

データ・ディクショナリを用いた データベースの総合支援システム

木村 淳美, 近藤 秀文

中村 史朗, 吉田 郁三, 加藤 孝 (日立)

1. はじめに

データベースの概念が登場して10数年、現在、アプリケーション・システムの建設に際して、データベース管理システム(以下DBMSと略)を利用したデータベース・システム化が一般化して来ている。データベースの普及とともに、最近では、リレーショナル・データベースを始めとするデータ・モデル理論、分散データベースおよびデータベース・マシン等の将来のDBMSアーキテクチャの研究が活発化している。一方、アプリケーション・サイドのデータベース・システム建設上の問題を解決するための支援ツール、特に、データベース設計支援ツールおよびデータ・ディクショナリ/ディレクトリ・システムに関する研究も活発に行われている。本報告はこの分野に関するものである。

データベース・システムは、企業のリソース(データ、プログラム等)の集中化を図ることによって、高度なデータの共用、データの一元管理、タイム・ラグの防止等、様々な利点をもたらす。その反面、リソース相互の複雑な関連を正確に把握しておく必要があるため、システム建設の負担を増加させる傾向がある。一方、このことは、システム建設者間の意思疎通が阻害されて、システムの統一性が欠如し、トラブルを発生する要因ともなっている。

このような危険を防止し、データベース・システムの建設を成功させるためには、システム建設者を統率しデータベース・システムに責任を持つデータベース管理者DBA*の存在が必要不可欠である。また、システム建設の負担を軽減し、統一性の向上のために、DBAと始めとするシステム建設者に対して、各種のツールを用意する必要がある。これらのツールの内でも最も重要なものが、データベース設計システムとデータ・ディクショナリ・システムである。

現在、この二つのツールは独立に論じられている。これに対し、本システムは、データ・ディクショナリを中核として、データベース・システムの設計から開発・運用・拡張の全フェーズに渡り一貫してサポートする、データベース総合支援システムを目標としている。この方式により、一貫した資源管理が可能となり、データベース・システム建設の負担軽減および生産性、整合性を向上することが出来る。以下、本システム的方式、構成および特徴を述べる。

*DBA: Data Base Administrator

2. データベース・システムの建設過程と支援ツール

建設過程は、図2.1に示す様に、以下のフェーズに分かれる。これらのフェーズは、実世界の実態ニーズの変化に対処して行くためのサイクルを描く。

- ・ 分析-実世界の実態およびニーズの分析把握
- ・ 設計-データベースの論理・物理構造の設計およびプログラム設計
- ・ データベースおよびプログラムの開発
- ・ データベース・システムの導入・運用

上記の各フェーズに応じ、データベース・システム建設の負担を軽減し、効率的かつ正確に機能遂行をサポートする、各種の支援ツールが必要となる。特に、

- ・ 実世界の実態・ニーズおよびデータベース・システム・リソースの相互関連

と全体に渡って把握することが重要である。大規模なデータベース・システムにおいては

- ・データ・アイテムの種類—数千～数万
- ・プログラム本数—数百～数千

に達し、人手による設計、管理が困難な状態になる。これを解決するための手段を提供するのが、データベース設計支援システム、および、データ・ディクショナリ・システムである。さらに、これらと各種支援ツールとの接続を行うことが、総合支援システム化へのキー・ポイントである。

3. データベース総合支援システム

本データベース総合支援システムは 図2.1 データベースシステム建設のライフ・サイクル、2. 前述したデータベース・システムのライフ・サイクルの全体に渡ってデータベース・システムの建設をサポートすることを目的としている。本システムの全体機能構成を図3.1に示す。基本的な考え方は各構成システムのデータ・ディクショナリを中核とし、これらに連続性を持たせることにより、以下の各システムと接続する方式である。

- (1) 要求定義・分析システムとデータベース設計システム
 - (2) データベース設計システムとデータ・ディクショナリ・システム
 - (3) データ・ディクショナリ・システムと各種支援ツール
- 以下、各Rの考え方を述べる。

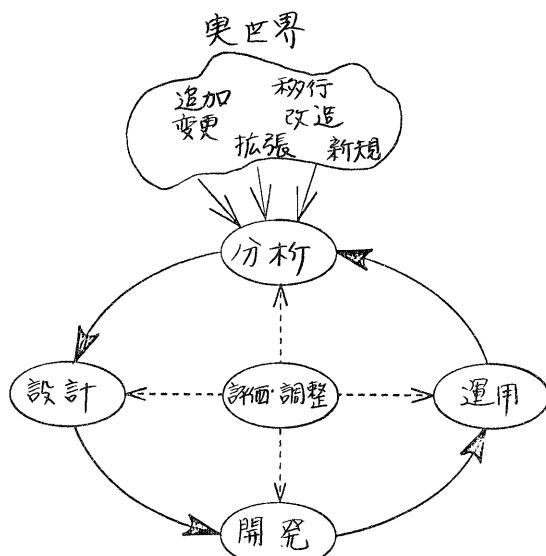
(1) 要求定義・分析システムとデータベース設計システム

今後のデータベース・システムにとって、実世界の要求分析から始まるデータベース設計ツールの開発は、最重要課題の一つである。現在、データベースの分野では、概念スキームの重要性が認識されている。概念スキームは、当該データベース・システムが対象としている実世界のデータ構造を、正確かつ忠実に表現したものである。したがって、まず、実世界の要求定義・分析技法が必要となる。一方、現在注目を集めているソフトウェア・エンジニアリングでは、実世界の要求分析からプログラム開発までが主体象となっている。この場合の要求分析の結果出てくる情報は、プロセス、データおよびこれらの相互関連情報である。これらの情報は、データベース設計にもそのまま使用可能である。したがって、これらの情報をディクショナリ化することにより、データベース設計システムのディクショナリと連続性を持たせることが可能となり、システム設計プロセス全体を支援することが出来る。

(2) データベース設計システムとデータ・ディクショナリ・システム

現行のデータ・ディクショナリ・システムは、主に、データベース・システム設計完了後の開発・運用の支援を対象としている。したがって設計後のドキュメンテーションや設計変更には威力を発揮するが、

- ・設計完了後のデータ量が膨大であり、ディクショナリ作成負担が大きい、
- ・上記に起因する入力もれによる実システムの世界とディクショナリ上での世界で不整合の発生



等の不都合が生じやすい。

データベース設計システムが対象とするディクショナリの世界では、

- ・データおよびそれらの属性
- ・データベースの論理構造および物理構造
- ・プロセス（プログラム）、プロセスとデータ、プロセス間の関連

が明らかにされている。したがって、この情報をデータ・ディクショナリ・システムのディクショナリに反映させることにより、連続性を持たせることが出来る。こうすることにより、システムの設計が完了した時点で、データ・ディクショナリ・システムのディクショナリが論理レベルでは出来上がっていることになる。

(3) データ・ディクショナリ・システムと各種支援ツール

システム開発・運用支援を目的として、これまでに各種のツールが開発されている。の中には、DBMSのユーティリティの一種として開発されているものもある。ディクショナリはメタ・データの一括管理を行なうため、これらのツールは、ディクショナリ結合の形をとった方が

- ・システム構成上
- ・利用者の負担面
- ・システム維持・管理面

いかににおいてもすぐれている。

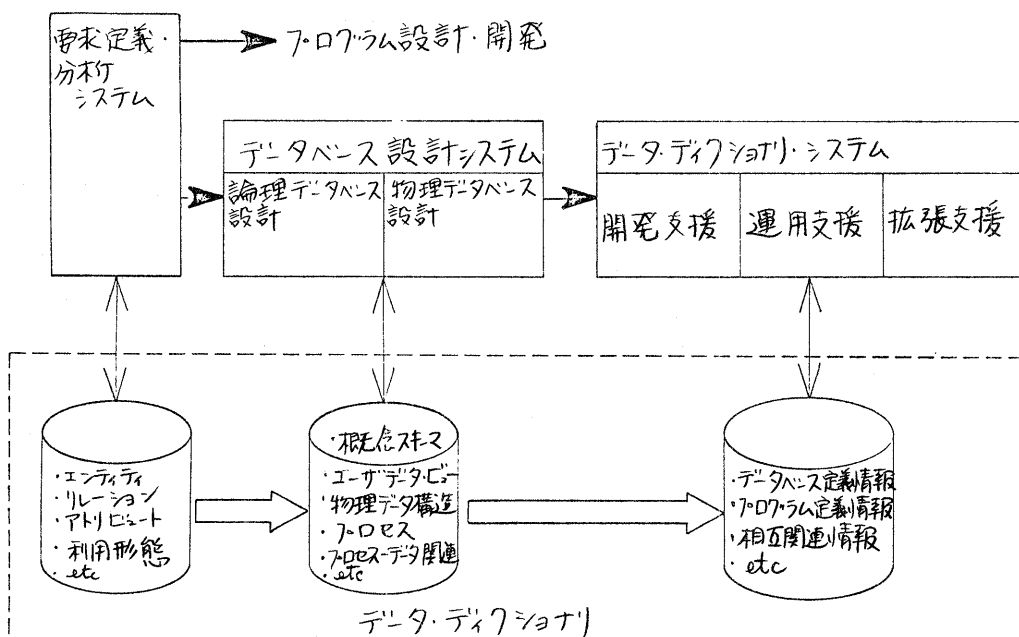
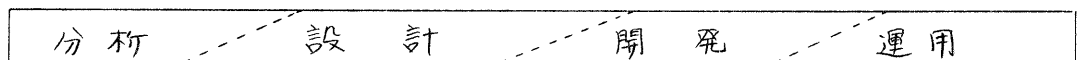


図3.1 データベース総合支援システムの機能構成

4. データベース設計とデータ・ディクショナリ

4.1 データベースの設計段階と処理内容

図4.1は、①データベースの設計段階、②各設計段階で使用する設計プログラム・ライブラリ、および③設計プログラムが使用するデータ・ディクショナリを示すものである。以下②で示す記号は、すべて図4.1中の記号を示す。

データベース設計は、図4.1に示すように、要求仕様分析①の結果得られたデータに基づいて、下記の2段階で実施する。

(1) 論理データベース設計

- ・概念スキーマ生成(⑥)：⑥のデータを入力とし、概念スキーマ⑦を生成する。
- ・データの意味に重点をおいたDBMSデータ構造の生成(⑧)：階層データ構造をもつDBMSであるADM (Adaptable Data Manager) を対象として、データの意味に重点をおいたセグメント設計および階層データ構造生成を行なう。
- ・⑦のデータを入力とし、⑧、⑨、および⑩のデータと出力する。

(2) 物理データベース設計

- ・性能に重点をおいたDBMSデータ構造の生成(③)：ADMを対象として、I/O回数およびディスク所要量と評価尺度とした階層データ構造の設計を行なう。
- ・③および④のデータを入力し、⑤のデータと出力する。
- ・格納構造生成：V SAMを対象として、レコード、Cエ、CAの各サイズの設定、I/O回数・ディスク所要量の見積り、データ・セットの適正配置設定などを行なう。
- ・⑤⑥のデータを入力し、⑥、⑦、⑧のデータと出力する。
- ・⑧のデータを出力する。

データベース設計が終了すると、結果がデータ・ディクショナリ・システムに渡され、運用に移される。各段階の処理は、自動処理および人手による処理からなり、対象システムの規模および複雑さにより、途中の段階から開始することもある。上述の論理および物理データベース設計のうち、本報告では、現在適用実験中の物理データベース設計について少し詳しく述べる。

4.2 物理データベース設計

(1) 入力データ：(i) ユーザ・データ構造(⑩)。応用プログラムが必要とするデータ構造を階層データ構造形式で表現したものであり、その例を図4.2に示す。階層データ構造の各要素はセグメントであり、セグメント・サイズおよびセグメント間のオカレンス数に関する情報等が記述される。(ii) ユーザ・データ構造アクセス要求(④)。ユーザ・データ構造のセグメントに対して、各セグメントに対する検索、追加、削除、および置換の各アクセス・タイプ、頻度が記述される。

(2) 処理内容：(i) 階層データ構造生成。ユーザ・データ構造と統合し、ADMの階層データ構造の候補を自動生成する。生成するは、(11)で述べた生成規則を使用する。図4.2のように入力データを入力すると、たとえば図4.3のような候補が出力される。(ii) 階層データ構造妥当性検証。生成された構造についてADMの規則違反の有無を状態遷移図法[8]により検査する。(iii) 最適階層データ構造決定。

(iv) および(vii)で述べる見積りに基づき、ディスク所要量がほぼ等しいときにはI/O回数の少ないものを選ぶという基準で階層データ構造を選択する[10]。(vi) レコード、Cエ、CAの各サイズの設定。それを用い、無駄な空きエリアの減少、I/O回数の減少、および1シリンダに固定の方法で算出する。(v) Cエ・CA分割発生率の見積り。Cエ・CA空き率とCエ・CA分割発生率との関係と算出する[9]。

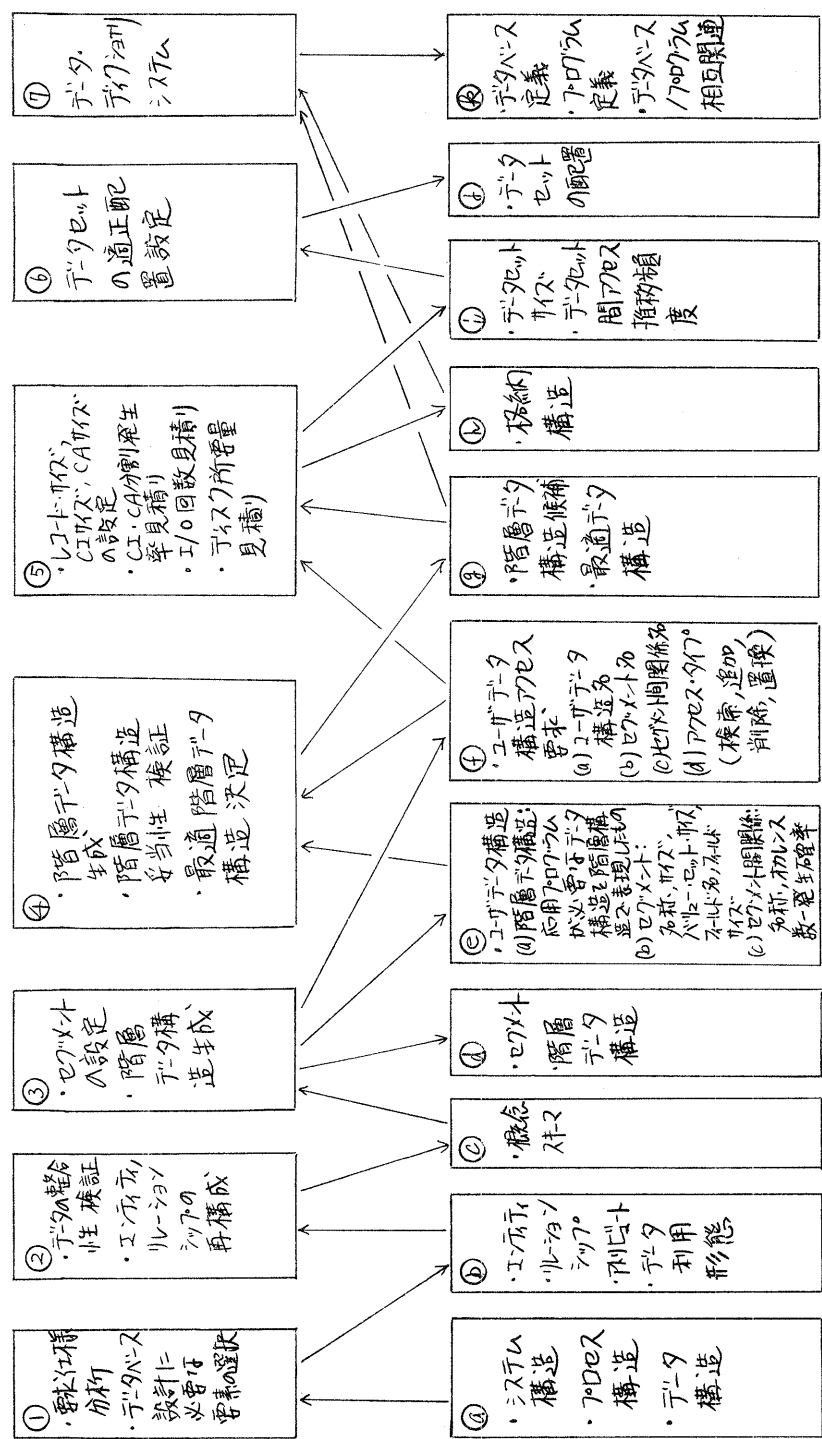
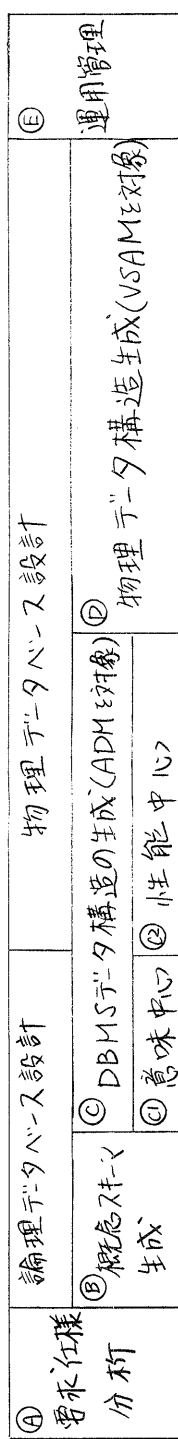


図4.1 データベースの設計段階, 設計プログラム・ライブラリ, データ・ディクショナリ

(vi) E/O回数見積り。(i), (iv), および(v)の結果を用いた見積り式によりE/O回数と計算する[12]。(vii)ディスク所要量見積り。(i), (iv)の結果と㊸で示したオカレンス数を用いたディスク所要量を計算する。

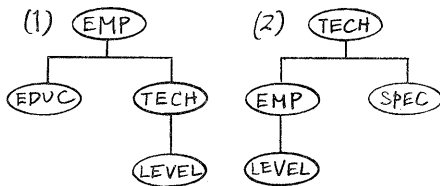


図4.2 ユーザデータ構造の例(入力データ)

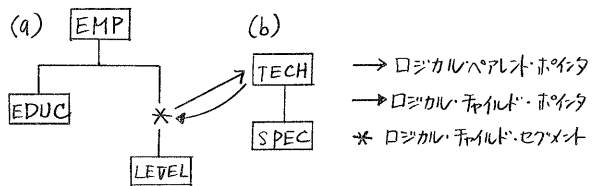
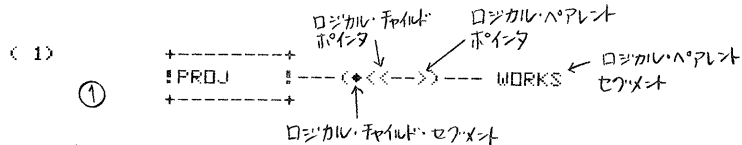


図4.3 階層データ構造の例(出力データ)

(3)出力データ
: 図4.4は生成された階層データ構造候補の1部分の出力例である。①は、その構造の図表示である。②は、アクセス法としてHISAMを選んだときのロジカル・レコード・サイズとして、216バイトを提案したことを示す。図4.5は、E/O回数およびディスク容量の出力例である。

INDIVIDUAL HIERARCHICAL ST.



INFORMATION FOR DBD DEFINITION

HISAM

②	05"カル・レコード".レングス(KSDS, INDEX COMP.)	= 0
	3"ロツキング".フアクタ(KSDS, INDEX COMP.)	= 0
	01"サイズ"(KSDS, INDEX COMP.)	= 0
	05"カル・レコード".レングス(KSDS, DATA COMP.)	= 2.160000E+02
	3"ロツキング".フアクタ(KSDS, DATA COMP.)	= 8.000000E+00
	01"サイズ"(KSDS, DATA COMP.)	= 0
	01% (KSDS, DATA COMP.)	= 5.000000E-02
	01% (KSDS, DATA COMP.)	= 6.000000E-02
	05"カル・レコード".レングス(ESDS)	= 0.000000E+00

HDAM (DATA)

	ラング"サイズ".レングス(ESDS)	= 3.040000E+02
	3"ロツキング".フアクタ(ESDS)	= 5.000000E+00
	01"サイズ"(ESDS)	= 0
	01"サイズ".フアクタ(ESDS)	= 7.000000E-01

図4.4 出力の例: データベース中の独立した階層構造および格納構造

図4.5は、4つの独立した階層構造全体について、E/O回数統計が12443.7、ディスク所要量は18シリンダであることを示す。4つの構造のアクセス法は、それぞれD, A, S, Aであることを示す。図4.5 出力の例: E/O回数およびディスク所要量

NO	I/O	CYL.	ACCESS METHOD
1	1.24437E+04	18	D A S A
2	7.76787E+03	17	A A S A
3	1.19085E+04	22	D A A A
4	7.23272E+03	21	A A A A

データベース中の独立した階層データ構造a
データベース中の独立した階層データ構造b
データベース中の独立した階層データ構造c
データベース中の独立した階層データ構造d

アクセス法
S: HISAM
D: HDAM
A: HDAM

5. データ・ディクショナリ・システム

5.1 全体機能および構成

データ・ディクショナリ・システムはDBAの幅広い活動をサポートするものである。従って、以前述べた様に単なるアプリケーション・プログラムとしてではなく、総合支援システムの中を考えた行かなければならない。

図5.1に、筆者らが開発しているデータ・ディクショナリ・システムの全体機能構成を示す。構成上の特徴は、ディクショナリ・アクセスにDBMSを用いていること、および、エンド・ユーザ言語と結合していることである。機能的には、設計後の開発・運用・拡張フェーズが対象となるが、データベース設計情報出力、クロス・レファレンス、システム変更シミュレーション等の機能は、設計段階においても適用が可能である。ディクショナリ・データベースは、7個のデータベースから構成されている。

以下に各機能の特徴について述べる。

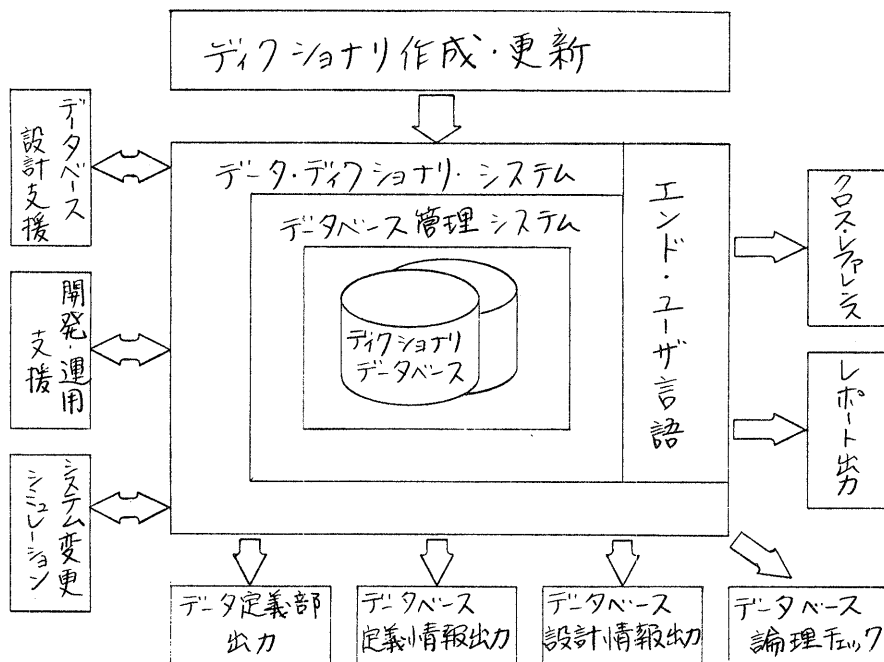


図5.1 データ・ディクショナリ・システムの機能と構成

5.2 ディクショナリ作成・更新機能

本機能は、データ・ディクショナリ・システムの基本機能であり、広範囲なメタ・データとディクショナリに蓄積し、更新を行なう。本機能の必要とすることは、

- ・作成・更新時の入力負担の軽減
- ・ディクショナリの完全性の確保

が計れることである。本システムでは、この点に関し、特にディクショナリ情報の基本となるデータベース情報に重点を置いて次の様に対処している。

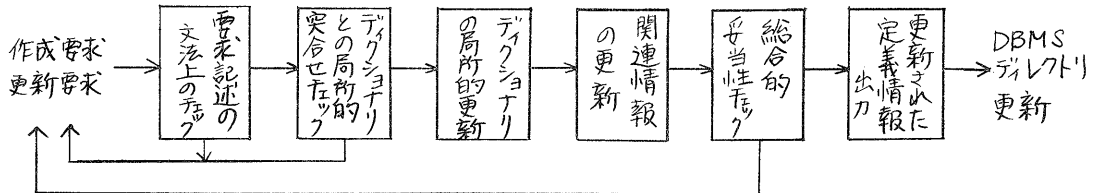
(1) 作成・更新時の入力負担の軽減

作成時には、DBMSへのデータベース定義パラメータと、そのまま入力する。既作成情報の更新時には、直接変更と指示された情報以外に、その情報と関連と

持つ情報の自動修正を行なう。

(2) ディクショナリの完全性の確保

作成時、更新時ともに局所のおよび総合的なチェックを強化している。
作成、更新の内部手順を次に示す。



以上の方式を実現することによって、作成・更新の入力負担の軽減と、関連部分の修正もれによるディクショナリおよびディレクトリの不整合を防止している。

5.3 データベース設計情報出力機能

データベースの設計ドキュメントは、データベース建設において共通の基盤となる重要な資料である。本機能は、次の2種類のドキュメントを作成する。

(1) データベース構造：データ構造を主体としたドキュメント

(2) セグメント・フォーマット：セグメントのフィールド構成を主体
本機能では、メインプロッタへの出力およびデータベース定義情報を全て記憶することによって、出力の質を向上させ、出カドキュメントの即時設計資料化を可能にしている。この結果、ドキュメントの作成・保守負担を排除するとともに、データベース設計ドキュメントの一貫性欠如によるデータベース・システム建設時のトラブルを防止することが出来る。図5.2に出力例を示す。

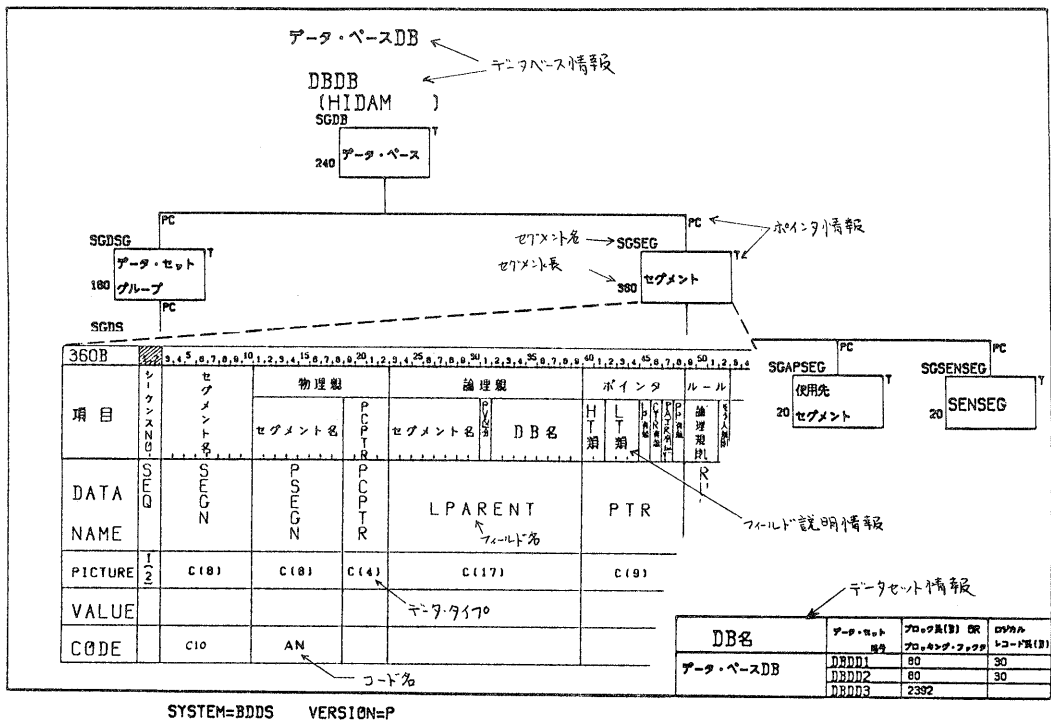


図5.2 データベース設計情報出力例

5.4 クロス・レファレンス機能

本機能は、データ・ディクショナリ・システムの中核機能の一つであり、ディクショナリに入力される情報の大半は、クロス・レファレンスでの使用を目的の一つとしている。したがって、その出力要求は多種多様な範囲におよぶと考えられる。したがって、出力要求の多様化に柔軟に対処できる機能とするため、ディクショナリを仲介としてエンド・ユーザ言語との結合を行なっている。本方式により、次のことが可能となる。

- (1) ディクショナリの情報を用いるため、エンド・ユーザ言語用のデータベース定義の負担が不用となる。
- (2) 標準手続を登録しておくことにより、以降は即時出力が可能である。
- (3) 突発要求に対して柔軟な対応が可能である。

本機能による手続例および出力例を図5.3に示す。(本稿では省略するがレポート出力機能は、本方式をユーザのデータベースまで拡張したものである。)

* CROSS REFERENCE LIST ***** RELATION OF FIELD AND PROGRAM

FIELD NAME	DATABASE NAME	SEGMENT NAME	PROCOPT	PROGRAM NAME
AP	DBDB	SGDB	I	%DIDBMT
BAM	DBDB	SGDB	I	%DIDBMT
BLOCK	DBDB	SGDSG	I	%DIDBMT
BYTES	DBDB	SGSEG	I	%DIDBMT
FLDINF	FLDDB	SGFLD	I	FLDISRT
CTLINE	DBDB	SGDB	I	%DIDBMT

```
%QFLD (WKLKEY1,WKLKEY2)
SEL %QFLDPRG
IF WSVL2 NOT =FLDKEYF
THEN
LET WSVL2=FLDKEYF
LET WSVL3=FLDKEYD
LET WSVL4=FLDKEYS
LST FLDKEYF,FLDKEYD,FLDKEYS,PROCOPT OF %QFLDPRG,
PROGN OF %QFLDPRG
ELSE
IF WSVL3 NOT = FLDKEYD
THEN
LET WSVL3=FLDKEYD
LET WSVL4=FLDKEYS
LST FLDKEYD,FLDKEYS,PROCOPT OF %QFLDPRG,
PROGN OF %QFLDPRG
ELSE
IF WSVL4 NOT = FLDKEYS
THEN
LET WSVL4=FLDKEYS
LST FLDKEYS,PROCOPT OF %QFLDPRG,PROGN OF %QFLDPRG
ELSE
LST PROCOPT OF %QFLDPRG,PROGN OF %QFLDPRG
ANYWAY
ANYWAY
```

(b) フィールド - プログラム関連

* CROSS REFERENCE LIST ***** RELATION OF PROGRAM AND FIELD

PROGRAM NAME	PROCOPT	FIELD NAME	DATABASE NAME	SEGMENT NAME
%DIDBD	G	KEY	DBDB	SGDB
%DIDBMT	I	AP	DBDB	SGDB
	I	BAM	DBDB	SGDB
	I	BLOCK	DBDB	SGDSG
	I	BYTES	DBDB	SGSEG
	I	CTLINE	DBDB	SGSEG
	I	DBAM	DBDB	SGDB

フィールドに対する
操作

(c) プログラム - フィールド関連

(d) クロス・レファレンス用標準手続

図5.3 クロス・レファレンス出力例

5.5 開発中および開発計画機能

以下では、現在開発中および開発を計画している機能についての考え方を示す。

(1) データ定義部出力機能

プログラムのデータ定義部の自動出力およびメンテナンスを行なう。本機能を真に有効なものにするには、定義情報の修正に伴うデータ定義部情報のメンテナンス機能の充実が不可欠である。例えばAという定義情報を修正した場合、Aを使用している全プログラムのデータ定義部の修正トランザクションを発生させる仕組みが必要である。

(2) システム変更シミュレーション

本機能は、ディクショナリ中の相互関連情報を用いる点ではクロス・レファレンスと同じであるが、次の二点を簡的に要する。

- (a) 問
- | | | |
|--|---|--|
| <ul style="list-style-type: none"> ・変更対象リソース ・変更内容 | → | <ul style="list-style-type: none"> ・変更により影響を受けるリソース ・影響の内容 |
|--|---|--|
- (b) したがって、データのセマンティクスを考慮に入れる必要が生じる。

6. おわりに

本報告では、データ・ディクショナリと中核としたデータベース総合支援システムを提案した。

物理データベース設計システムは、データベース記憶媒体のI/O回数およびスペース効率を最適にする物理データベース構造を導出する方式であり、性能効率を重視している。データ・ディクショナリ・システムでは、DBAの実質的な負担の軽減を目的として、妥当性チェック、自動修正の実施および出力情報の質の向上を計った。現在、これらのシステムは、実際のデータベース・システムへの適用を開始しており、適用の効果が明らかになると考えている。

今後、論理データベース設計および、システム拡張支援を述べたデータベース変更シミュレータに重点を置き研究を進める考えである。

参考文献

- 1) Data Base Management System Requirements : GUIDE / SHARE
Data Base Requirements Group (1970-11)
- 2) The Data Base Administrator : GUIDE (1972-11)
- 3) The Data Dictionary / Directory Function : EDP Analyzer (1974-11)
- 4) The Data Dictionary / Directory Requirements : GUIDE (1975)
- 5) H.C. Lefkovits : Data Dictionary Systems : Q.E.D. Information Sciences, Inc. (1977)
- 6) データベース・システムに関する調査—データ・ディクショナリ/ディレクトリ：データベース専門委員会報告書，日本電子工業振興協会(昭51-3)
- 7) ANSI / X3 / SPARC Study Group on Data Base Management Systems, "Interim Report" (1975-2)
- 8) J.H. Mommens : Automatic Generation of Physical Data Structures, ACM-SIGMOD ; 1975
- 9) D.G. Keehn et al, : VSAM data set design parameters, IBM System Journal, Vol 13, No3 ; 1974
- 10) M.F. Mitoma et al : Optimal data base scheme design, Proc. of 1st VLDB Conference ; 1975
- 11) 近藤 他 : 木構造DBMSを対象としたデータ構造適正化の一考察, 情報処理第19回全国大会 No 61-4 ; 1978
- 12) 近藤 他 : アプリケーションに則したDBMSのデータ構造適正化のための入出力量計算法, 情報処理第20回全国大会 No 60-2 ; 1979