

情報システム管理のためのデータベース

寺野隆雄・鈴木道夫

(電力中央研究所・情報処理研究所)

あらまし

我々は現在、ある電力会社の情報処理部門と共同で、社内の情報処理システムに関する諸情報を管理するためのシステム（以下ではこれを“情報管理サポートシステム”という）を開発中である。本論文では、このためのパイロットシステムとして作成した、情報処理システムのJCLを入力データとするようなデータベースについて報告する。これはシステムの保守にあたって、プログラムやファイル間の関連情報を把握するのに有用であると同時に、今後開発予定の情報管理サポートシステムの核となるものである。また、本論文では併せて、これに利用した小規模なCODASYL型データ管理システムGPLAN/DMS⁽³⁾の概略を紹介する。

1. 問題の背景

近年、情報処理システム（ソフトウェアシステム）の大規模化と複雑化に伴って、その開発、保守、運用の諸作業、このうちでも特に保守作業が、その困難さにおいて、質的にも量的にも大きな問題となってきている⁽⁶⁾。この背景として考えておかなければならないのは、オーに今日までにすでに膨大な数のプログラムが蓄積されているという点、オニに個々のシステムに対する要求が高度化しているにもかかわらず、情報処理部門の要員数が頭打ちになっているという点である。

一例として、現在我々が共同作業中の電力会社の情報処理部門をとり上げよう。ここでは、業務に必要な現在稼働中のプログラム（ほとんどがCOBOLで書かれている）は6000本にも達し、その数は今後もなお増え続けている。また、一般に企業では、一度作成されたプログラムは、業務に大きな変更がない限り、何度も手直しされて長期間使用されるのが普通である。したがって、よほど注意深く情報処理システム全体が管理されていない限り、その間

にシステムのドキュメントとその実体であるプログラムとが食い違うことがしばしば起こることになる。プログラムの総数が少なく、しかも、個々の情報処理システムごとに、システムの細部までを理解したベテランがいる場合には、このようなことが起こってもたゞしい問題にはならない。しかし、プログラム数が6000本にも達する場合には事情が全く異なり、ドキュメントとプログラムとの矛盾が原因で保守漏れが起こったり、バグが生じたりすることも多い。

次に要員の問題を考えよう。図1に同じ電力会社の場合の情報処理部門の要員数の推移を、図2に開発・保守の行なわれたプログラム数を示した。ここで注目すべきことは、開発・保守量の増加にもかかわらず合計人数にほとんど変化がないこと、女子プログラマが大巾に増加したことの二点である。オー点は、情報処理システムの大規模化と複雑化に対して、もはや従来のように要員の増加、すなわち手作業による人海戦術という方法がとれなくなっ

ていることを意味している。これに対し、オニ突はモジュラプログラミングやデシジョンテーブルなどのプログラミング技法の進歩により、プログラミング/コーディングの作業が以前に比べ易くなっていることを示すものである。

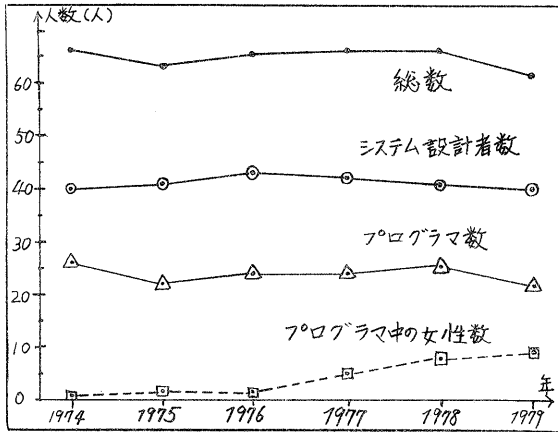


図1. 情報処理部門に属する要員数の推移 (ある電力会社の例)

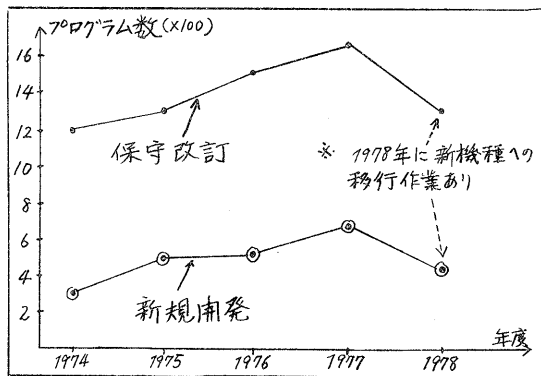


図2. 開発・保守改訂されたプログラム数の推移 (ある電力会社の例)

2. 情報管理サポートシステム の概念

それでは、蓄積プログラム数の増加と情報処理部門の人員数の相対的な減少という現状のもとで、膨大な情報処理システムを維持し、さらに、高度化された要求に対応して、より効率よく、より信頼性と保安性の高いシステムを

開発していくためにはどのようにすればよいのだろうか。我々は情報処理部門に属するシステム全体をまた一つの情報管理システムとして掌握することが重要であると考え、次のような二つの観点からこの問題にアプローチすることとした。

- (i) 情報処理システム開発方法論の体系化。
- (ii) 情報管理サポートシステムの開発。

開発方法論の体系化は、計算機ユーザの立場にたつて、情報処理部門の業務を見直すことにより、既存システムの整備と新規開発の効率化を図るものである。この体系の概要は図3及び表1の形にまとめられるが、これは次の考えを基礎に作成したものである。

- (a) 情報システムを業務分析から保守に至るまでのライフサイクルの中で統一的に把握すること。
- (b) 諸情報の一元的管理によって、その重複、矛盾を除去し、効率的利用を図ること。
- (c) 機械処理可能な作業を可及的に機械化するとともに、作業自体の機械化が困難な設計面に対しても、設計方法論を確立する、過去の設計ノウハウを共用化する等の形で効率化を促進すること。

情報管理サポートシステムは、この体系に基づいて開発されるべき、プログラマズワークベンチ(PWB)⁽²⁾によく似たシステムである。このシステムは最終的には図4に示すように、情報管理のためのデータベースとそれを分析する各種のアナライザ、ツール群、ならびに会話型を中心とするユーザインタフェース等から構成されることになる。なお、このシステムのねらいは次の二点に集約される。

- (a) 情報処理システムの開発や保守に

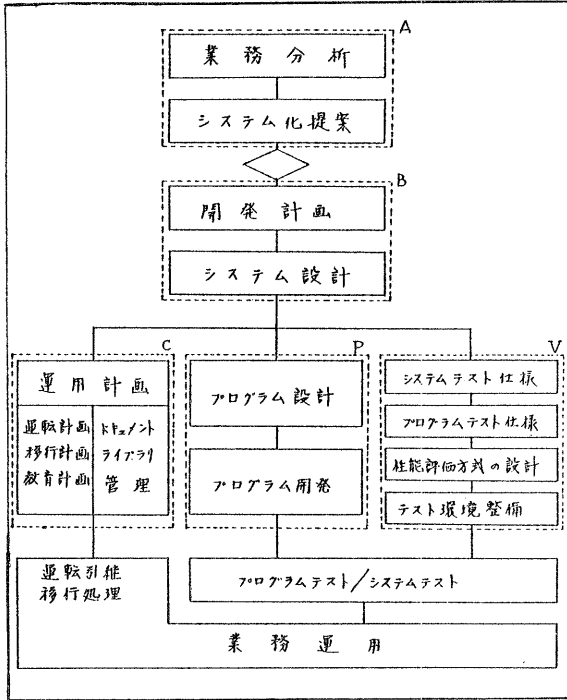


図3. 情報処理システムの開発手順 (計算機ユーザの場合)

表1. 情報処理システムの開発体系(作業とドキュメント)

管理フェーズ	主な作業内容	主なドキュメント等の作成/利用関係
A. アナリシス	<ul style="list-style-type: none"> 業務内容分析 システム化提案 	<ul style="list-style-type: none"> ドキュメント作成 → ドキュメント使用 システム提案書
B. ベースデザイン	<ul style="list-style-type: none"> 機械化設計 開発計画 	<ul style="list-style-type: none"> システム設計書
C. イントロール	<ul style="list-style-type: none"> システム運用計画 開発管理 	<ul style="list-style-type: none"> 運用計画書 フォーマットライブラリ
P. プログラマシオン	<ul style="list-style-type: none"> プログラム設計 プログラム開発 	<ul style="list-style-type: none"> プログラム仕様 プログラム
V. バリゲーション	<ul style="list-style-type: none"> テスト設計 テスト実施 	<ul style="list-style-type: none"> テスト仕様 テスト報告 (テスト済プログラム)
O. オペレーション	<ul style="list-style-type: none"> 運用管理 計算機運転 	<ul style="list-style-type: none"> 運転記録
M. 保守作業 (メンテナンス)	<ul style="list-style-type: none"> 現行システムの分析 システム改訂 	<ul style="list-style-type: none"> システム改訂記録

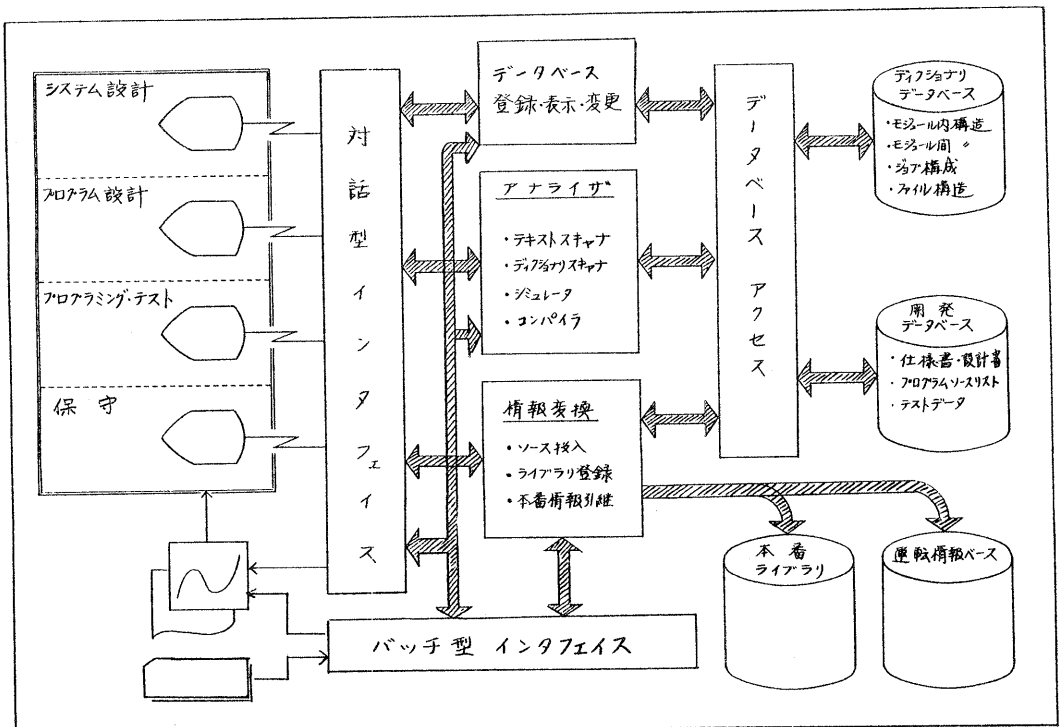


図4. 情報管理サポートシステムの概念図

あたって既存のシステムに関する諸情報を提供すること。

- (b) 開発中あるいは既存のシステムの分析を行なって、従来は担当者が手作業で行っていた仕事をサポートすること。

ここで示した情報管理サポートシステムの概念はもちろん情報処理部門の業務全般に関連するものである。しかし既存システムを整備しつつ、このシステムを開発していかねばならないという立場から3/3以降では特に情報処理システムの保守という視点で、図3に示したサポートシステムのデータベースについて検討していくことにする。

3. メンテナンス用データベースについて

Swansonは[5]で情報処理システムの保守作業を修正保守(Corrective maintenance)、環境適応化保守(Adaptive maintenance)、完全化保守(Perfective maintenance)の三種類に大きく分類している。このうち企業において最も重要なのは業務の変更に伴って遂行される環境適応化保守で、その際の問題点はおよそ表2のようにまとめることができる。

表2. 保守業務の問題点

管理面	<ul style="list-style-type: none"> システム依頼者の保守に対する考え方が 保守の実施(必要性)の判断基準がない 保守作業見積りの基準、方法が未確立
設計面	<ul style="list-style-type: none"> システム設計時点での保守への考慮の不足 システム設計思想・構造をこの保守が99/1
作業面	<ul style="list-style-type: none"> 対象プログラムの発見が困難 保守の波及範囲が捉えにくい ドキュメントの不備・矛盾 保守後の信頼性に疑問が残る

表2のうち、管理面と設計面で生じる問題点に対しては図2で示した設計方法論の体系化という手段で接近し、新規開発システムの保守性を向上させるしか方法はないだろう。すなわち、これらの問題点は、既存システムにはそのまま残されることになる。しかし、作業面で生じる問題点は、既存システムを図4のような適当なサポートシステムの管理下に置くことによってある程度解決可能である。そのためには、既存ドキュメントの探索、とりわけシステムの関連性把握に重きを置いた次のような5つの機能を持つシステムを考えればよい。

- (i) システム構成要素間の関連把握が可能なこと。
- (ii) データ名によるプログラムとファイルの検索が可能なこと。
- (iii) プログラム内のロジックが解析できること。
- (iv) プログラムのテスト履歴、保守履歴を保持していること。
- (v) ドキュメント修正とレポート作成が手軽にできること。

我々は、計算機ユーザの立場から考察しているのでSCCS[4]のようなバージョン管理機能は必要としない。

(i)~(v)の機能に対する情報源は表

表3. 情報管理サポートシステムの情報源

機能 \ 情報源	(i) システム関連	(ii) データ名による検索	(iii) プログラム解析	(iv) 履歴保持	(v) ドキュメント
ジョブ制御言語(JCL)	○			○	
ソースプログラム	△	○	○	○	△
ファイル・レイアウト	△	○	△		○
テスト仕様、テスト結果				○	
その他の設計書、仕様書				○	○

3のようになる。これらの情報から作成されるデータベースをメンテナンス用データベースと呼ぼう。これは図5のようなスキーマを持つと考えられる。

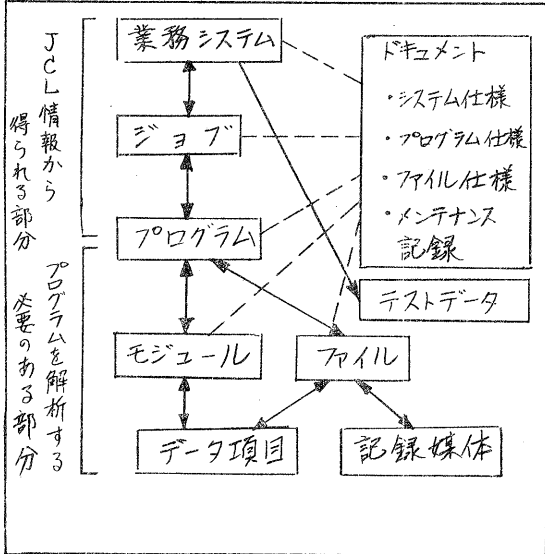


図5. メンテナンス用データベースの概念スキーマ

ここで、注意しなければならないのは、既存システムから直ちに図5のスキーマに従うデータベースができるわけではないことである。表3の情報源のうち、そのまま機械処理可能なものは、現存のものとは矛盾がないJCLとソースプログラムだけで、その他のものは、人手で何らかの形に加工してから入力しなければならないからである。情報処理システムは表4のようにさまざまな性格のものから成り立っており、ごく少数のもの(*印)しか図5のスキーマを満たさな

表4. 情報処理システムの性質

a. 今後のシステム*	↑ 良システム
b. 既存のシステム	
c. 保守の対象となるもの	* ↓ 悪システム
c1. ドキュメントが完備しているもの	
c2. ドキュメントに矛盾があるもの	
c3. ドキュメントのないもの	
d. 保守の対象外のもの	

いに注目しよう。このとき、図5を重視するあまり、良いシステムだけを対象としたのではサポートシステムの価値は少なくなる。保守の誤りは、対象システムが「思わぬ所」へ影響していたために起こることが多いからである。しかし、表4でランクがC以上のシステムすべてを詳細に解析するのは作業が膨大になり現実的でない。そこで我々は、プログラム、ならびにファイル構造の内側を解析しないで得られるジョブ、プログラム、ファイル間の情報のみをまずデータベース化し、サポートシステムの核とすることにした。そして良いシステムはもちろん悪いシステムに対しても必要に応じて適当なマイライザを用いてより詳細な情報をデータベースに蓄積していくのである。この場合は、ランクがC以上のシステムをすべて対象とすることが可能となり、しかも、データベースの初期作成にはJCLデータを用いるだけで済む。このとき、データベースのスキーマは図6のようになる。なお、§5で説明するパイロットシステムはこれに基づいて作成されている。

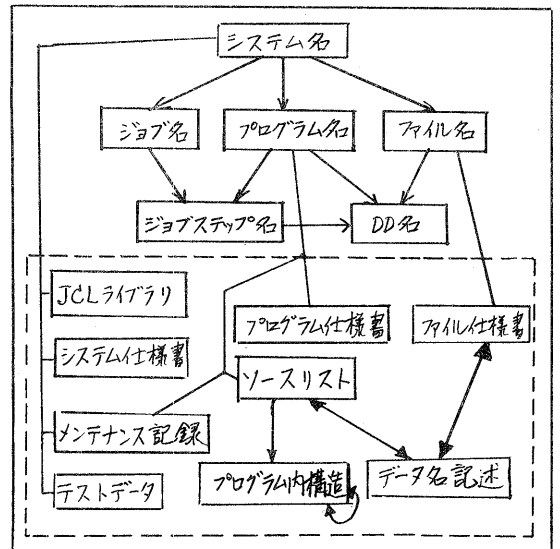


図6. JCLデータを中心に作成されるメンテナンス用データベース

4. GPLANデータ管理システムの概略

GPLANデータ管理システム (Generalized PLANning / Data Management System: GPLAN / DMS) は、Whinston, A.B.,等によって現在も開発が進められている比較的単純で小規模なCODASYL型データベース管理システムである。このシステムは、一般の計画管理問題のサポートに用いられるGPLAN (Generalized PLANning) と呼ばれるシステムの内部にあるデータベースを管理するために作成されたもので、すでに電力中研のミニコンNOVAの上に移植済みである。GPLAN/DMSについては作成者による教科書^[3]に詳しい解説があるので、ここではシステムの構成を中心にGPLAN/DMSを概観していくことにしよう。

GPLAN/DMSはFortranを親言語としており、システム自体もFortranで記述され、約4000ステップの大きさをもち100個あまりの副プログラム群からなる。システムの構成は図7に示したとおりである。

データベースを作成したいGPLAN/DMS利用者は、まず目的のデータベースのスキーマをごく簡単な文法規則にしたがってデータ記述言語 (GPLAN / Data Description Language: GPLAN / DDL) (図8Bを参照) で表わし、これをDDL解析プログラムにかけて、スキーマを計算機上に実現する。次に、GPLANデータ操作言語 (GPLAN Data Manipulation Language: GPLAN / DML) (図8Cを参照) を用いて、応用プログラムを作成し、スキーマにしたがってデータをデータベースに投入する。こうしてできたデータベースに、

やはりGPLAN/DMLを用いて作られた様々な応用プログラムを適宜追加して、データ管理システム全体を完成させる。

GPLAN/DMLはFortranで書かれた応用プログラムで使用されるので、その形式はすべてコマンドと呼ばれる副プログラムとなっている。すなわち、応用プログラムでは、データベース以外のデータの読み込みや変換、及びレポートの作成といった処理を普通のFortran文で行い、データベースにアクセスする場合のみ、GPLAN/DMLコマンドをCALL命令で呼び出すという使い方をする。換言すればGPLAN/DMLコマンドとはデータベースを利用するた

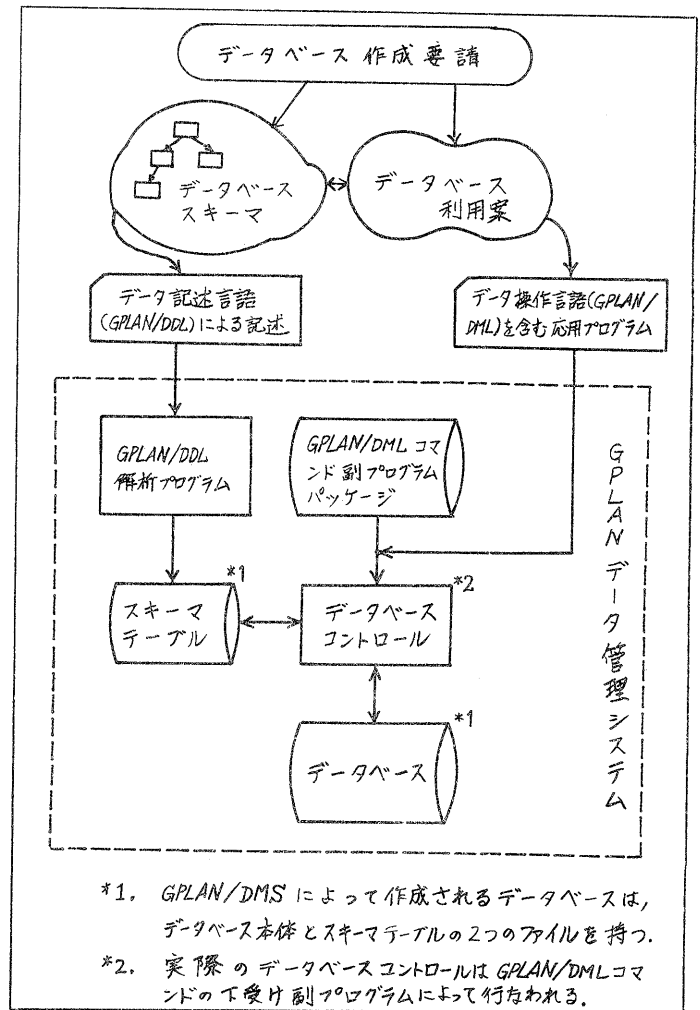
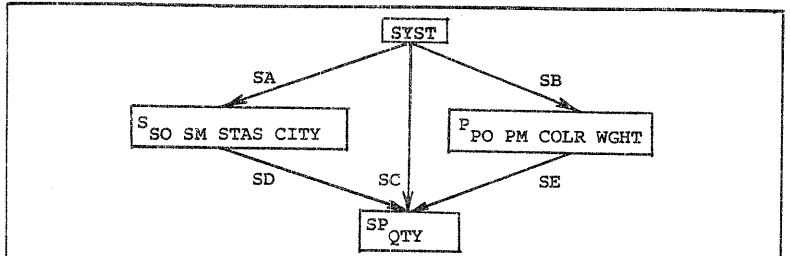


図7. GPLANデータ管理システムの構成

めのサブルーチン・パッケージであり、これとDDL解析プログラムが組み合わさってGPLANデータ管理システムを構成していることになる。図8にGPLAN/DDLによるスキーマの記述例とGPLAN/DMLを含む応用プログラムの例を示した。

現在のところ、GPLAN/DMLコマンドとしては、53個の副プログラムが登録されている(表5)。これら

- (a) データベースのユーティリティコマンド
- (b) スキーマの情報を得るコマンド
- (c) レコードの生成/消去に関するコマンド
- (d) 親子集合の生成/消去に関するコマンド
- (e) ネットワーク型データベースの探索(Navigation)に用いられるコマンド
- (f) データの投入/取出しに用いられるコマンドの6種類に大きく分類することができる。また、DMLのコマンドの意味、ならびに使用法については、(3)の7章に解説があるが、その使用法はかなり繁雑であり、十分使いこなすにはかなりの熟練を要する。



A. 供給者/部品データベースのスキーマ(バックマン線図)

RECORD	ITEM	FIELD	TYPE	LENGTH	DESCRIPTION
RECORD S	ITEM	SO	CHAR	5	1 SUPPLIER NUMBER SUPPLIER NAME STATUS CITY
	ITEM	SM	CHAR	20	
	ITEM	STAS	INTE	3	
	ITEM	CITY	CHAR	15	
	ITEM	QTY	INTE	4	
RECORD P	ITEM	PO	CHAR	6	PART NUMBER PART NAME COLOR WEIGHT
	ITEM	PM	CHAR	20	
	ITEM	COLR	CHAR	6	
	ITEM	WGHT	INTE	4	
	ITEM	QTY	INTE	4	
RECORD SP	ITEM	QTY	INTE	5	QUANTITY
	SET	SA	SORTED	50	
	OWNER	SYST			
	MEMBER	S			
	SET	SB	SORTED	50	
	OWNER	SYST			
	MEMBER	P			
	SET	SC	SORTED	QTY	
	OWNER	SYST			
	MEMBER	SP			
SET	SD	SORTED	QTY		
OWNER	S				
MEMBER	SP				
SET	SE	SORTED	QTY		
OWNER	P				
MEMBER	SP				
END					

B. GPLAN/DDLで上のA図のスキーマを記述した例

```

CALL GOPEN(1,2,2HOL,5,200,IER) ----- データベースを開く
IF (IER.NE.0) GO TO 50

C
C LIST SUPPLIER NUMBER AND NAME, PART NUMBER AND NAME,
C AND QUANTITY
C
WRITE (5,100)
CALL FEM (2HSA,IER) ----- 親子集合SAの最初の子を見出す
IF (IER.NE.0) GO TO 50
10 CALL SOM (2HSD,2HSA,IER) ----- 親子集合SAの現在子を親子集合SD
   IF (IER) 30,15,50 の現在親とする
15 CALL FEM (2HSE,IER) ----- 親子集合SDの現在子を親子集合SE
   IF (IER) 25,22,50 の現在子とする
22 CALL GEM (2HSD,2HSA,SNUMB,IER) ----- 親子集合SAの現在子に含まれ
   CALL GEM (2HQT,2HSD,QTY,IER) るデータを配列SNUMBにとり出す
   CALL GEM (2HPO,2HSE,PNUMB,IER) ----- 親子集合SEの現在親に含まれ
   CALL GEM (2HFM,2HSE,PNAME,IER) るデータを配列PNUMBにとり出す
WRITE (5,200) SNUMB,SNAMB,PNUMB,PNAME,QTY
25 CALL FEM (2HSD,IER) ----- 親子集合SDの次の子を見出す
   IF (IER) 30,20,50
30 CALL FEM (2HSA,IER) ----- データベースを閉じる
   IF (IER) 40,10,50
40 CALL GCLOS

```

C. GPLAN/DMLを用いた応用プログラムの一部 (下線部がGPLAN/DMLコマンド)

図8. GPLAN/DDLによるスキーマの記述例とGPLAN/DMLを含む応用プログラム例

表5. GPLAN/DML コマンド名一覧*1

UTILITY COMMONDS (a)*2		DATA MANIPULATION COMMANDS				
Open data base	OPEN					
Close data base	CLOS					
Dump tables	DUMP					
Error messages	ERR					
SCHEMA INFORMATION COMMANDS (b)		KEY	CURRENT MEMBER	CURRENT OWNER	CURRENT RECORD	
Get length of item type	GLI	Delete record (c)	DRK	DRM	DRO	DRR
Get length of record type	GLR	Get data (f)	GETK	GETM	GETO	GETR
Get names of item types	GNI	Get field (f)	GFK	GFM	GFO	GFR
Get names of record types	GNR	Get key (e)		GKM	GKO	GKR
Check current member types	CMT	Get record type (b)	GTK	GTM	GTO	
Check current owner type	COT	Set field (f)	SFK	SFM	SFO	SFR
CREATE AND ADD COMMANDS		Set current member(e)	SMK	SMM	SMO	SMR
Create record (c)	CR	Set current owner(e)	SOK	SOM	SOO	SOR
Create record/store data (c),(f)	CRS	Set current record(e)	SRK	SRM	SRO	
Add member to set (d)	AMS					
Remove member from set (d)	RM					
Remove set (d)	RSET					
SEARCH COMMONDS (e)						
Find first member	FFM					
Find last member	FLM					
Find next member	FNM					
Find previous member	FPM					
Find member based on sort key	FMSK					

*1. 表5は[3] p.155より再録したものである。

*2. かっこ内の英字はコマンドの分類を示す。

(a) データベースユーティリティコマンド。

(b) スキーマ情報を得るコマンド。

(c) レコードの生成/消去に関するコマンド。

(d) 親子集合の生成/消去に関するコマンド。

(e) 探索(Navigation)に用いられるコマンド。

(f) データの投入/取出しに関するコマンド。

5. パイロットシステムの実現

移植されたGPLAN/DMSのテストと§3で述べたメンテナンス用データベースの有効性の確認を兼ねて、ミニコンNOVAの上にパイロットシステムを実現した。なお、パイロットシステムでは、図6のスキーマ中で実線で囲った部分は削除されている。これは、ソースプログラム等を加えるとミニコンの記憶容量の制限からごく小規模なテストしかできないこと、ならびに、パイロットシステムをデータベースの有効性の判定のためだけに使用する場合にはこれらのデータは不要であることによる。図9にあらためてパイロットシステムのスキーマを示した。コンで箱の外に

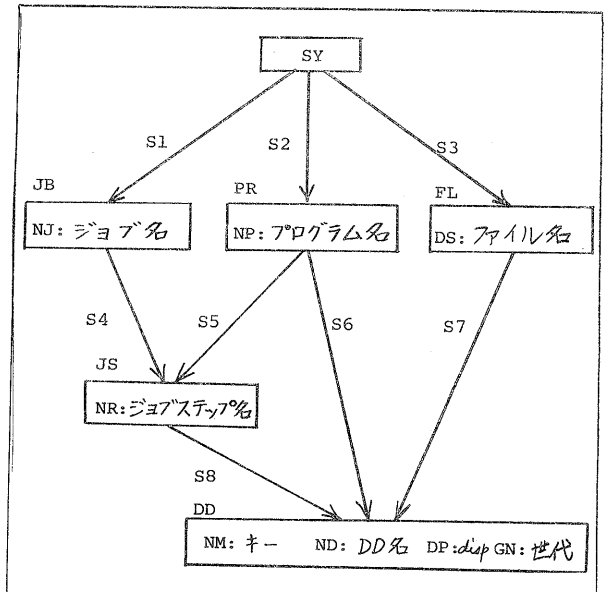


図9. パイロットシステムのデータベーススキーマ

*** JC1D1 VER=001 UATE=78-06-16

```

..EXC=JC1D1A,PUS=Y                                00000100
..DDI=SYS040,DSN=JC1DT79,UNT=1,DSP=0,K,K,RL=600    00000200
..DDU=SYS210,DSN=FM1DD03(0),UNT=M,DSP=S           00000300
..DDU=SYS120,DSN=JC1DD51,UNT=G,DSP=0              00000400
..DDU=SYS010,DSN=HA1DT15,UNT=G,DSP=0              00000500
..DDU=SYS020,DSN=JA1DD79,UNT=G,DSP=0              00000600
..DDU=SYS110,DSN=JC1DW01,UNT=W,DSP=N,RL=600,SPC=9  00000700
..DDU=SYS180,DSN=JC1DW02,UNT=W,DSP=N,RL=500,SPC=1 00000800
..DDU=SYS190,DSN=JC1DW03,UNT=W,DSP=N,RL=500,SPC=1 00000900
..DDU=SYS030,DSN=DUMMY,RL=600                     00001000
..EXC=JC1DUA,P=JSKSORT                             00001100
  
```

図10. パイロットシステムへの入力データの一部 (JCL ライブラリ出力)

つけた名前がレコード名、箱の中が項目名、矢印につけた名前が親子集合名を表わし、これらの名前がレポート出力例(図11)で使われているものと一致している。なお、SYはGRAN/DMSに特有なシステムレコードを表わす個々の名前である。(データベース中のレコードはすべてSYを始点とする道の上に存在する。)

パイロットシステム作成時の入力データには図10に示したジョブ制御言語(JCL)ラブラリのプリントイメージを用いた。投入データはおよそ3700行で、データベース中には業務システムごとに別ファイルの形式で保存されている。

また、データベース検索プログラムとしては、汎用レポートジェネレータを作成した。これにより、データベース利用者が

スキーマを知っていれば、アクセスパスを指定することにより任意のレコード実現値を出力できるようにしたもので、図11にその出力例をアクセスパスを指定するコマンドとともに示した。

このパイロットシステムを実用的なメンテナンス用データベースとするためには、図6あるいは図5に示したスキーマをすべて実現すべくネットワークを拡張するとともに、さらに会話型で使用できる使い方の易しいユーザインタフェース、出力形式の整ったレポート作成プログラム等を開発する必要がある。しかし、機能的には、現在のままだも情報処理部門の人々には、十分役立つものと考えられる。

出力用コマンド: LIST(NJ=ALL,(D(S1)))
SUB(D(S4),U(S5),D(S6),U(S7))
END

```

NJ: JC1D1
NR: JC1D1A
NP: JC1D1A
NH: 1 ND: SYS040 DP: 0 GN:
DS: JC1DT79
NH: 2 ND: SYS210 DP: S GN: 0
DS: FM1DD03
NH: 3 ND: SYS120 DP: 0 GN:
DS: JC1DD51
NH: 4 ND: SYS010 DP: 0 GN:
DS: HA1DT15
NH: 5 ND: SYS020 DP: 0 GN:
DS: JA1DD79
NH: 6 ND: SYS110 DP: H GN:
DS: JC1DW01
NH: 7 ND: SYS180 DP: N GN:
DS: JC1DW02
NH: 8 ND: SYS190 DP: N GN:
DS: JC1DW03
NR: JC1DUA
NP: JSKSORT
NH: 9 ND: SORTIN DP: 0 GN:
DS: JC1DW01
NH: 10 ND: SORTUK DP: GN:
DS:
NH: 11 ND: SORTOUT DP: 0 GN:
DS: JC1DW01
  
```

出力用コマンド: LIST(NP=(JSKSORT),UNIQUE,(D(S2)))
SUB(D(S5),U(S4))
SUB(D(S6),U(S7))
END

```

NP: JSKSORT
NR: JC1DUA
NJ: JC1D1
NH: 9 ND: SORTIN DP: 0 GN:
DS: JC1DW01
NH: 10 ND: SORTUK DP: GN:
DS:
NH: 11 ND: SORTOUT DP: 0 GN:
DS: JC1DW01
NR: JC1DUA
NP: JSKSORT
NH: 9 ND: SORTIN DP: 0 GN:
DS: JC1DW01
NH: 10 ND: SORTUK DP: GN:
DS:
NH: 11 ND: SORTOUT DP: 0 GN:
DS: JC1DW01
  
```

図11. レポート作成プログラムの出力

6. 結 語

本論文では、まず §1 で情報管理サポートシステムが必要とされる背景について説明し、§2 では、その概念をシステム開発の一体系とともに提案した。さらに §3 で、そのサブシステムとなるメンテナンス用データベースの妥当性について検討し、§5 でそのパイロットシステムの実現について述べた。また、§4 ではこのために用いた小規模なデータベース管理システム GPLAN/DMS の概要を説明した。

本論文で提案した既存情報処理システムの JCL を入力データとするようなデータベースは、非常に単純なものであるにもかかわらず、プログラムより上のレベルでシステム、プログラム、ファイル間の相互関係を把握するうえで有用であり、しかも、容易に実現でき、稼働中のシステムとも矛盾しないという特徴を持っている。もちろん、情報処理に長い間従事している人々にとっては、より詳細なプログラムより下、ファイルより下のレベルの情報が必要であろう。その場合、図3に示したようなソースリストを走査する各種のテキストアナライザが必要となる。さらに、アナライザを使用するためには、データ名の標準化が必要不可欠の条件となるが、その目的では(1)のようなメンテナンスサポートツールを積極的に利用するののも一つの方法であろう。

今後の課題としては、情報管理サポートシステム全体の実現が第一にあげられよう。しかし、その前には情報処理部門に従事する人々に対する教育の問題、端末装置などの計算機環境整備の問題、ドキュメントの日本語処理の問題など解決すべき課題がいくつか残っていることを指摘しておかねばならない。

謝 辞

GPLAN/DMS の移植作業に御協力をいただきました、筑波大学電子情報工学系の今野 浩助教授に感謝の意を表します。

参 考 文 献

- (1) 情報処理振興事業協会：メンテナンス支援プログラム MAID 利用者マニュアル。(1979年2月)。
- (2) Dolotta, T.A., and Mashey, J.R.: An Introduction to the Programmer's Workbench. Proc. 2nd Int. Conf. on Software Engineering, pp.164-168 (Oct. 1976).
- (3)* Haseman, W.D., and Whinston, A.B.: Introduction to Data Management. Richard, D. Irwin, INC., Illinois (1977).
- (4) Rockkind, M.J.: The Source Code Control System. IEEE Trans. on Software Engineering, Vol. SE-1, No. 4, pp.364-370 (Dec. 1975).
- (5) Swanson, E.B.: The Dimension of Maintenance. Proc. 2nd Int. Conf. on Software Engineering, pp.492-497 (Oct. 1976).
- (6) Zelkowitz, M.V.: Perspective on Software Engineering. ACM Computing Surveys, Vol. 10, No. 2, pp.197-216 (1978).

* 文献(3)は1980年1月現在翻訳作業が進行しており、1980年春には bit の臨時増刊として発行が予定されている。