

制御用ソフトウェア一貫プログラミングシステム —SPLとその周辺システム—

Industrial Realtime Software Programming System —SPL and Its Related Systems—

近年、計算機制御システムの開発ソフトウェア量は、制御規模の拡大と複雑化に伴って急増化しており、これに伴いソフトウェアを信頼性高く、効率良く開発するための近代的ソフトウェア生産システムの要求が強まってきている。

日立製作所では、これに応ずるために、制御用トップダウンストラクチャードプログラミング言語SPLを開発し効果を挙げてきたが、このたび更に、SPLを中心として、共通テーブル設計支援ツール、プログラム設計技法及び保守ドキュメントジェネレータをはじめとする周辺ツール・技法の開発を進め、ソフトウェア設計から保守までを首尾一貫した思想で支援するプログラミングシステムとして体系化した。

この結果、ソフトウェアの開発作業を円滑に進めることができ、信頼性と保守性の高いソフトウェアを効率良く開発することが可能となった。

森 清三* Kiyomi Mori
林 利弘* Toshihiro Hayashi
野木兼六** Kenroku Nogi
五嶋 将*** Susumu Goshima

1 緒 言

制御用をはじめとするリアルタイムアプリケーションソフトウェアを取り巻く環境は、近年、作成ソフトウェア量の増大、ソフトウェア人口の不足及びシステムの早期立上げ、といった形でますます厳しさを増してきている。

このような環境下において、与えられた期限と費用の中で所期の目的を達成するソフトウェアシステムを、信頼性高く、かつ効率良く構築・保守していくためには、従来の手工業的なソフトウェア作りを近代的なソフトウェア作りに変革していくことが必須である。

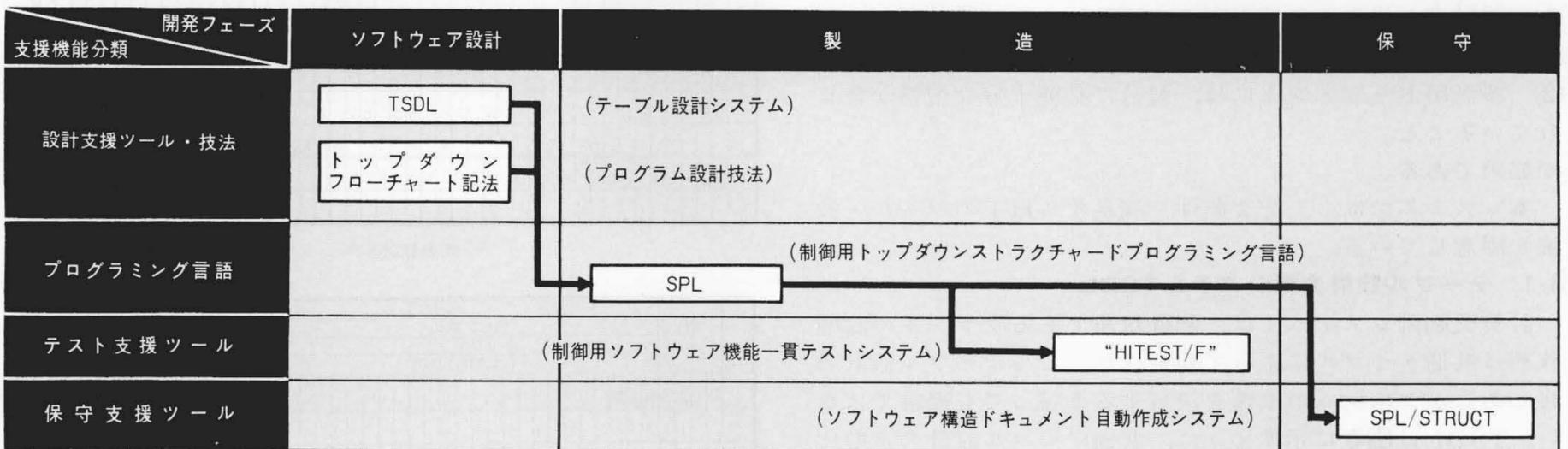
このニーズに応ずるために、昭和51年にソフトウェアエンジニアリング技法に基づいた新しい思想の制御用トップダウンストラクチャードプログラミング言語SPL(Software Pro-

duction Language)²⁾を開発し、以来、今日まで数多くのシステムに適用し大きな効果を挙げてきた。

このたび、SPLの効果を更に拡大することを目的として、設計から保守までを支援する各種ツール・技法を開発し、制御用ソフトウェア一貫プログラミングシステムとして体系化した。

2 システムの体系と構成

本システムの体系は、図1に示すとおり、SPLを中心に
(1) 共通テーブルの設計を支援する言語システムTSDL(Table Specification Language)
(2) プログラムの設計を支援するトップダウンフローチャー



注：略語説明 TSDL(Table Specification Description Language)
SPL(Software Production Language)
"HITEST/F"("Hitachi Integrated Test System/Function")
SPL/STRUCT(SPL/Software Structure Documentation System)

図1 制御用ソフトウェア一貫プログラミングシステムの体系
SPLを中心に各種ツール・技法が開発・整備されており、ソフトウェア設計から保守までを一貫した形で支援できるように体系化されている。

* 日立製作所大みか工場 ** 日立製作所システム開発研究所 *** 日立製作所日立研究所

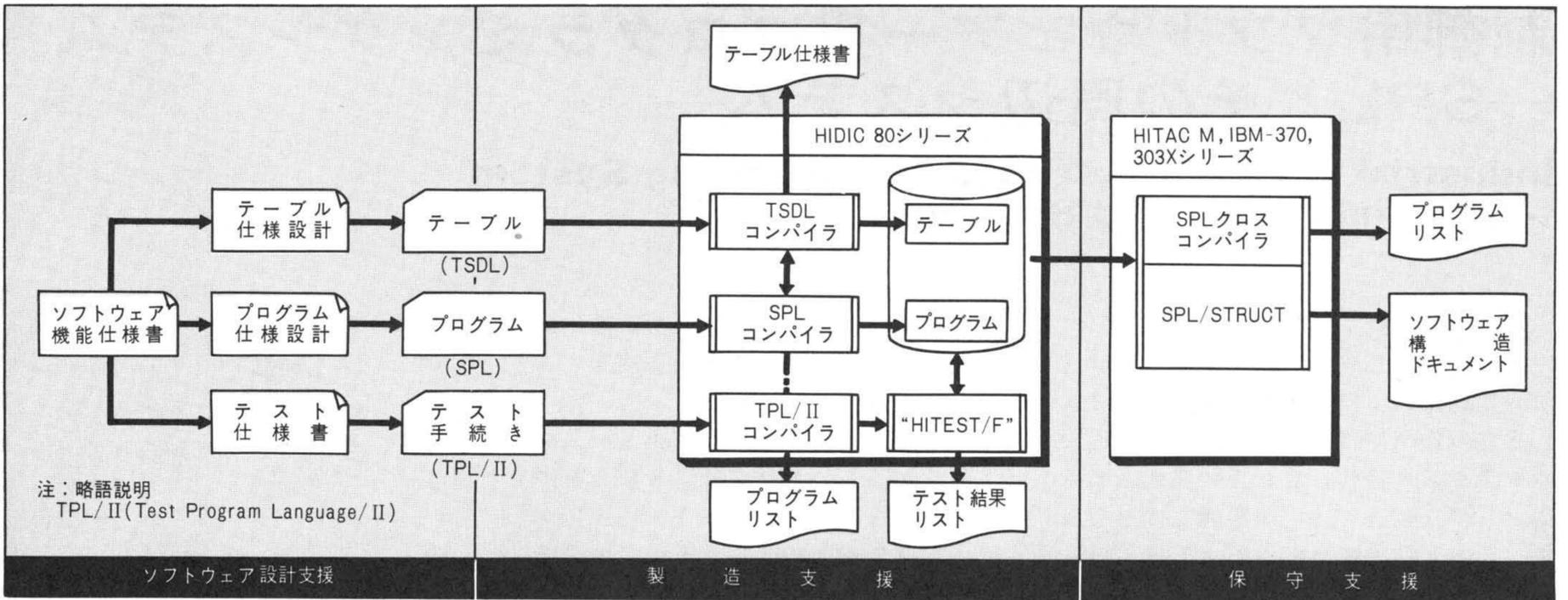


図2 制御用ソフトウェア一貫プログラミングシステムの構成と作業の流れ 本システムの各ツール・技法は、相互に有機的なつながりをもっており、各工程間の作業の流れが円滑に進められるようになっている。

ト記法

(3) 保守ドキュメントを自動生成するシステムSPL/STRUCT (SPL/Software Structure Documentation System)

の設計・保守支援ツール・技法を用意することにより、ソフトウェア設計から保守までを首尾一貫した形で支援できるようにしている。

また、システム構成上の面では、各工程間のギャップを取り除き、作業が円滑に流れるようにすることが重要であるとの認識に立ち、図2に示すように、各ツール・技法間に有機的なつながりをもたせ、各工程間のつなぎ作業が自動的にあるいは容易に行なえるようにしている。

3 ソフトウェア設計支援機能

ソフトウェア設計作業は、システム設計で決定されたソフトウェア機能仕様書に基づき、共通テーブル及びプログラムの設計を行なう作業である。これら作業では、

- (1) 設計者の思考を自然な形でガイドし、かつ設計レビューが容易な設計用ドキュメントが用意されていること。
 - (2) 設計用ドキュメントには、製造に必要な情報が含まれていること。
- が必須である。

本システムでは、上記2条件を満足する以下のツール・技法を用意している。

3.1 テーブル設計支援システムTSDL

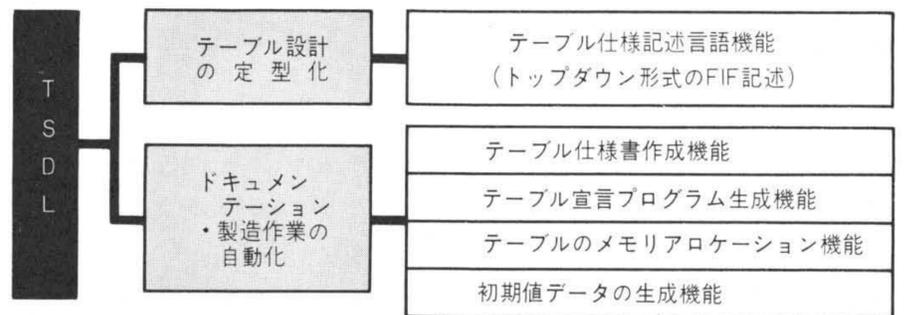
計算機制御システムでは、制御対象であるプラントの設備状態は共通テーブルによって規定され、このテーブル設計の良しあしがシステムの成否を決定すると言っても過言ではない。TSDLは図3に示すように、共通テーブル設計の定型化及びドキュメンテーション・製造作業の自動化をねらいとして、以下の機能をもっている。

(1) テーブル仕様記述言語機能

この言語は、共通テーブル設計者の思考ガイド、個人差解消を図るために、図4に示すようにFIF (Fill In the Form) 記述形式としている。

(2) ドキュメンテーション・製造自動化機能

テーブル仕様書を図形式でラインプリンタに印刷するとともに、テーブル宣言プログラムを自動生成する。テーブル宣



注：略語説明 FIF(Fill In the Form)

図3 テーブル設計支援システムTSDLのねらいと機能 テーブル設計が定型化され、またテーブルに関するドキュメンテーション及び各種製造作業が自動化される。

1	2	3	10	20	30	40
TABLE			フ'ヒン	カンリ	テ-フ'ル	
TNAME			PARTS			
MEMORY			GLOBAL	*BULK	COMMON	DCOMMON LOCAL
			BASED()	EQU(
PURPOSE			シ'ト'ウ	シ'フ'ヒン	シ'ヨウ	ハ'カクノウ
REMARKS			コノ	テ-フ'ル	ニハ	ヨヒ'ヒン
			ヨヒ'ヒン	ノ	シ'ヨウ	ハ'カクノウ
					SPARE	テ-フ'ル
REVISION			80/02/08	モリ	キヨミ	フ'ヒン

(a) 概要記述FIFシート

			LEVEL 3				
			LEVEL 2				
			LEVEL 1				
1	2	3	6	7	34	36	41
			ELEMENT			NAME	AR
DIAGRAM							
	1		フ'ヒン	カンリ	テ-フ'ル	PARTS	1000
	2		フ'ヒン	メイ	シヨウ	NAME	
	3		フ'ヒン	ノ	コスウ	NO	
	4		フ'ヒン	シ'ヨウ	ホウ	INFORM	2
	5		ソウコ	ハ'ン	コ'ウ	SHNO	
	6		ヒン	シツ	ランク	QRANK	
	7		ニン	シキ	ヒヨウ	ID	

(b) テーブル構成記述FIFシート

図4 テーブル仕様記述言語の記述例 テーブル仕様はFIF(空欄記述)形式で記述され、テーブル設計がガイドされるようになっている。なお、白地の部分はユーザーが記述する部分である。

```

1-----2-----3-----4-----5-----6-----7R-----8
DCL OPT(BULK) ;                                00000550
C+-----+-----+-----+-----+-----+-----+-----+-----+ 00000600
C+TABLE :フ*ヒン カリ テ*フル                  + 00000650
C+PURPOSE:シ*ト*ウ*シ*ト ノ フ*ヒン シ*ヨ*カ*ウ オ カクノスル      + 00000700
C+REMARKS:コノ テ*フル ニハ ヨ*ヒ*フ*ヒン ノ シ*ヨ*カ*ウ ハ カクノサレナイ + 00000750
C+      ヨ*ヒ*フ*ヒン ノ シ*ヨ*カ*ウ ハ SPARE テ*フル オ サシヨウ ノ コト  + 00000800
C+REVISION '79/02/08  月 日 年                フ*ヒン テ*ス スク ノ ア*コ*ウ  + 00000850
C+-----+-----+-----+-----+-----+-----+-----+-----+ 00000900
VAR PARTS (1000)                                #/ 00000950
  ( NAME :CHAR(6) ; /# フ*ヒン カリ テ*フル #/ 00001000
  NO :INT ; /# フ*ヒン メイシヨウ #/ 00001050
  INFORM(2) :BIT(8) ; /# フ*ヒン シ*ヨ*カ*ウ #/ 00001100
  ( SHNO :BIT(4) ; /# ソ*コ ハ*コ*ウ #/ 00001150
  GRANK :BIT(4) ; /# ヒンシツ ランク #/ 00001200
  ID :BIT(4) ); /# ニンシ*ヒヨウ #/ 00001250
END ;
END PCENV ;
00001300
00001350
1-----2-----3-----4-----5-----6-----7R-----8
    
```

図5 テーブル宣言プログラムの出力例 コンピュータに入る情報、保守に必要な情報が過不足なく、データ宣言プログラムとして自動出力される。

言プログラムの出力例を図5に示すが、コンピュータに入力される情報及び保守に必要な情報が過不足なく出力されるようになっている。

このほかに、主記憶あるいは補助記憶装置へのテーブルのメモリアロケーション及びそれへの初期値データローディングも自動的に実行されるようになっている。

3.2 プログラム設計支援技法—トップダウンフローチャート記法

これは、処理手続きであるプログラム設計の定型化を目的とした記法であり、次に述べるような特長をもっている。

(1) トップダウン設計

プログラムは、図6(a)に示すように、概略(WHAT)レベルから詳細(HOW)レベルへの段階的詳細化という具合に、トップダウンに設計される。これにより設計者の思考がガイドされ、また第三者による設計レビューも容易となり、プログラム設計の品質が向上する。

(2) SPLとの親和性

SPLはトップダウンコーディングを支援しており、図6(b)に示すように、トップダウンフローチャートで設計された

プログラムは、1対1の形で機械的にコーディングすることができる。また、設計の内容がそのままの形で実プログラムに反映されるので、保守性も向上する。

4 製造支援機能

ソフトウェアの製造を支援するツールは、設計書に基づき容易に製造作業が行なえるツールであることが望ましい。このために、以下の二つの製造支援ツールを用意している。

4.1 制御用トップダウンストラクチャードプログラミング言語SPL

SPLは、ソフトウェアの信頼性・保守性向上を図るために、構造化機能と段階的詳細化機能、テーブル部とプログラム部の分離コンパイル機能、及び共通テーブル・プログラムの一元管理機能に代表される機能を備えた高級言語である。

このSPLの機能により、プログラミング作業は次のように行なわれる。すなわち、ソフトウェア設計がテーブルとプログラムとに分けて行なわれるのに対応して、コーディングもテーブルとプログラムとに分けて行なわれる。このうち、テーブルのコーディングは、テーブル宣言プログラムがTSDLにより自動出力されるので、その作業は不要である。一方、プログラムのコーディングは、SPLの段階的詳細化機能により、トップダウンフローチャートと1対1に対応した形で機械的に行なうことができる。また、これによりプログラムのリストもトップダウンな形でリストアップされ、見やすく保守性が高い。

SPLは、このほかに、プログラムの構造を解析し整形する自動インデント機能も用意し、プログラム修正の容易化を図っている。

4.2 制御用ソフトウェア機能一貫テストシステム“HITEST/F”

通常、機能テストはプログラム単体及び組合せ、タスク組合せ、並びに総合といった過程に分けて実施される。“HITEST/F”

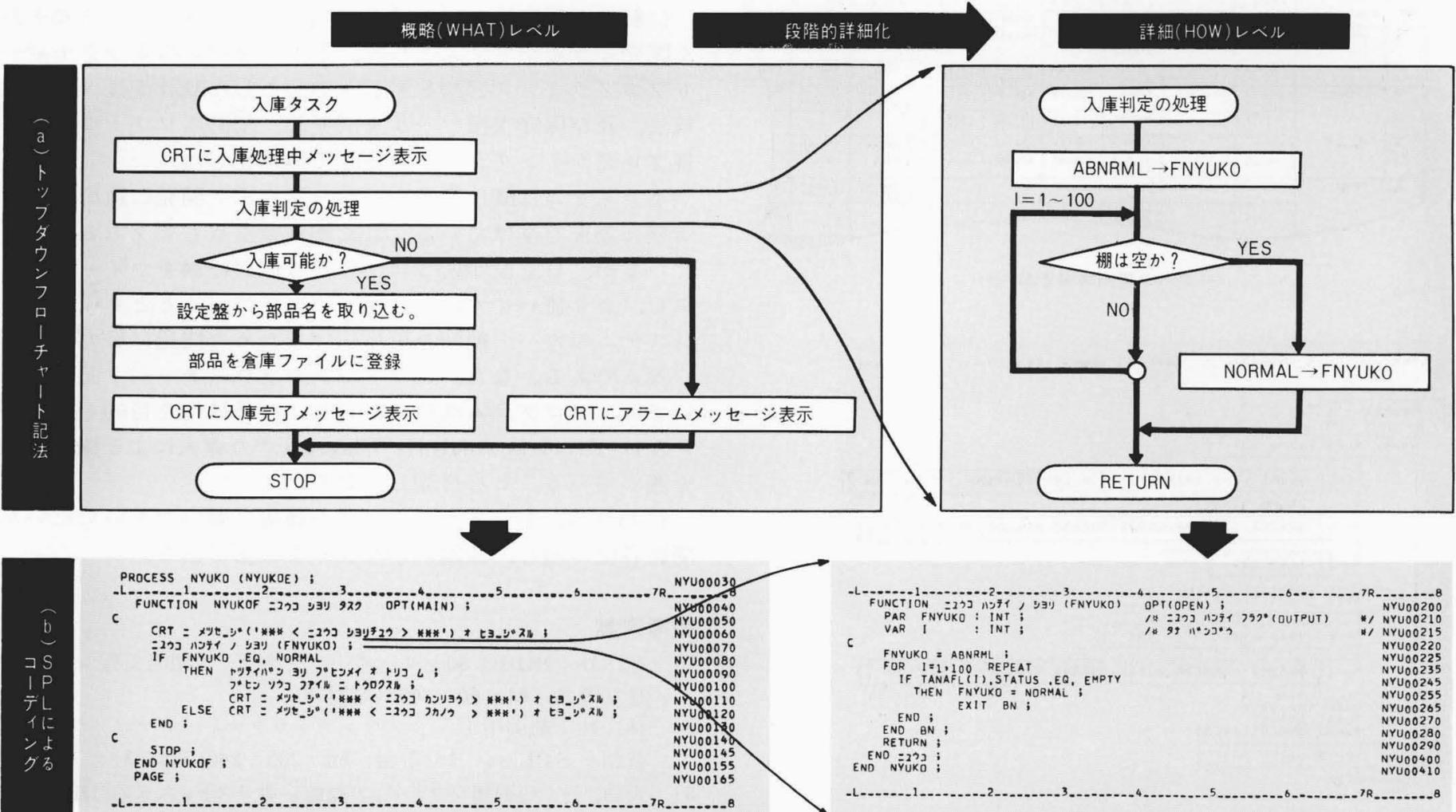


図6 トップダウンフローチャート記法によるプログラムの設計とSPLによるコーディング プログラムはトップダウンフローチャート記法により段階的に設計され、しかもそれに対応してSPLによりトップダウンにコーディングされる。

表1 SPL/STRUCTが出力するドキュメントの構成と種類 システム全体を鳥観するドキュメントからプログラム単体のドキュメントまでトップダウンに体系化されている。

区分	ドキュメント名称	ドキュメントの内容
システム全体	表紙	システムの名称及びドキュメントの目次
	凡例	ドキュメントに使用される記号, 略号の意味説明
	タスク間制御関係図	システムを構成する全タスクに対して, それらの間の制御関係を示す。
	タスク⇔共通テーブル相互参照図	システムを構成する全共通テーブルに対して, それらを参照するタスクの参照状況を示す。
	タスク⇔共通ルーチン相互参照図	システムで共通に使用されるルーチンに対して, それらをコールしているタスク群を一覧する。
	共通テーブル構造図	共通テーブルの詳細仕様(構造図, 意味など)を示す。
プログラム単体	タスクスケルトンチャート	各タスクの処理の概要をスケルトン形式で示す。
	プログラム処理仕様	プログラムのトップダウンリスト(SPLコンパイラリスト)

F” (“Hitachi Integrated Test System/Function”)³⁾ は, テスト手続き記述言語TPL/II (Test Program Language/II) を用意し, テストの全過程に対して一貫した形で支援できるようにになっている。これにより, SPLで作成されたソフトウェアを効率良くテストすることができる。

5 保守支援機能

計算機制御システムは, 設置後10年以上にわたって使用されるのが普通であり, この間種々のソフトウェアの改造・拡張が行なわれる。また, 保守担当者は必ずしもそのソフトウェアの開発担当者であるとは限らない。

したがって, ソフトウェアの保守を信頼性高く, 効率良く行なうためには, システム全体の動きが把握でき, かつ100%正確な保守ドキュメントが必要である。このために, 以下のソフトウェア構造ドキュメントを自動生成するシステムSPL/STRUCTを用意し, 保守作業の支援を図っている。

5.1 ドキュメントの構成と特長

SPL/STRUCTで自動生成されるドキュメントの構成と種類は, 表1に示すとおりであり, 次に述べるような特長がある。

- (1) 概略レベルから詳細レベルへとトップダウンに体系化されており, 第三者に分かりやすい。
- (2) タスク⇔タスク間, タスク⇔共通テーブル間のリンケージなど, システム全体を鳥観する横断的な保守情報が得られる。
- (3) ドキュメントは, すべてソースプログラムを基に作成されるため正確である。また, SPLで一元的に管理された共通テーブル・手続き仕様からドキュメントが作成されるので, ドキュメント間の矛盾がない。
- (4) 図表形式であり, 視覚化されている。

5.2 出力ドキュメントの例

出力ドキュメントの例の一部として, タスク間制御関係図, タスクスケルトンチャートを図7に示す。

本例では, ドキュメントの印刷は和文で行なわれているが, 英文で印刷することも可能である。

6 結 言

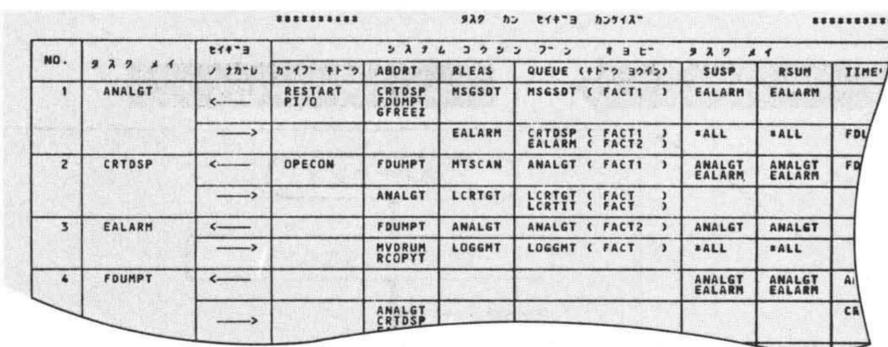
信頼性と保守性の高い計算機制御用ソフトウェアを効率良く開発できるシステムとして, トップダウンストラクチャードプログラミング言語SPLを中心とした設計支援ツール・技法, 及び保守支援ツールを開発し, 制御用ソフトウェア一貫プログラミングシステムとして体系化した。

本システムは既に数多くのソフトウェア開発に適用され, 所期の効果を挙げている。まだ不十分な点も多々あると考えているが, 日立製作所では, 今後も適用経験をフィードバックし, より使いやすいシステムに育てていくとともに, このシステムのカバー範囲を更に広げるための開発に努力していく考えである。なお, プログラムのフローチャート記法については, プログラム構造のいっそうの単純化を目的として, PAD(Problem Analysis Diagram)⁴⁾の導入による機能拡張を進めていることを付記しておく。

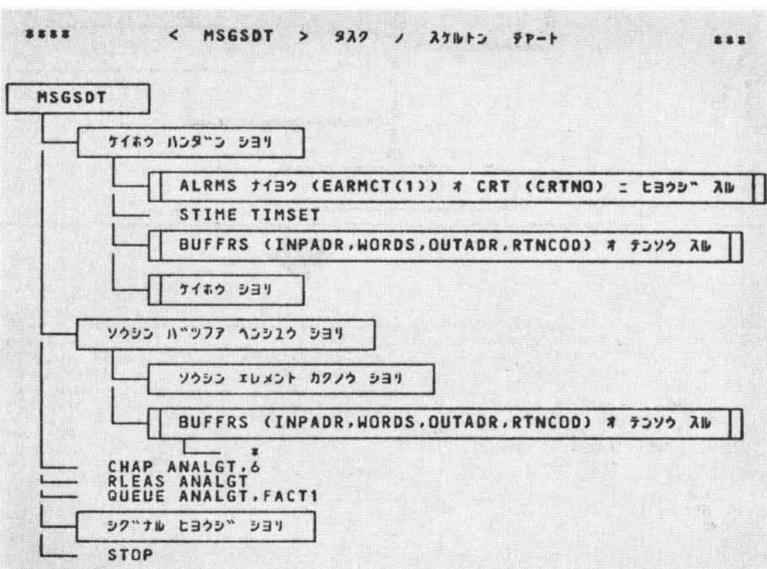
終わりに, 本システムの開発と適用に御協力をいただいた関係各位に対し, 深謝の意を表わす次第である。

参考文献

- 1) 林, 外: HIDIC 80シリーズ ソフトウェア開発支援システム, 日立評論, 61, 603~606(昭54-8)
- 2) 林, 外: 制御用トップダウンストラクチャードプログラミング言語—SPL—, 日立評論, 60, 235~240(昭53-3)
- 3) 大島, 外: 制御用ソフトウェア機能一貫テストシステム“HITEST/F”, 日立評論, 62, 893~898(昭55-12)
- 4) 二村, 外: プログラムの木構造化図面“PAD”, 日立評論, 62, 871~874(昭55-12)



(a) タスク間制御関係図の例



(b) タスクスケルトンチャートの例

図7 出力ドキュメントの例 システムを構成する全タスクに対して, それらの間の制御関係が一覧できる。また, 各タスクの処理の概要が把握できる。