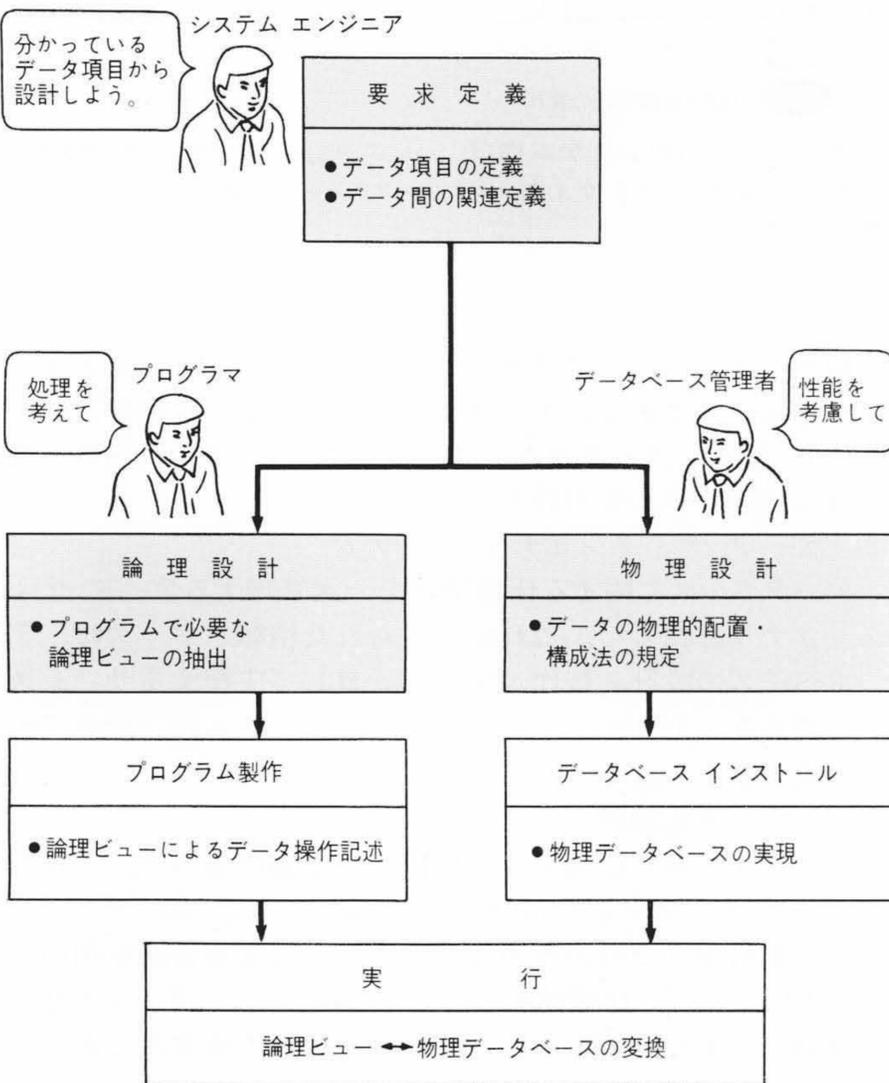


図4 ADF/RSにおけるデータベース設計手順 ADF/RSでは、「分かっているデータ項目からの積み上げによるデータベース全体の構築」、「論理設計と物理設計の独立化によるそれぞれの最適化設計」を支援するデータベース設計手順を実現する。

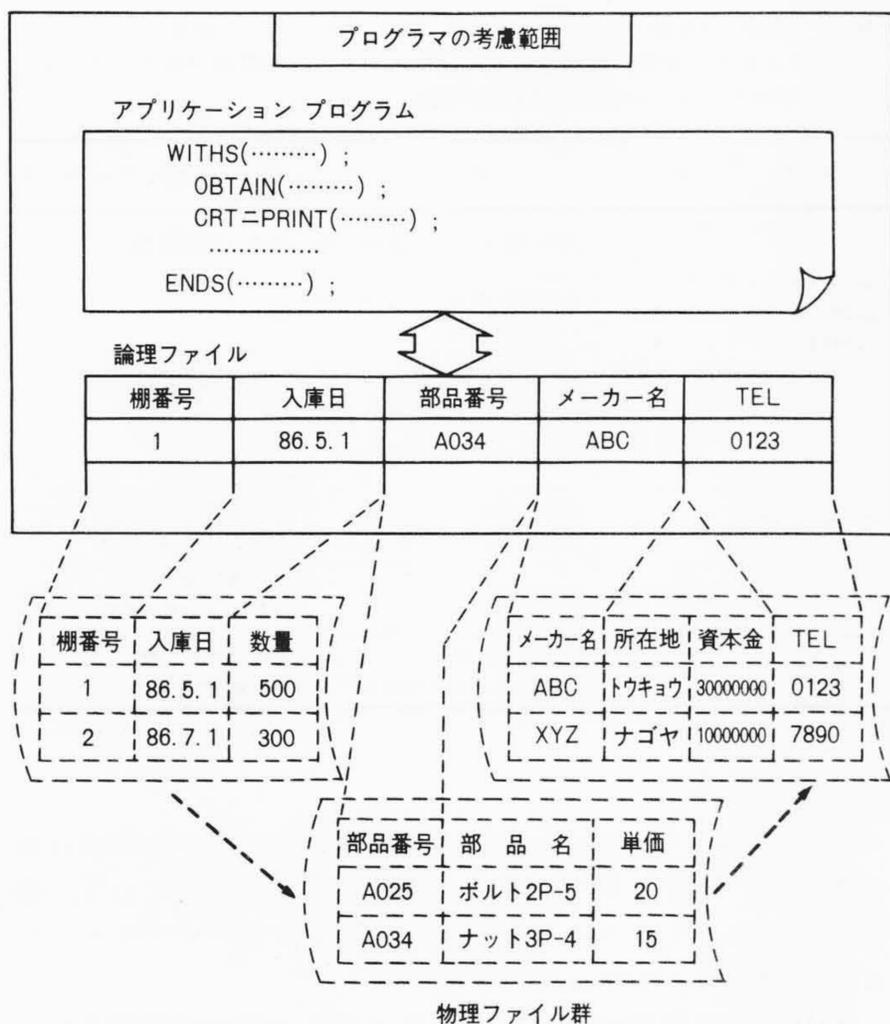


図5 ADF/RS論理ファイル機能 ADF/RSでは、論理ファイル機能を用いて、処理に必要となる項目だけを集めた仮想的な一つのファイルイメージでデータ操作が行なえる。このためプログラマは、リレーショナルデータベースのように複数の表を意識する必要がない。

しない範囲でデータ操作を行なえるようにするほうが、データ操作の信頼性が高い。

ADF/RSでは、データ間の関連を管理し、それに基づき自然な形でデータ操作が行なえるインタフェースを提供し、プログラムの生産性・信頼性及び性能の向上を図るようにしている。

(3) データベース自動排他制御機能

従来、データベースに対する排他制御は、個々のプログラムから占有、解放の手続きを発行させるといった形で実施していたため、プログラマに他のプログラムの動作タイミングを意識させるといった不必要な負担をかける結果となっていた。また、排他制御に起因して発生する各種の不具合は、テスト段階での検出が困難な場合が多く、実稼動中のシステムに重大な障害を発生させる場合もあった。

これに対しADF/RSでは、データベースに関する排他制御の一切を自動的に実施するようにし、プログラマから排他制御に関する意識を完全に排除している。

一方、参照を行なうプログラム同士は、同時に動作可能とする共有ロック方式の採用や、占有範囲の極小化を図ることなどにより、プログラムの並行処理性を確保している。

これらによって、高性能かつ高信頼度なリアルタイム制御システムの構築が容易に行なえるようになった。

5.3 高性能データベース処理支援機能

プログラムとデータベースとの間のデータ授受を、効率よく実行することを支援する機能である。

(1) 多様な物理データベース構成法の提供

リアルタイム制御システムでは、処理対象データに関連して、さまざまなアプリケーションに対応したさまざまなデータ処理要求や、異なるアクセス性能が必要となる場合が多い。

ADF/RSでは、これらの多様な要求を満たすため、

- (a) キー順(B-tree)、時刻順(Cyclic)、ハッシュなどの多様な物理ファイル編成法を選択できる。
- (b) データ間の関連を高速にアクセスする手段として、ポインタやチェーンリンクなど、物理レベルで関連を記憶できる。
- (c) 検索のための条件式評価が頻繁に起こるデータ項目に対して、評価を高速に行なうための索引ファイルを付加できる。

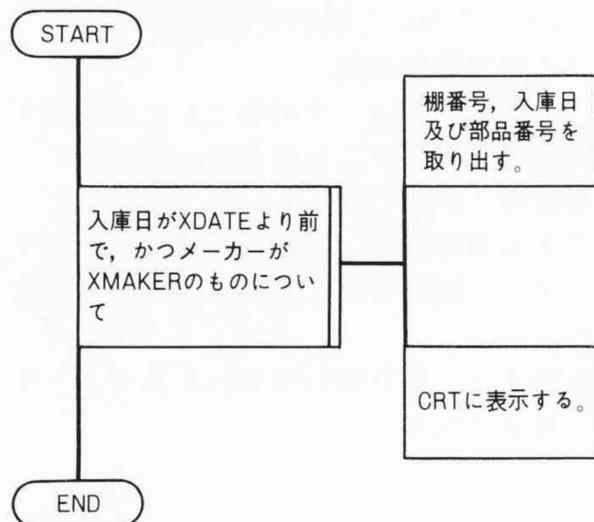
といったさまざまな物理データベース構成法を提供する。

更に、データベース アクセス性能向上の点から、

- (d) 頻繁にアクセスされるデータ、高速性が要求されるデ

論理ファイル「在庫部品(ZBUHIN)」

棚番号 (TNO)	入庫日 (NDATE)	部品番号 (BNO)	メーカー名 (MAKER)



(a) PAD記述

```

VAR COND : CONDITION_BLOCK ;
VAR SCB : STREAM_CONTROL_BLOCK ;
VAR WK : STRUCT(TANA : INT,
                DATE : DATE_TYPE,
                BUHIN : CHAR(4) ) ;
.....
RC=ANDCND(COND,
           ,CND(COND, ZBUHIN, NDATE, 'LT', XDATE)
           ,CND(COND, ZBUHIN, MAKER, 'EQ', XMAKER)) ;
WHILE LCATS(COND) OF (SCB) ON (ZBUHIN) /*条件を満たすレコードを順次 */
  REPEAT /* 繰返し処理する */
    EOBTAN (TNO) OF (SCB) TO (WK.TANA) ; /* 棚番号を取り出す */
    EOBTAN (NDATE) OF (SCB) TO (WK.DATE) ; /* 入庫日を取り出す */
    EOBTAN (BNO) OF (SCB) TO (WK.BUHIN) ; /* 部品番号を取り出す */
    CRT=(WK)オヒョウジスル ;
  END;
ENDS OF (SCB) ;
    
```

(b) データベース操作マクロ命令を用いたプログラム例

注：略語説明 PAD(Problem Analysis Diagram)

図6 ADF/RSデータベース操作プログラミング例 ADF/RSでは論理ファイルだけを意識し、また必要とするデータに対する条件式を指定するだけで、容易にデータ操作が行なえるため、プログラムが簡潔に記述できる。

ータを主記憶上に配置できる。

これらにより、アプリケーションの性能要求に適合した最適な物理データベースを容易に実現できる。

一方、これらの物理データベース形態の違いは、論理ファイル機能によりプログラム インタフェース上には現われないため、プログラムの生産性を阻害せずに、保守性の高いプログラムの作成が可能である。また、プログラム作成後でも、物理データベースの変更ができるため、プログラムの修正を伴わずに、アクセス性能のチューニングが可能である。

5.4 テスト・保守・運用支援機能

プログラム テスト、データベースの保守・運用管理を効率よく、信頼度高く行なうことを支援するユーティリティ群である。

図7に、データベース保守・テストを支援するユーティリティであるデータベース操作コマンド言語の使用例を示す。本コマンドを用いれば、プログラム インタフェースと同様に、論理レベルでデータベースに対し、データの追加・削除・更新や、見やすい形での印字出力が可能となる。これにより、データベース初期データの登録や、テスト用データの設定・確認などが、プログラムを作成することなく容易に行なえるようになる。

ADF/RSでは、本ユーティリティのほか、データベースのインストール、初期化、データベース バックアップ取得・回復など、各種のユーティリティを提供し、データベースの保守、運用・管理を容易に行なえるようにしている。

6 ADF/RSの適用状況と結果

表3はADF/RSと一般のリレーショナル データベースとの比較を示したものである。ADF/RSは、一般のリレーシヨナ

```
$ SDLU SOKODB/L=LP:
OBTAIN ALL OF BUHIN:
  BNO, BNAME
OBTAIN SURYO>1000 OF TANA:
  TNO, SURYO
END;
END;
```

すべての部品について、
BNO(部品番号), BNAME
(部品名)を印字し、
各部品について、SURYO
(数量)>1000の棚のTNO
(棚番号), SURYO(数量)
を印字する。

(a) 入力例

ADVANCED DATA MANAGEMENT FACILITIES (ADF) UTILITY LIST				
BNO	BNAME	TNO	SURYO	
' A025 '	' ボルト2P_5 '	18	1500	
		24	2000	
		45	3500	
' A034 '	' ナット3P_4 '	2	1800	
		37	2000	
' C027 '	' リベット2P_7 '	6	2500	
' C047 '	' リベット4P_7 '	1	1200	
		14	1900	
		38	2000	
		40	1700	
' D003 '	' コイル3FX '	49	1000	

(b) 出力例

図7 ADF/RSデータベース操作コマンド使用例 データベース操作コマンドを用いると、データベースの内容を見やすい形で印字出力できる。

表3 ADF/RSとリレーショナル データベースの比較 ADF/RSは、リアルタイム性能・信頼性の面で優位にあり、定型業務主体のリアルタイム アプリケーションに対する適用性が高い。

分類	項目	ADF/RS	リレーショナル データベース
データベース設計・製作支援機能	データ ディクショナリ	設計・製作・保守・実行支援	保守・実行支援
	物理構造	性能設計項目としてユーザーが選択可能	固定
	データ記憶媒体	主記憶・補助記憶	通常、補助記憶のみ
データ操作機能	インタフェース	2次元の表+階層表	2次元の表
	操作	選択・順序化(結合・射影は事前定義)	関係演算(選択・射影・結合)順序化
	関連の定義	事前定義	実行時定義(結合などの関係演算による。)
適用性		定型業務主体 リアルタイム処理向き	非定型業務主体 事務処理向き

ル データベースと比べ、操作の柔軟性の点で若干の制約はあるものの、リアルタイム性能・信頼性の面で優れており、特に制御用システムのような定型的な処理の多いアプリケーションに対して、十分に威力を発揮する。

ADF/RSは昭和60年6月に、第1号ユーザーに適用され、その後、電力・火力・鉄鋼・生産管理など各分野に適用されており、これらの適用を通じて、リアルタイム用としての所期の目標を達成していることが確認できた。特に顕著な効果としては、

- (1) データベースの物理構造が決まらなくても、プログラムの作成が可能になり、システムの開発期間を大幅に短縮。
- (2) データベース処理に関するプログラムの作成量を、従来の1/2~1/3に縮減。
- (3) 新たな処理要求に対するデータベースの拡張・変更を、既存プログラムに影響を与えずに迅速に対処可能。などが挙げられる。

7 結 言

以上、ADF/RSの開発思想・方針、構成、機能及び適用結果について述べた。

ADF/RSは、一般のリレーショナル データベースの特徴に加え、

- (1) データ間の関連管理機能
- (2) データ ディクショナリを用いた設計支援機能
- (3) 物理データベースの性能調整機能

など、リアルタイム アプリケーションを対象とした高性能化及び生産性向上支援機能を具備することにより、従来リアルタイム制御分野に適用困難とされていたリレーショナル データベースを、リアルタイム制御分野に対しても適用可能とすることができ、リアルタイム制御分野でのデータベース化実現の基盤が確立できた。

今後は、適用結果を踏まえ、更に使いやすいシステムとするように努力していく考えである。

参考文献

- 1) E.F.Codd: A Relational Model of Data for Large Shared Data Banks, CACM 13, No. 6 (1970-6)
- 2) E.F.Codd: Relational Database: A Practical Foundation for Productivity, CACM 24, No.10(1981-10)