U.D.C. 681.323.06:519.68

HIDIC 80シリーズ ソフトウェア開発支援システム

Industrial Realtime Software Development Support Systems for HIDIC 80 Series

林　利弘* *Hayashi Toshihiro*
平井浩二* *Hirai Kôji*
野木兼六** *Nogi Kenroku*
福岡和彦** *Fukuoka Kazuhiko*
五嶋　将*** *Goshima Susumu*

計算機制御システムの開発ソフトウェア量は，制御規模の拡大とマンマシン処理の高度化に伴って急増しており，ソフトウェアの生産性，信頼性及び保守性の向上は，解決を急がねばならない最も重要な課題の一つとなっている。

日立製作所では，従来から各種ソフトウェアツールの開発を通してこの課題の解決に取り組んできたが，HIDIC 80シリーズでは更に一歩進めて，ソフトウェアエンジニアリングの観点から各種ツールの開発を進め，これらツールを計算機制御システムの設計段階から保守段階までを首尾一貫した思想で支援するソフトウェア開発支援システムとして体系化し，このたび完成した。

1 緒　言

近年，計算機制御システムはマルチコンピュータ化や機能分散ネットワーク化により制御規模が急速に拡大し，カラーCRT(Cathode Ray Tube)を用いたマンマシン処理も高度化しているが，この結果，システム当たりの開発ソフトウェア量は数年前に比べて3～4倍と急増している。

このような環境下にあって，与えられた期限と費用の中で所期の目的を達成するシステムを信頼性高く構築し，効率よく保守していくには従来の手工業的なソフトウェア作りを近代工業的なソフトウェア作りに変えていくことが必須の課題となっている。

HIDIC 80シリーズではこの課題に応ずるために，近年発展の著しいソフトウェアエンジニアリング手法を駆使した各種ツールを一貫した思想のもとで開発し，ソフトウェア開発支援システムとして体系化した。

2 ソフトウェア開発支援システムの体系

ソフトウェア開発支援システムは**図1**に示すように，開発初期段階の事前性能テストから開発完了前の事後テスト及び保守に至るまでの全開発課程に対し，幅広く体系的に支援するもので，制御用アプリケーションソフトウェアの開発を品質高く，かつ効率よく行なえるようになっている。

このシステムの大部分は，HIDIC 80シリーズのオペレーティングシステムであるTSES(Time Sharing Executive System)の下で，リアルタイム処理の空き時間を用いてバッチモード若しくは会話モードのいずれかで利用でき，一部はHITAC Mシリーズ若しくはIBM370シリーズの下でのクロスシステムとして利用できる。

3 ソフトウェア設計・製作支援機能

3.1 設計支援ツール

計算機制御システムでは，各種ソフトウェア構造はテーブルによって通常規定され，テーブル設計はソフトウェア設計の最初に行なわねばならない重要な作業である。テーブル仕様記述言語(Table Specification Description Language: TSDL)ではこの設計作業をFIF(Fill in the Form)シートにより標準化するとともに，テーブル仕様書の作成やテーブルに関する各種プログラミング作業を自動化し，テーブル設計作業の高信頼化と高効率化を可能にしている。

3.2 手続き向きプログラミング言語

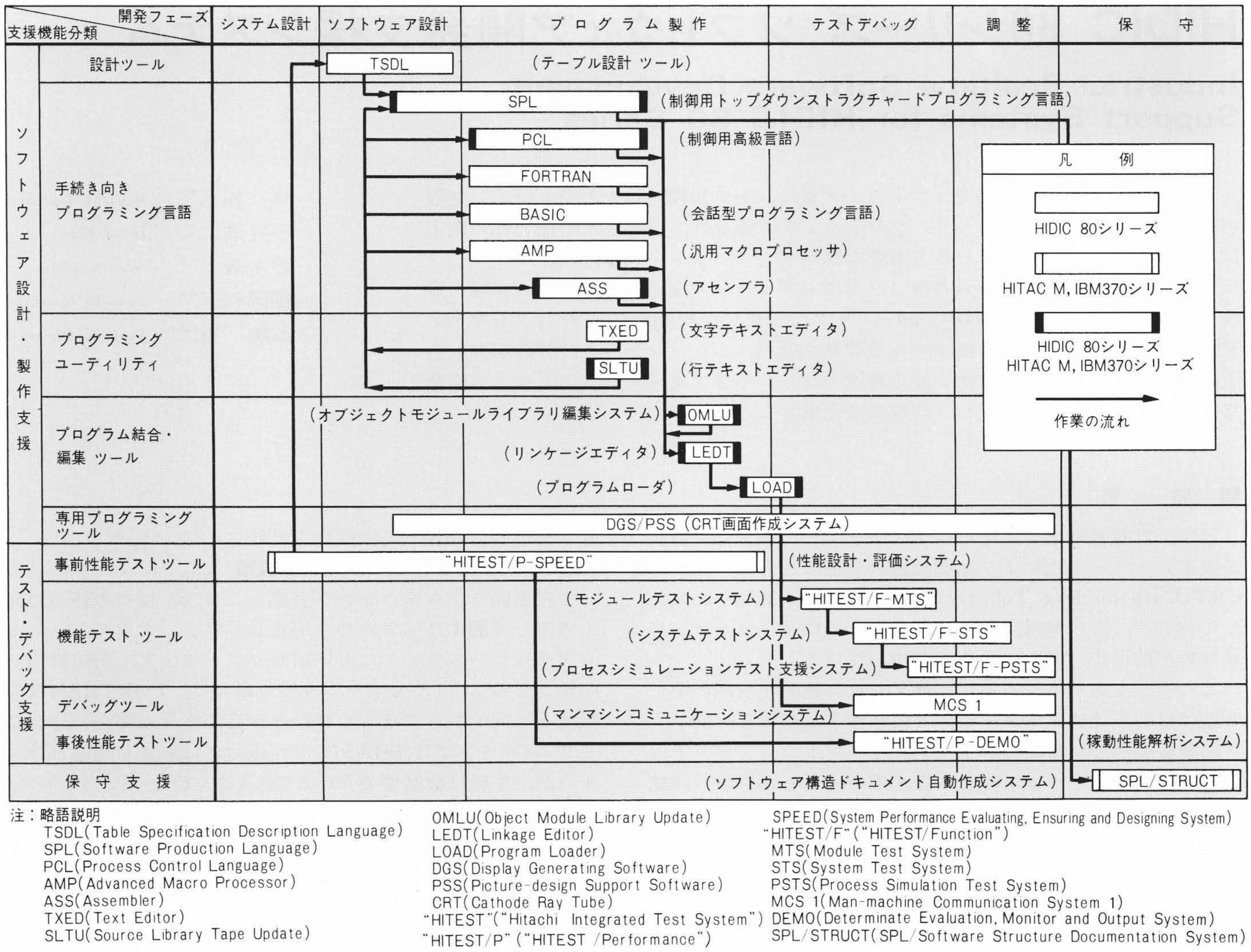
手続き向きプログラミング言語としては，種々の使用局面を考慮して**図1**に示す六つを用意している。すなわち，アセンブラはハードウェアに近い制御ソフトウェアや特に性能が問題となるソフトウェアを作成する場合に，FORTRANは他機種との間でプログラムの移行性が必要となる場合に，会話型プログラミング言語BASICは性能的にはあまり問題にならないが，手軽に会話型でプログラミングを行ないたい場合に，汎用マクロプロセッサ(Advanced Macro Processor：AMP)は応用分野別に問題向き言語を作り，各種手続きをマクロ定義したり，データベースを生成したりする場合にそれぞれ有用となる。

制御用高級言語(Process Control Language：PCL)と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(Software Production Language：SPL)はいずれも制御用アプリケーションソフトウェアを高級言語ベースで能率よく開発できるように，特別に設計開発された言語システムである。このうちPCLは，FORTRANをベースに制御用に必要な機能を拡張追加し，性能的にも十分な最適化技法を取り入れた言語として昭和47年に開発され，HIDIC 80シリーズでは各種の適用経験をもとに幾つかの機能上，性能上の改良を加え，いっそう使いやすくしている。一方，SPLはPCLのもつ各種制御用機能は包含しながら，更にソフトウェアの本質的高信頼性，高保守性及び標準化のしやすさといったソフトウェアエンジニアリング的な機能を，大幅に強化した新しい思想の言語である[1]。**図2**にSPLの主な機能を，**図3**にSPLのコーディング例を示す。

3.3 プログラミングユーティリティ

プログラミング作業を便利にするユーティリティとして，2種のソーステキストエディタがある。一つは会話型プログラミングモードの下で，文字単位のテキスト編集ができる文字テキストエディタ(Text Editor：TXED)であり，もう一つはバッチ型プログラミングモードの下で，行単位のテキスト編集ができる行テキストエディタ(Source Library Tape Update：SLTU)である。これらにより，ソースプログラムの修正は非常に簡単になる。

* 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



注：略語説明
TSDL(Table Specification Description Language)
SPL(Software Production Language)
PCL(Process Control Language)
AMP(Advanced Macro Processor)
ASS(Assembler)
TXED(Text Editor)
SLTU(Source Library Tape Update)
OMLU(Object Module Library Update)
LEDT(Linkage Editor)
LOAD(Program Loader)
DGS(Display Generating Software)
PSS(Picture-design Support Software)
CRT(Cathode Ray Tube)
"HITEST"("Hitachi Integrated Test System")
"HITEST/P"("HITEST /Performance")
SPEED(System Performance Evaluating, Ensuring and Designing System)
"HITEST/F"("HITEST/Function")
MTS(Module Test System)
STS(System Test System)
PSTS(Process Simulation Test System)
MCS 1(Man-machine Communication System 1)
DEMO(Determinate Evaluation, Monitor and Output System)
SPL/STRUCT(SPL/Software Structure Documentation System)

図1　HIDIC 80シリーズ ソフトウェア開発支援システム体系　このシステムは，システム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おり，この体系に沿って各種ツールが開発・配備されている。

3.4　プログラム結合・編集ツール

プログラミング言語によって作られるプログラムは，通常，デックと呼ばれる比較的小さな機能単位(オブジェクトモジュール)である。これを，リアルタイムモニタ下で同時併行処理される，より大きな機能単位(ロードモジュール)として結合・編集するツールがリンケージエディタ(Linkage Editor：LEDT)であり，単純構造から複雑なオーバレイ構造までの多様な構造を定義できるとともに，結合・編集する個々のモジュールに対しても細かい選択ルールの指示ができる。そして，結合・編集するモジュールのライブラリへの登録・削除を行なうツールがオブジェクトモジュールライブラリ編集システム(Object Module Library Update：OMLU)である。また，プログラムローダ(Program Loader：LOAD)は，ロードモジュールに対してシステム共通変数のアドレスや共通定数の値を決定して所定のエリアに格納するツールであるが，HIDIC 80シリーズでは，特に，この共通変数のアドレス決定や格納エリアの決定を，ローダ自身のメモリ管理機構によって自動的に行なうことができ，また，オンラインでの保守やデバッグ時に必要となるマンマシンコミュニケーションシステム(Man-Machine Communication System 1：MCS 1)の情報もすべて自動的に作成する。

3.5　専用プログラミングツール

これまでに述べてきたツールは，いずれも対象を特定しない汎用ツールであるが，対象を特定できれば専用ツール化でき，より簡易なプログラミングが可能となる。CRT画面作成システム(Display Generating Software/Picture-design Support Software：DGS/PSS)は，CRT表示プログラミングという対象に絞って，複雑なカラー図形の作画データ作成とオンライン表示処理を，カラーCRTを用いて会話型で簡便に行なうツールである[2]。主な機能とこのツールを用いた場合のプログラミング形態を**図4**に示す。

4　テスト・デバッグ支援機能

ソフトウェア開発作業の半分以上が，性能上，機能上の不具合点(バグ)を発見するためのテスト作業と，発見されたバグの原因を追求し対策を行なうデバッグ作業とに費されている。

したがって，テスト・デバッグ作業の効率を上げることは全体の生産性向上に大きく寄与するとともに，完成したソフトウェアの信頼性向上にも直結する。

このような観点から開発したのが，制御用ソフトウェア一貫テストシステム("Hitachi Integrated Test System"："HITEST")であり(**図5**)，またMCS 1である。

4.1　事前性能テストツール

計算機制御システムはその応答性が重要であるが，通常，システムの応答性が把握できるのはソフトウェア開発の終了

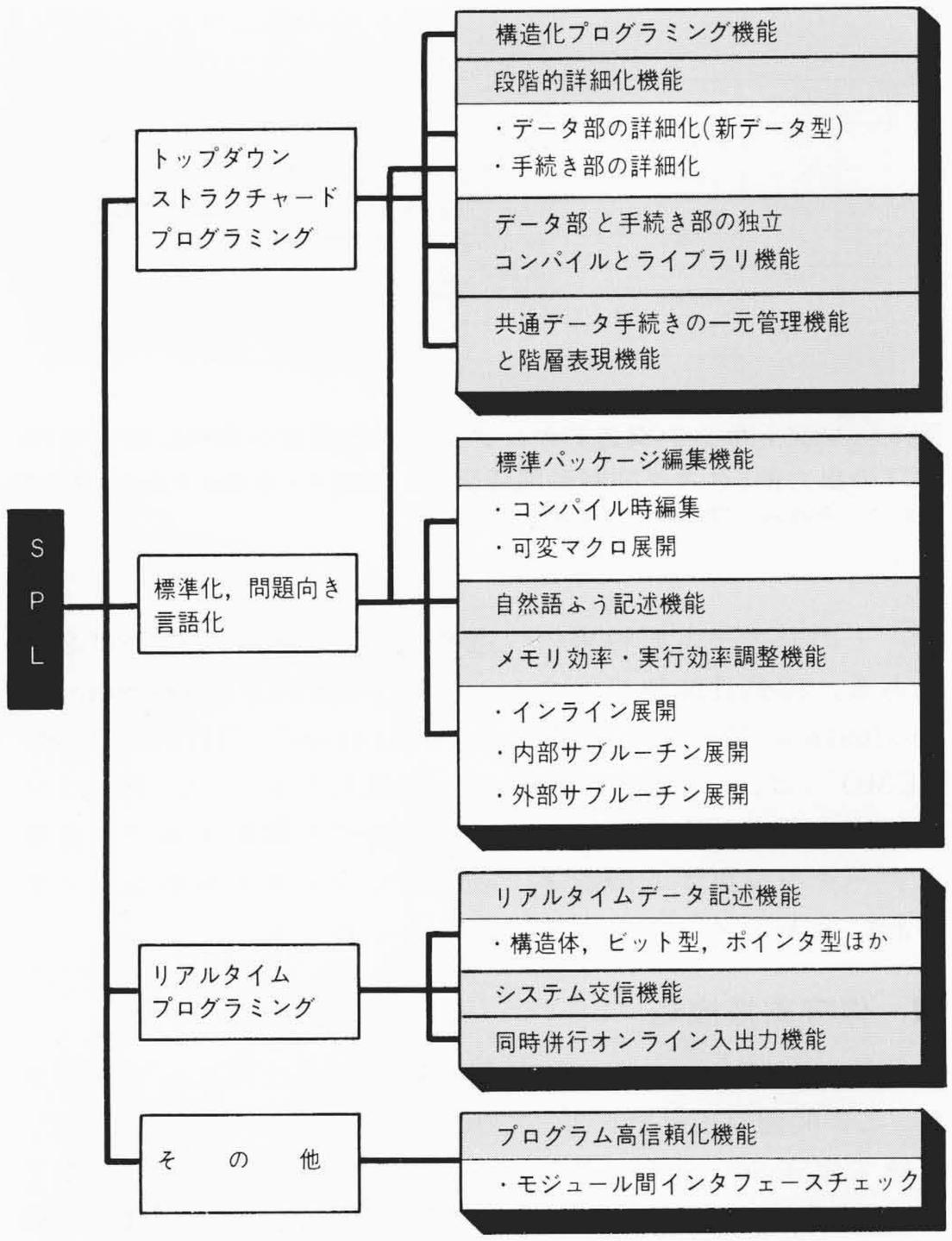


図2　制御用トップダウンストラクチャードプログラミング言語SPLのねらいと機能　SPLは段階的詳細化機能により，トップダウンプログラミング言語と標準化(問題向き)言語を表裏一体のものとして扱うことができる。

```
 PAGE 00006      HIDIC-80  SPL   LIST

MODULE    LNO.    SPL  SOURCE  STATEMENTS  ( FILE INDEX H-80 SPL V2-T2 )  ( 1978  8/20 ) ID SEQ.

          -----  -L-------1---------2---------3---------4---------5---------6---------7R--------8

                 @PAGE
                 @COMPILE MODE @
                 @GENERATE
          00001   PROCESS   NYUKO (NYUKOE) ;                                                    NYU00100
          -----  -L-------1---------2---------3---------4---------5---------6---------7R--------8
NYUKO     . 00002      FUNCTION   NYUKOF ﾆｭｳｺ ｼｮﾘ ﾀｽｸ      OPT(MAIN) ;                          NYU00110
NYUKO     . 00003          RESERVE   TANAFL,ZAIKFL,TAZAFL ;                                   NYU00120
NYUKO     . 00004          CRT ﾆ ﾒｯｾｰｼﾞ('*** ｸ ﾆｭｳｺ ｼｮﾘﾁｭｳ > ***') ｦ ﾋｮｳｼﾞｽﾙ ;                NYU00130
NYUKO     . 00005          ﾆｭｳｺ ﾊﾝﾃｲ ﾉ ｼｮﾘ  (FNYUKO) ;                                    NYU00140
NYUKO     . 00006          IF   FNYUKO .EQ. NORMAL                                           NYU00150  1
NYUKO     . 00007               THEN   ｾﾂﾃｲﾊﾟﾗﾒﾝ ﾖﾘ ﾄﾘｺﾑ (RTURNB=BEND)  ;                    NYU00160  .
NYUKO     . 00008                      ﾆｭｳｺ ｼｮﾘ ｦ ｽﾙ ;                                      NYU00170  .
NYUKO     . 00009                      ﾆｭｰｺ ｼｽﾃﾑ ｦ ｼｭｯﾘｮｸｽﾙ ;                              NYU00180  .
NYUKO     . 00010                      CRT ﾆ ﾒｯｾｰｼﾞ('*** ｸ ﾆｭｳｺ ｶﾝﾘｮｳ > ***') ｦ ﾋｮｳｼﾞｽﾙ ;  NYU00190  .
NYUKO     . 00011               ELSE   CRT ﾆ ﾒｯｾｰｼﾞ('*** ｸ ﾆｭｳｺ ﾌｶﾉｳ > ***') ｦ ﾋｮｳｼﾞｽﾙ ;  NYU00200  .
NYUKO     . 00012          END BEND ;                                                        NYU00210  1
NYUKO     . 00013          FREE   TANAFL,TAZAFL,ZAIKFL ;                                     NYU00220
NYUKO     . 00014          STOP ;                                                            NYU00230
NYUKO     . 00015      END NYUKOF ;                                                          NYU00240
NYUKO     . 00016     TITLE '**********   ﾆｭｳｺ ｦ ｵｺﾅｳ ｼｮﾘ   **********'  ;                  NYU00250

          -----  -L-------1---------2---------3---------4---------5---------6---------7R--------8
```

図3　SPLのコーディング例　SPLは，プログラムを自然語ふうな記述を用いてトップダウンな形に作成でき，共通変数についてはシステム全体で一元管理できるので，各モジュール内での宣言は不要である。

段階時点であり，この時点で問題点が発生するとその対策には多大の工数を必要とする。したがって，システムの応答性に問題がないか否かを，設計の初期段階でテストできれば非常に効果が大きい。“HITEST”の中の性能設計評価システム(System Performance Evaluating, Ensuring and Designing System：SPEED)は，このような要求に応ずるために開発したものであり，設計初期段階で得られる簡単なデータ(プログラム-テーブル関係，メモリ容量など)から，ハードウェア利用形態の主要項目(テーブル配置，プログラム配置及び起動タイミング)を自動的に決定する最適化設計支援機能と，その結果を用いてシミュレーション手法によりシステムの応答時間と余裕度を予測する事前評価機能から成っている[3]。

4.2　機能テスト支援ツール

通常，機能テストはデック単体及び組合せ，タスク組合せ，総合といった過程に分けて実施される。機能テストシステム(“HITEST /Function”: “HITEST /F”)は，これらのテスト過程に対応して三つのサブシステム，すなわちモジュールテ

No.	分　類	機　能
1	作　画	(1) ドットパターン作成機能
		(2) 図形反転機能
		(3) 図形呼出機能
		(4) 図形変色機能
		(5) 図形複写機能
		(6) 図形移動機能
		(7) 文字埋込機能
		(8) エリアの移動機能
		(9) エリアの拡大・縮小機能
2	登録・削除	(1) 図形登録・削除機能
		(2) 画面登録・削除機能
3	FG画面 変化仕様定義	(1) 変化場所定義機能
		(2) 図形変化仕様定義機能
		(3) メッセージ入力仕様定義機能
		(4) メッセージ出力仕様定義機能
		(5) バーグラフ仕様定義機能
		(6) トレンドグラフ仕様定義機能
4	ユーティリティ	(1) 図形・画面のライブラリ化機能
		(2) 図形・画面情報の印字機能
		(3) 画面データオブジェクトの出力機能
		(4) 画面データのテスト表示機能

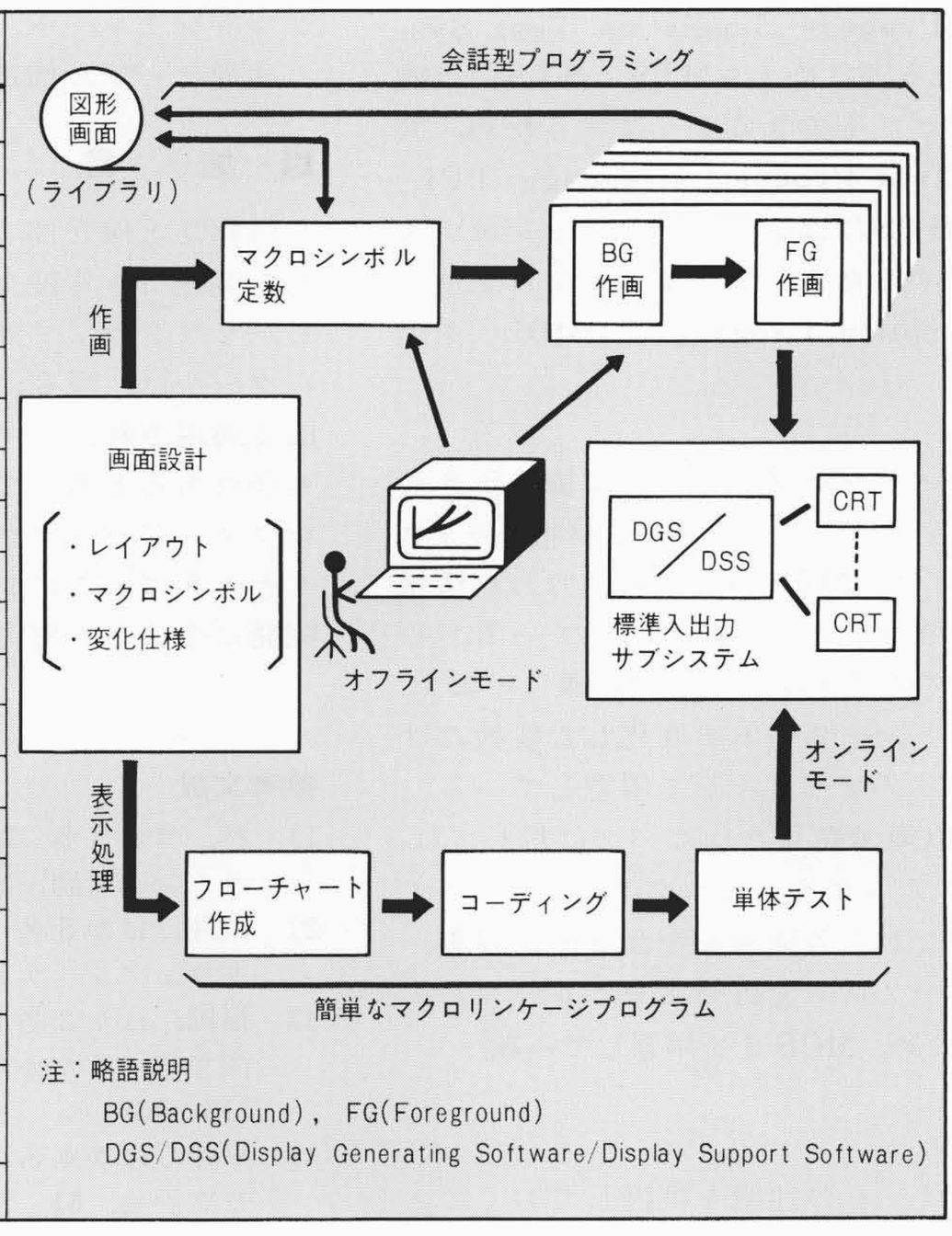


図4　CRT画面作成システム(DGS/PSS)の機能と作業形態　DGS/PSSを用いれば，作画作業と表示処理作業は併行に行なうことができ，しかも作業は非常に簡易化される。なおDGS/PSSによって作成されたデータ及びプログラムはDGS/DSSによってオンライン表示される。

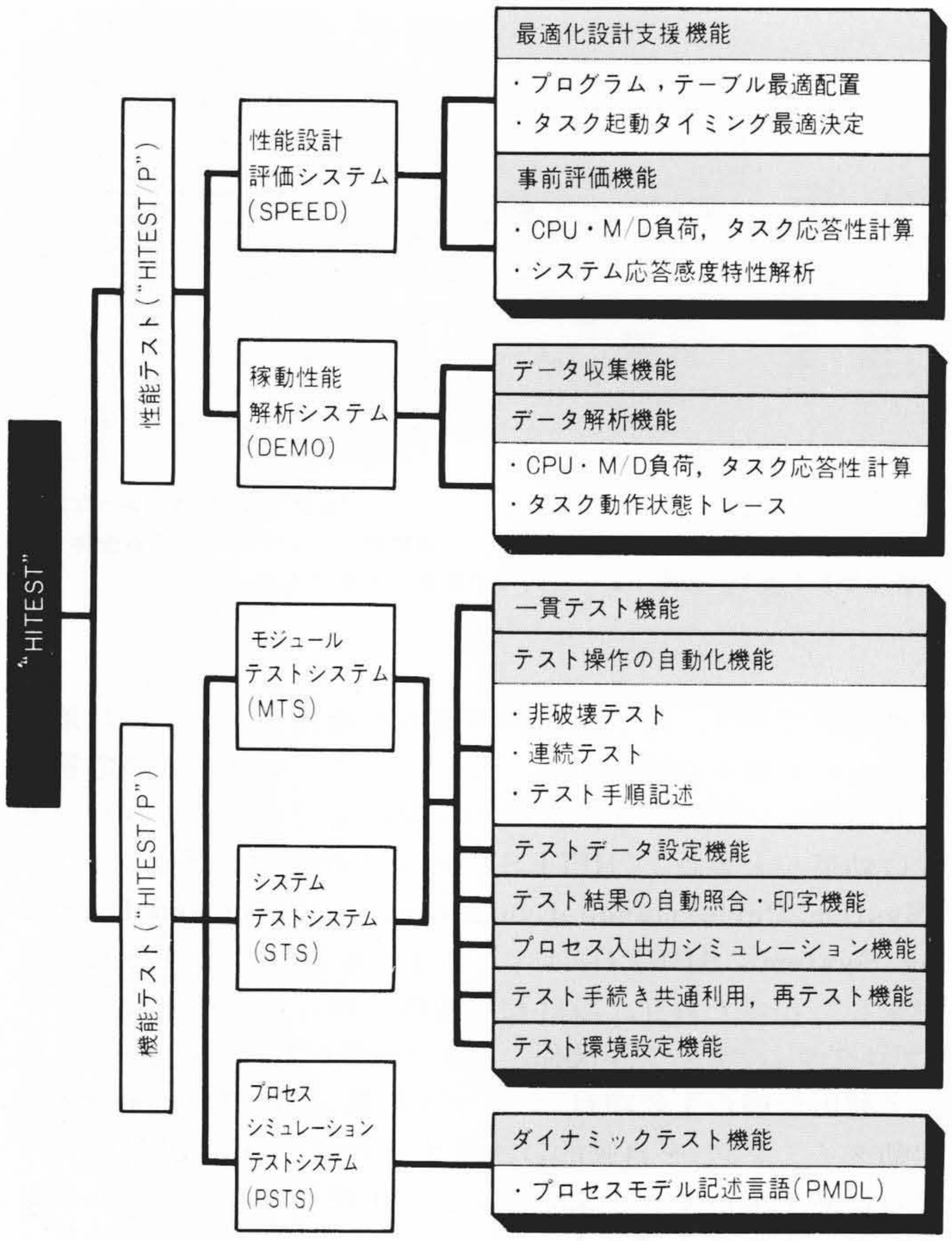


図5　制御用ソフトウェア一貫テストシステム"HITEST"の構成と機能　"HITEST"は，性能テストと機能テストをそれぞれ一貫した形で支援するシステムで，特に，機能テストはテスト手続きを記述するための専用言語TPLをもっている。

ストシステム(Module Test System：MTS)，システムテストシステム(System Test System：STS)及びプロセスシミュレーションテストシステム(Process Simulation Test System：PSTS)から成っている。このうちMTSとSTSには機能テストを一貫した形でサポートできるようにするため，共通のテスト手続き記述言語(Test Program Language：TPL)を用意している。また，PSTSにはシミュレーションモデルを容易に作成できるようにするため，プロセスモデル記述言語(Process Model Description Language：PMDL)を用意している。

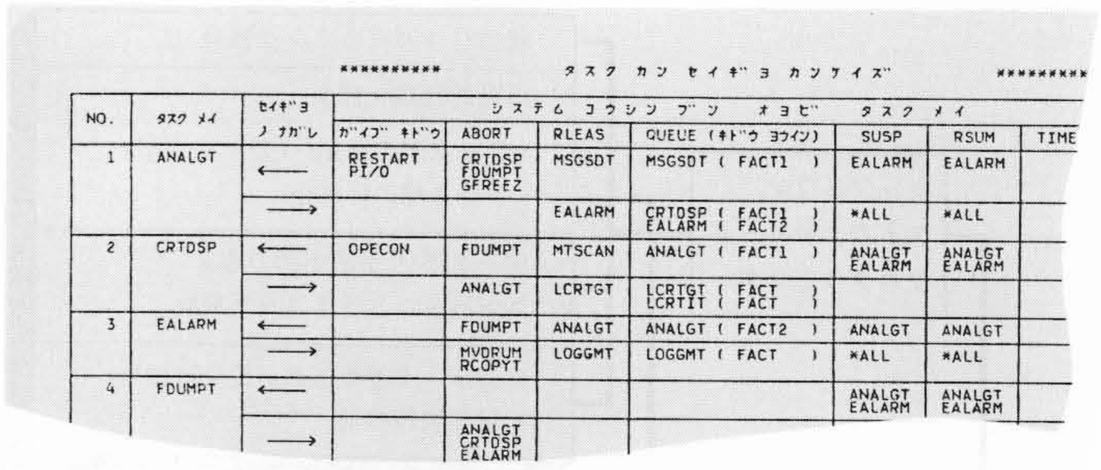
4.3　デバッグツール

SPLプログラムでは，ソースプログラムがよく構造化されており，信頼度レベルが高いので，デック単体及び組合せテストで発見されたバグに対しては，MTSが収集・出力した情報を用いて机上デバッグを行なうことで十分と考えている。しかし，PCLやFORTRANプログラムは構造化が難しいので，高級言語レベルでの文番号トレースや値の変化した変数だけをトレースする機能をデバッグツールとして用意している。

また，タスク組合せテスト以降で発見されたバグに対しては，STSの収集情報に加えてリアルタイムモニタ下でのタスクの動きを総合的にチェックしながら各種情報を会話的に収集，変更し，また，タスク制御マクロの発行順序をトレースしてデバッグを行なう必要があるため，MCS 1を用意している。

4.4　事後性能テストツール

完成したシステムに対して，その性能を把握しておくこと



タスク カン セイギョ カンケイズ

NO.	タスク メイ	セイギョ ノ ナガレ	システム コウシン ブン オヨビ タスク メイ					
			カイブ キドウ	ABORT	RLEAS	QUEUE (キドウ ヨウイン)	SUSP	RSUM	TIME
1	ANALGT	←	RESTART PI/O	CRTDSP FDUMPT GFREEZ	MSGSDT	MSGSDT (FACT1)	EALARM	EALARM	
		→			EALARM	CRTDSP (FACT1) EALARM (FACT2)	*ALL	*ALL	
2	CRTDSP	←	OPECON	FDUMPT	MTSCAN	ANALGT (FACT1)	ANALGT EALARM	ANALGT EALARM	
		→		ANALGT	LCRTGT	LCRTGT (FACT) LCRTIT (FACT)			
3	EALARM	←		FDUMPT	ANALGT	ANALGT (FACT2)	ANALGT	ANALGT	
		→		MVDRUM RCOPYT	LOGGMT	LOGGMT (FACT)	*ALL	*ALL	
4	FDUMPT	←					ANALGT EALARM	ANALGT EALARM	
		→		ANALGT CRTDSP EALARM					

図6　ソフトウェア構造ドキュメント自動作成システムSPL/STRUCTの出力例(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が一覧できる。

は，そのシステムの将来の改造や拡張を計画する場合に必要である。稼動性能解析システム("HITEST/P-Determinate Evaluation, Monitor and Output System"："HITEST/P-DEMO")は，このような観点から開発したもので，各種リソースの負荷率や応答性，タスク動作特性を解析するのに必要なデータを収集する機能と，収集したデータを解析し見やすい形にドキュメント化する機能をもっている。

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，この間種々のソフトウェアの改造・拡張が行なわれるので，正確で保守のしやすいドキュメントを作成することが重要である。SPLプログラムは本質的に保守性は高いが，HIDIC 80シリーズではこのような観点から更に保守性を追求し，システム全体を鳥観できる保守ドキュメントをSPLプログラムから自動的に作成できるソフトウェア構造ドキュメント自動作成システム(SPL/Software Structure Documentation System：SPL/STRUCT)を用意している(**図6**)。このシステムが作成するドキュメントとしては，(1)タスク間制御関係図，(2)タスク-システム共通テーブル相互参照図，(3)システムスケルトンチャート，(4)タスクスケルトンチャート，(5)システム共通テーブル構造図，(6)その他がある。

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生産性高く開発できる種々のツールを，一貫した思想のもとで開発し，ソフトウェア開発支援システムとして体系づけた。これらツールは，現在，アプリケーションソフトウェアの開発に広く適用され，いずれも好評を得ているが，まだ不十分な点も多々あると考えている。日立製作所では，今後も適用経験をフィードバックさせ，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今後とも努力していく考えである。

参考文献

1)　林，ほか3名：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）
2)　政井，ほか5名：プロセスディスプレイ装置の画面作成ソフトウェアシステム"DGS"，日立評論，**60**，613～618（昭53-8）
3)　福岡，ほか3名：コンピュータ・システムの応答時間を短時間で求めるための一手法，日経エレクトロニクス，1978.8.21，116～133
4)　平井，ほか4名：HIDIC 80シリーズ基本制御ソフトウェア，日立評論，**61**，599～602（昭54-8）