

プログラム自動生成システム“SDL/PAD”

Automatic Program Generation System “SDL/PAD”

計算機のプログラム開発の効率向上の重要課題に、作業の機械化がある。現在、プログラム仕様書の作成・修正及び計算機言語への変換は、機械的作業が多いにもかかわらず、すべて人手で行なわれており、生産性向上、プログラム品質向上の阻害要因となっている。プログラム自動生成システムSDL/PADは、これらの問題の解決を目的とした、計算機応用の作業自動化システムである。

SDL/PADシステムは、構造化設計などの優れた設計技法に基づく新しい仕様記述法を用いて記述された仕様を入力とし、FORTRAN S-PL/H, PL/Mなどの計算機言語によるプログラム及び図式表現した仕様書を、自動的に作成する。本システムにより、従来は人手によっていた仕様から計算機言語への変換作業は不要となり、プログラム作成効率が従来に比べ2倍以上高められるとともに、仕様書とプログラムとの一体化を図ることができる。

前沢裕行* *Hiroyuki Maezawa*
 斎藤和正* *Kazumasa Saitô*
 小林正和* *Masakazu Kobayashi*
 二村良彦** *Yoshihiko Futamura*

1 緒言

現在、プログラム開発は図1に示すように、(1)設計者ないしはプログラマが、プログラムの仕様書を作成する、(2)プログラマが仕様書から、計算機言語を用いてプログラム文を作成する、(3)プログラム文を、計算機を用いて機械語のプログラムに変換する、の諸過程を経て通常進められている。これらの過程のうち、(3)については従来から高級言語コンパイラによって自動化されてきたが、(1)及び(2)の仕様書の作成・修正及びプログラム文の作成については、機械的な作業が多いにもかかわらず、すべての作業が人手で行なわれており、作業効率向上の阻害要因となっている。

加えて、仕様書及びプログラム文の記述方法が必ずしも標準化されていないため、(1)品質が作業者の能力に依存する度合いが高いこと、(2)仕様書不備のまま、プログラム文だけが作成されやすいこと、(3)仕様書とプログラム文とに重複記述があり、しかも記述内容の不一致が生じやすいこと、などのソフトウェアの品質の面での問題が生じている。

プログラムの開発段階では、仕様書及びプログラム文は、頻繁に修正されるのが常である。この際に、仕様書とプログラム文とが2本立てになっていると、プログラム文だけ修正し、仕様書の中の関連事項が修正されない事態が生じやすい。仕様書は、プログラム完成後の保守のためには重要なドキュメントである。仕様書とプログラム文との間に内容の不一致があるのでは、保守時に仕様書を利用できず保守作業の効率を低下させてしまう。これを防止するために、プログラムが完成した段階で、改めてプログラム文から仕様書を人手で作成するという作業も生じている。

プログラム自動生成システムSDL/PAD(Software Design Language/Problem Analysis Diagram)^{1)~3)}は、これらの問題の解決を目的としたシステムであり、次の特徴をもっている。
 (1) 新しい設計技法を取り入れて標準化した仕様の記述法^{4)~6)}の提供
 (2) 上記(1)の記述法に従って作成された仕様を入力とし、プログラム文を自動編集するソフトウェアツールの提供
 (3) 上記(1)、(2)とは逆にプログラム文を入力とし、上記(1)

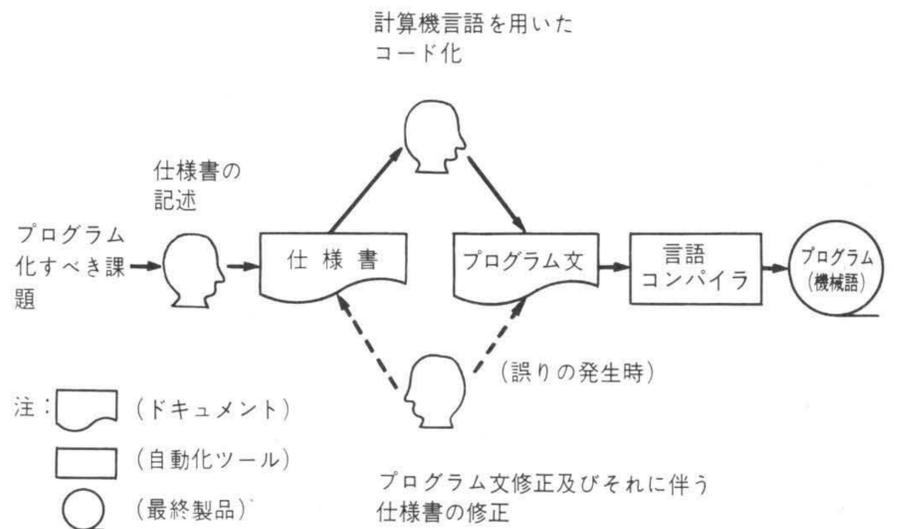


図1 現行のプログラム開発手順 現行のプログラム開発では、仕様書作成・修正及びプログラム文の作成のすべての作業が人手で行なわれており、作業効率が悪い。

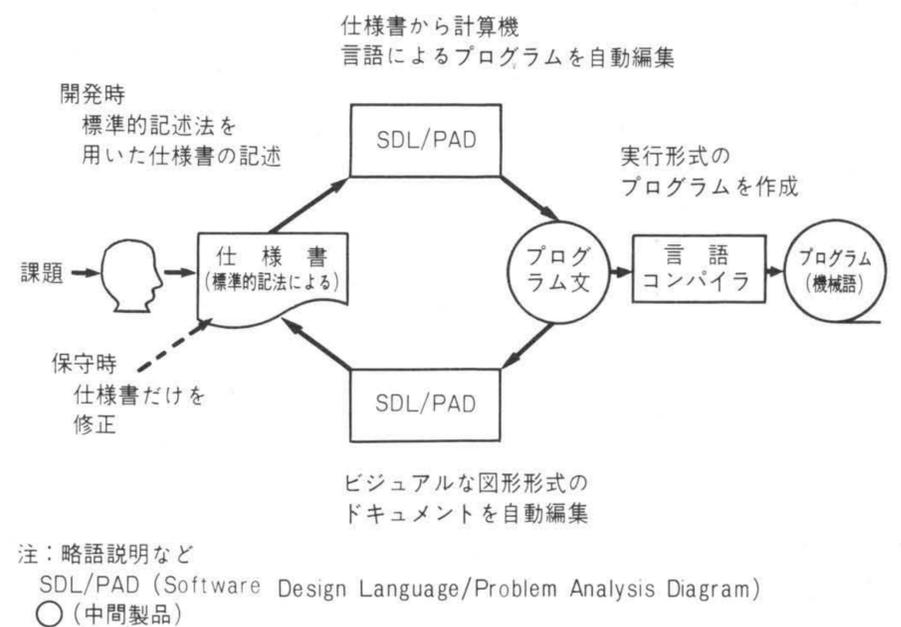


図2 SDL/PADシステムを用いたプログラム開発手順 プログラム文及び仕様書の作成が機械化され、作業効率が向上するとともに、プログラム文と仕様書との一元化、各々の品質の均一化を図ることができる。図1と図2を比較すれば、SDL/PADシステムのねらいが明らかになる。

* 日立製作所システム開発研究所 ** 日立製作所中央研究所

の記述法に従った仕様書を自動編集するソフトウェアツールの提供

(1), (2)により, 高い品質のプログラムを効率的に作成できることが期待できる。しかも, プログラムの品質の均一化も期待できる。(3)により, 仕様書のない既存プログラムの保守時にも, 保守用の仕様書を自動作成でき, 保守作業の効率向上が期待できる。

図2に, SDL/PADシステムを用いた, 新しいプログラム開発方式を示す。

2 新しいプログラム設計法を取り入れた仕様記述法

最初に, プログラムSDL/PADの基礎となるプログラム設計法と, 仕様の記述法について説明する。

2.1 SDL/PADシステムの基礎となるプログラム設計法

プログラムを整理された, 分かりやすい構造にすることは, 誤りの混入を防止し, たとえ誤りがあっても早期に検出を容易にすることから, 開発作業の効率を高めることになる。また, プログラムの完成後の機能改良及び拡張を容易にし, 保守作業の効率向上を促す。これによってプログラムの寿命を長くする効果がある。近年, プログラムを良い構造にする設計技法の研究⁴⁾が進み, 次の技法の有効性が立証されてきた。

(1) プログラム全体を, 小さなできれば単機能の部品(モジュ

ール)に分け, その部品を階層的に組み合わせて全体の機能を実現する構造とする(階層構造化)⁷⁾。

(2) 作りたいプログラムを最も一般的な形でまず表現し, 次の各構成部分を詳細な動作の記述に徐々に展開してゆき, これをプログラムが完成するまで続ける(段階的詳細化)⁸⁾。

(3) プログラムの実行順序を定める文を, (a)接続文, (b)条件文, (c)繰返し文の三つの基本文だけに限定し, プログラムを複雑にするGOTO文の使用を制限する(GOTOレス化)⁹⁾。

SDL/PADシステムは, 以上の技法を踏まえた設計を支援する機能をもっている。

2.2 SDL/PADで採用した仕様記述法

図3に, SDL/PADシステムの仕様記述法を示す。SDL/PADシステムではソフトウェアの仕様を, (1)モジュール階層, (2)モジュール入出力, (3)モジュール処理概要, (4)モジュール処理詳細及び内部データ関連, の4種で記述する方式をとっている。

モジュール階層仕様では, プログラムの階層構造化設計の結果できあがる部品(モジュール)の階層関係を記述する。モジュール入出力仕様では, モジュールごとに, 入力データ, 出力データ及びパラメータを記述する。モジュール処理概要では, モジュールが行なう処理手順の概要を記述する。ここでは, 処理の手順を人間に分かりやすい, 接続文, 条件文,

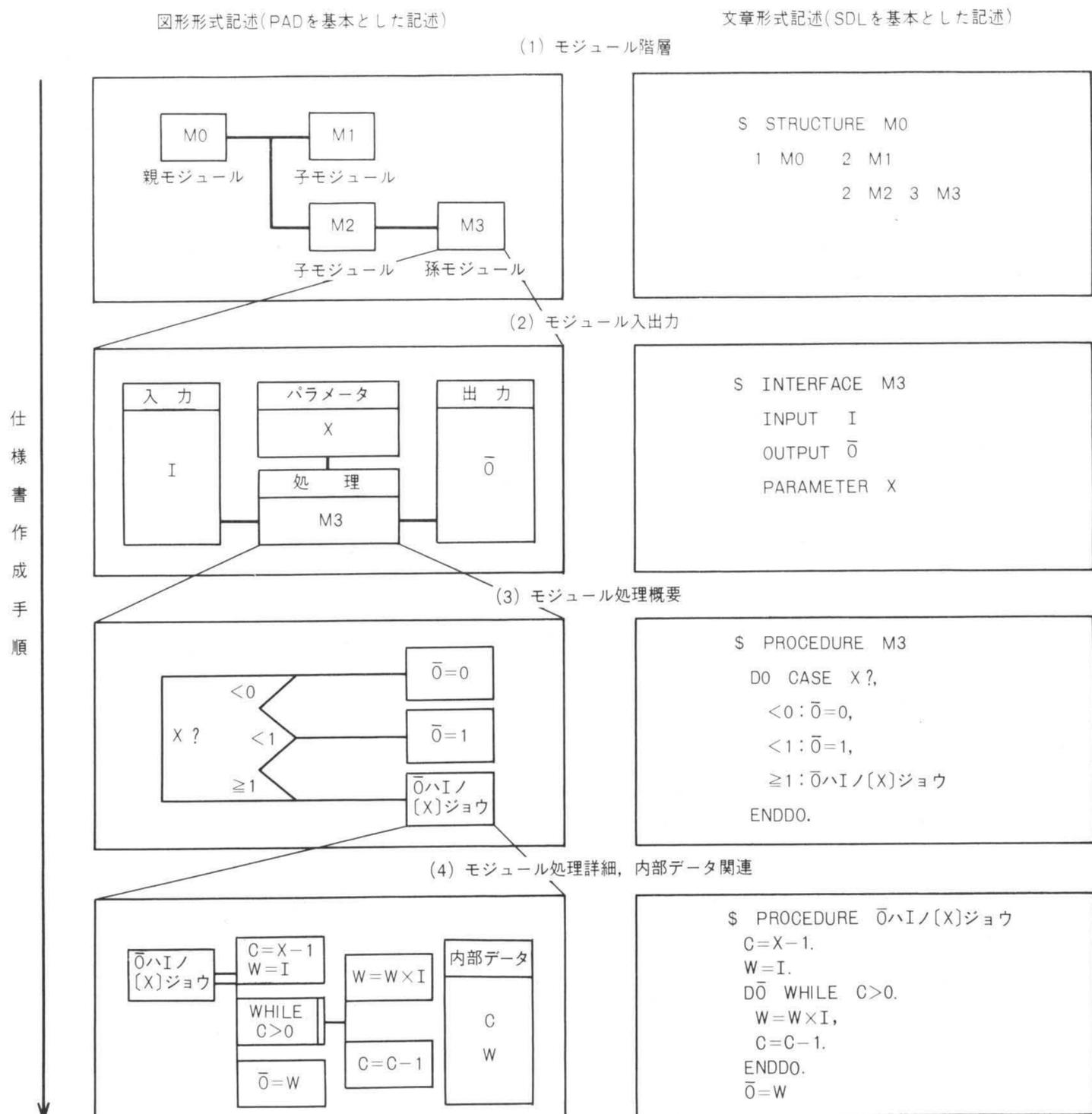


図3 SDL/PADシステムでの仕様書作成手順
4種の仕様書を作成することにより, 良い構造のプログラムを作成することができる。

表1 処理アルゴリズム仕様の記述法 文章形式と図形形式とは、各々一長一短がある。SDL/PADシステムの目的の一つは、計算機を用いて両者を融合し、長所だけを生かした仕様書作成支援を行なうことにある。

比較項目	文章形式	図形形式
	PDL	PAD
書き方	<pre> DO すべての注文 IF 受注量<在庫量 THEN 出荷量=受注量 ELSE 発注量=受注量- 在庫量 ENDIF ENDDO </pre>	
理解しやすさ	△ (条件が複雑になると字下げがあっても読みにくくなる。)	○ (記法を理解すれば、内容を直観的に読み取れる。)
書きやすさ	○ (書き下せる。)	△ (図の配置を考慮)
記述の挿入	○ (文の挿入だけ)	△ (図柄の修正要)
問題の整理	△	○

繰返し文の3種の型に当てはめて、自然語に近い文章で記述する。モジュール処理詳細及び内部データ仕様では、処理手順の詳細と、そこで利用するデータの詳細とを、計算機が意味を解釈できる形式(計算機言語レベル)で記述する。

仕様の記述法として、従来、例えばプログラム設計用言語PDL⁵⁾(Program Design Language)のような文章記述形式と、PAD図⁶⁾を例とする図形記述形式の2種がある。これらを比較すると、表1に示すように一長一短がある。SDL/PADでは場合に応じて優れた記述形式を選択できるように、SDL(PDLに比べて、モジュール構造記述など拡張)による文章とPADほかを用いた図形の両形式による仕様記述手段(図3)が兼備されている。これにより、設計者ないしプログラマは、記述しやすい文章形式で仕様を記述し、必要があれば理解し

やすい図形形式の仕様を随時に、計算機を利用して作り出すことができる。

以上の記述法の採用により、SDL/PADシステムでは、システムからの誘導に従って作業を進めることにより設計者及びプログラマは、難しい設計技法を意識せずに、品質が高くかつ保守の容易なプログラムの開発を行なうことができる。

3 SDL/PADシステムの機能とシステム構成

SDL/PADシステムは、次の四つの機能をもっている。

- (1) プログラム仕様の入力及び修正(バッチ及び対話方式)
 - (2) 計算機言語によるプログラム文の自動生成
 - (3) 図形形式ドキュメント(PAD図, HIPO図)表示及び印刷
 - (4) 計算機言語によるプログラム文を解析し、仕様書を作成
- 以上(1)~(4)の機能のうち、(1)、(3)は、計算機言語文法に依存しない機能であり、(2)、(4)は、計算機言語文法に依存する機能である。

図4に、SDL/PADシステムのシステム構成を示す。

以下に、上記四つの機能の概要を説明する。

(1) 仕様入力及び修正

仕様は、前章で説明したように、文章形式(SDL文)又は図形形式(PAD図)で記述される。文章形式の場合には、TSS(Time Sharing System)端末ないしカードによるバッチ方式で計算機に仕様を入れることができ、図形形式の場合には、グラフィックのTSS端末(H-8844)から仕様を計算機に入れることができる。

図5は、図形形式の場合の仕様入力方式を示す。入力用キーボードの特殊文字3文字を、□, □, □に当て、図形の形状をキーボードから入れる。図形の配置、結合関係は、画面にガイダンスが出るので、それに答える形式で入力する。

計算機に投入された仕様は、計算機内部の仕様ライブラリに記憶される。この仕様ライブラリを用いて、随時、未定義

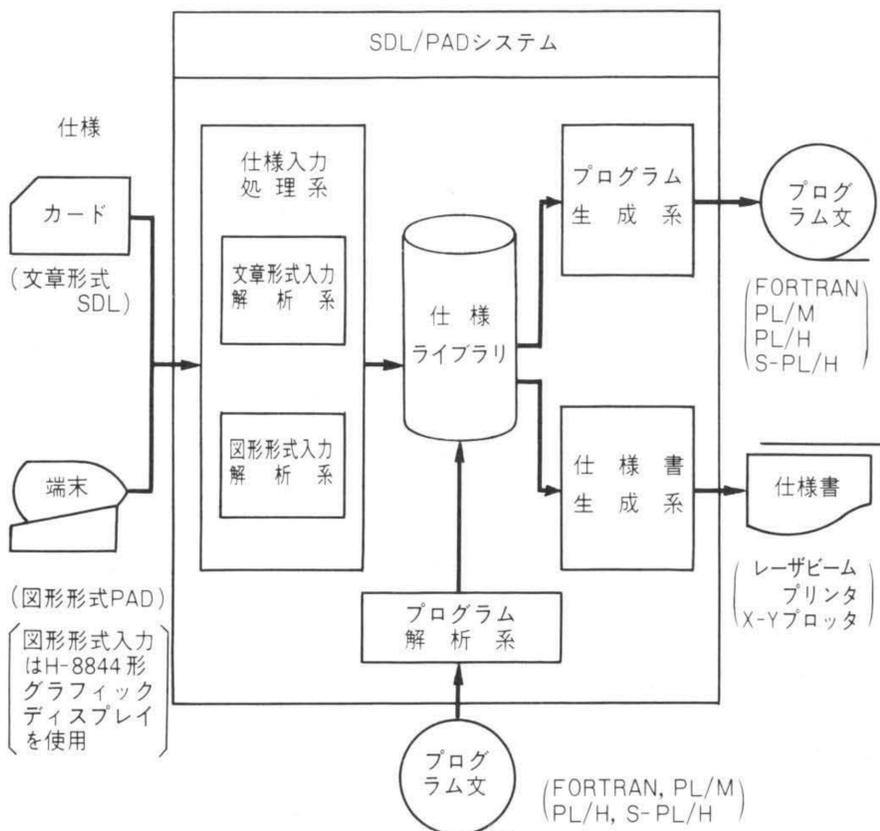


図4 SDL/PADシステムのシステム構成 SDL/PADシステム自体はFORTRANで記述されている。プログラム生成系とプログラム解析系とは計算機言語に依存する部分で全体の約半に当たる。

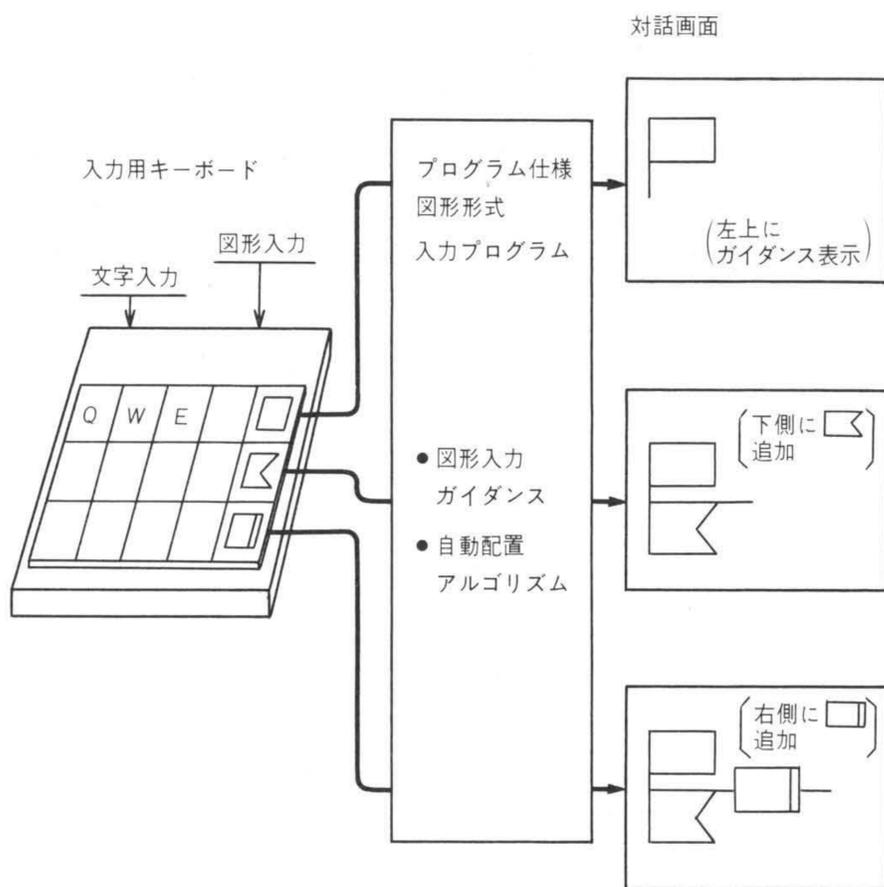


図5 図形形式のプログラム仕様の対話形入力方式 画面に表示されるガイダンスに従って、キーボードから図形形式のプログラム仕様を、計算機に入れることができる。

データの使用，詳細処理仕様の定義漏れなどの，仕様の不備を，計算機でチェックさせることができる。

(2) 計算機言語によるプログラム文の作成

仕様ライブラリに蓄積された仕様データから，計算機言語によるプログラム文を編集する。図6に仕様からプログラム文を編集する方式の概略を示す。編集は，(a)図形の配置及び形状(文章形式の場合はIF, THEN, ELSEなどのキーワード)を，計算機言語のキーワードに変換する，(b)図形の結線(文章形式の場合は文章の記述順序)情報をもとに，計算機言語文法に従った文の配列順序に変換する，の二つの過程を経て行なわれる。

編集されたプログラム文は，計算機言語のコンパイラに渡され，機械語のプログラムに変換される。コンパイラで誤りが発見された場合には，仕様と対応して，誤りを表示する機能をもSDL/PADシステムには備わっている(図7)。

以上の機能は，計算機言語に依存している。現段階では，SDL/PADシステムが扱える計算機言語は，FORTRAN及びマイクロコンピュータ用高級言語PL/H, S-PL/H, PL/M80及びPL/M86などである。

(3) 図形化ドキュメントの表示及び印刷

計算機に投入された仕様データから図形形式の仕様書を作成し，レーザビームプリンタないしX-Yプロッタに出力する。

また，グラフィックディスプレイで随時仕様書の必要なページを見ることが出来る。

(4) プログラム文から仕様書への変換

以上に述べた(1)~(3)の機能は，仕様データから，プログラム及び図形形式の仕様書を作成する機能について述べた。SDL/PADでは，更に，SDL/PADシステムを用いずに作成された既存のプログラム文を入力として，図形形式の仕様書を作る機能をもっている。これは，プログラム文を解読して制御フロー，手続き関数及びブロック構造を抽出して，PAD図，モジュール関連図を作成する。図8に，PL/Hのプログラム文から，PAD図を作成した例を示す。このソースプログラムからPAD図をはじめとする仕様書ドキュメントを作成する機能は，ADCAS(Auto Documentation Aid System)¹⁰⁾として既に製品化されている。ADCASで扱う計算機言語は，COBOL及びPL/Iである。

以上に述べた四つの機能により，仕様からソースプログラムを作ることができるとともに，逆にソースプログラムから仕様書を作ることができ，仕様とプログラムとの完全一体化を図ることができる。

4 SDL/PADシステムの代表的な利用法

SDL/PADシステムの利用法の代表例を次に説明する。

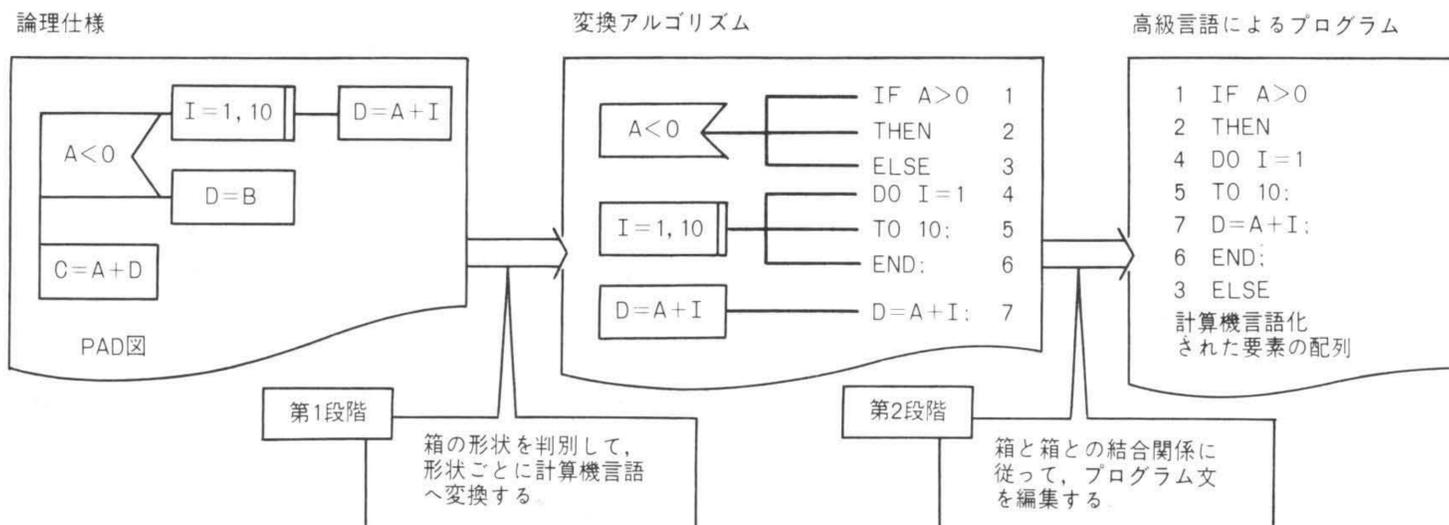


図6 プログラム自動編集の原理 図形形式の処理詳細仕様からプログラム文を編集する方式を示している。記述形式の処理詳細仕様からプログラム文を編集する場合は，図形の代わりに仕様記述言語(SDL)のキーワードを，計算機言語のキーワードに変換することになる。

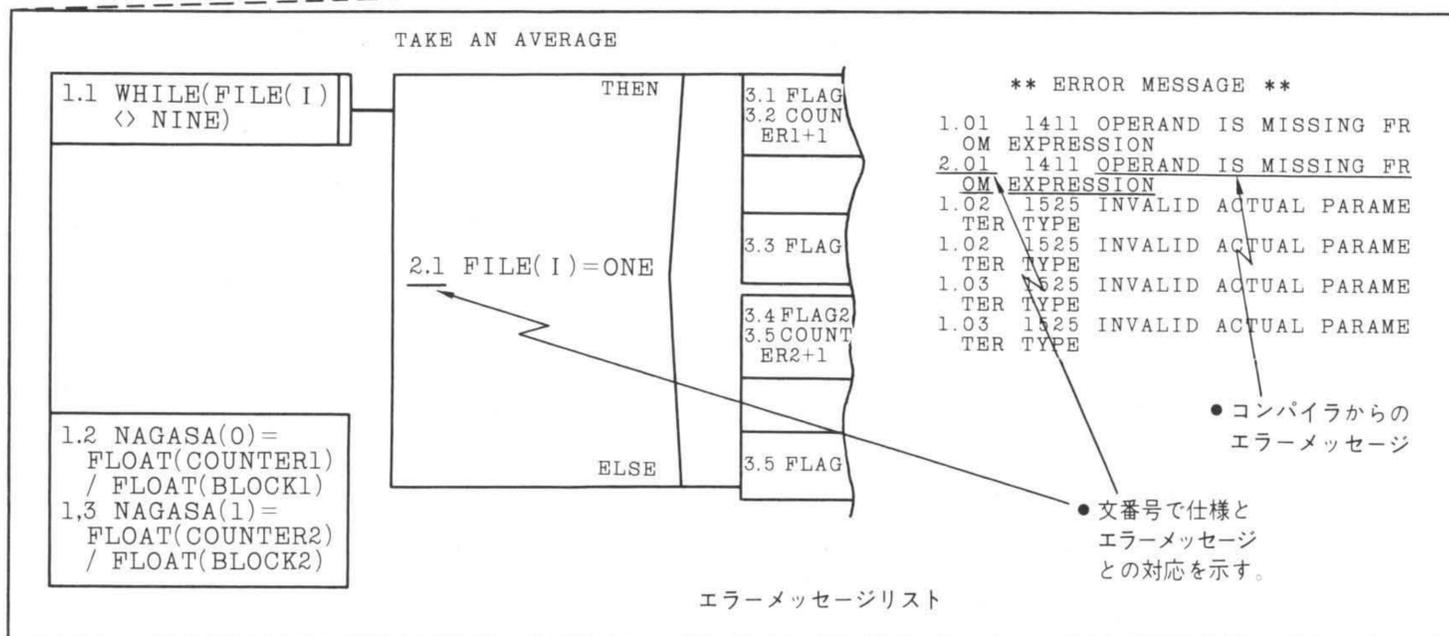
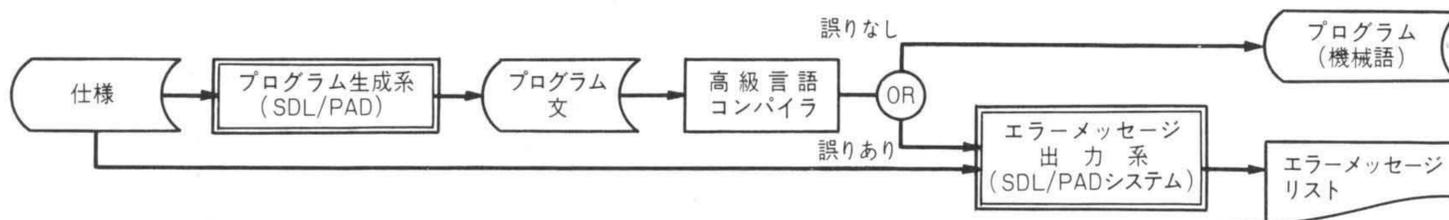


図7 SDL/PADシステムと計算機言語コンパイラとの関連 SDL/PADシステムで作成されたプログラム文は，コンパイラに受け渡されて機械語のプログラムに変換される。コンパイラで誤りが検出された場合には，SDL/PADシステムは仕様とエラーメッセージとを対応させて表示出力する。

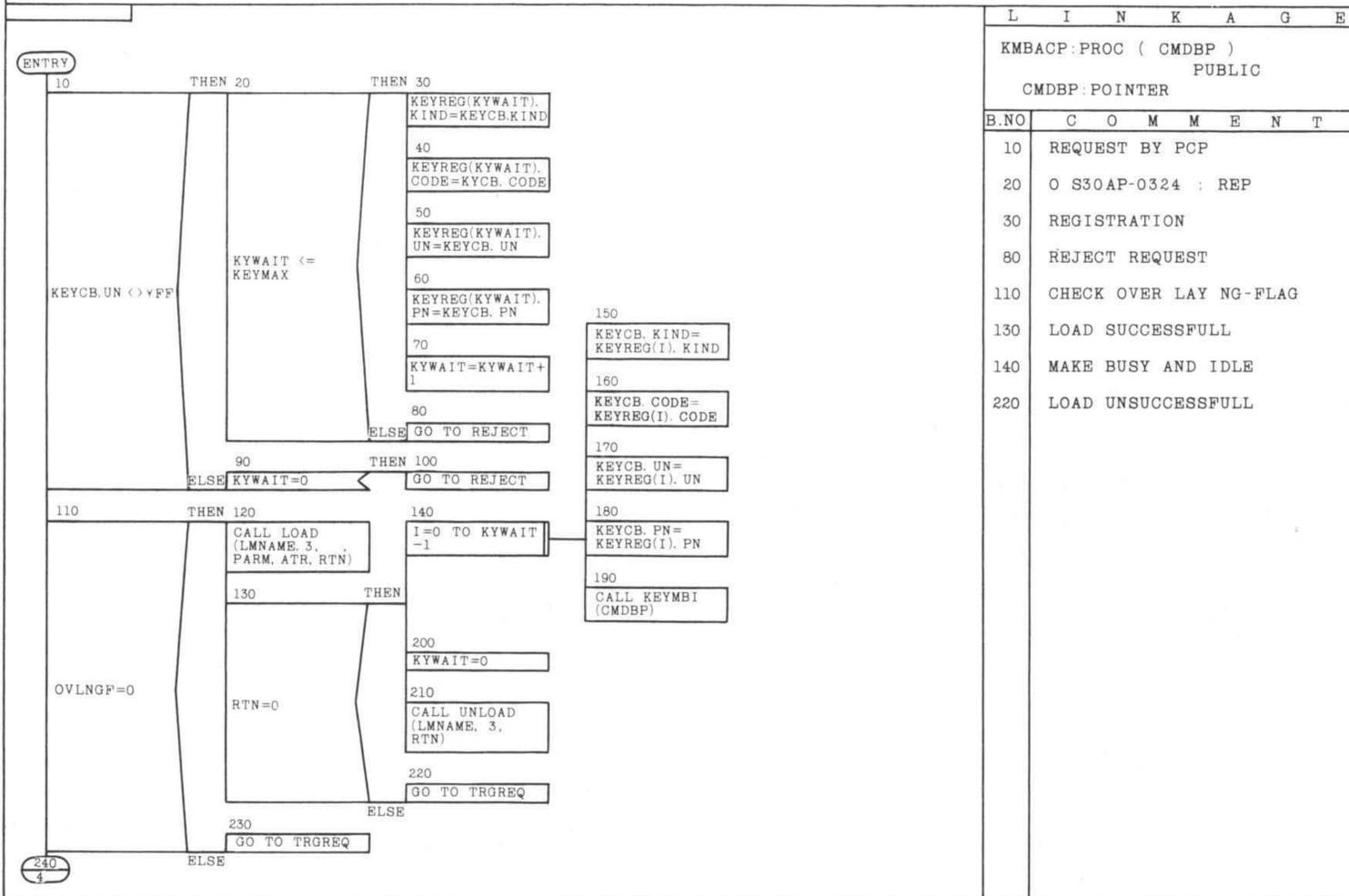
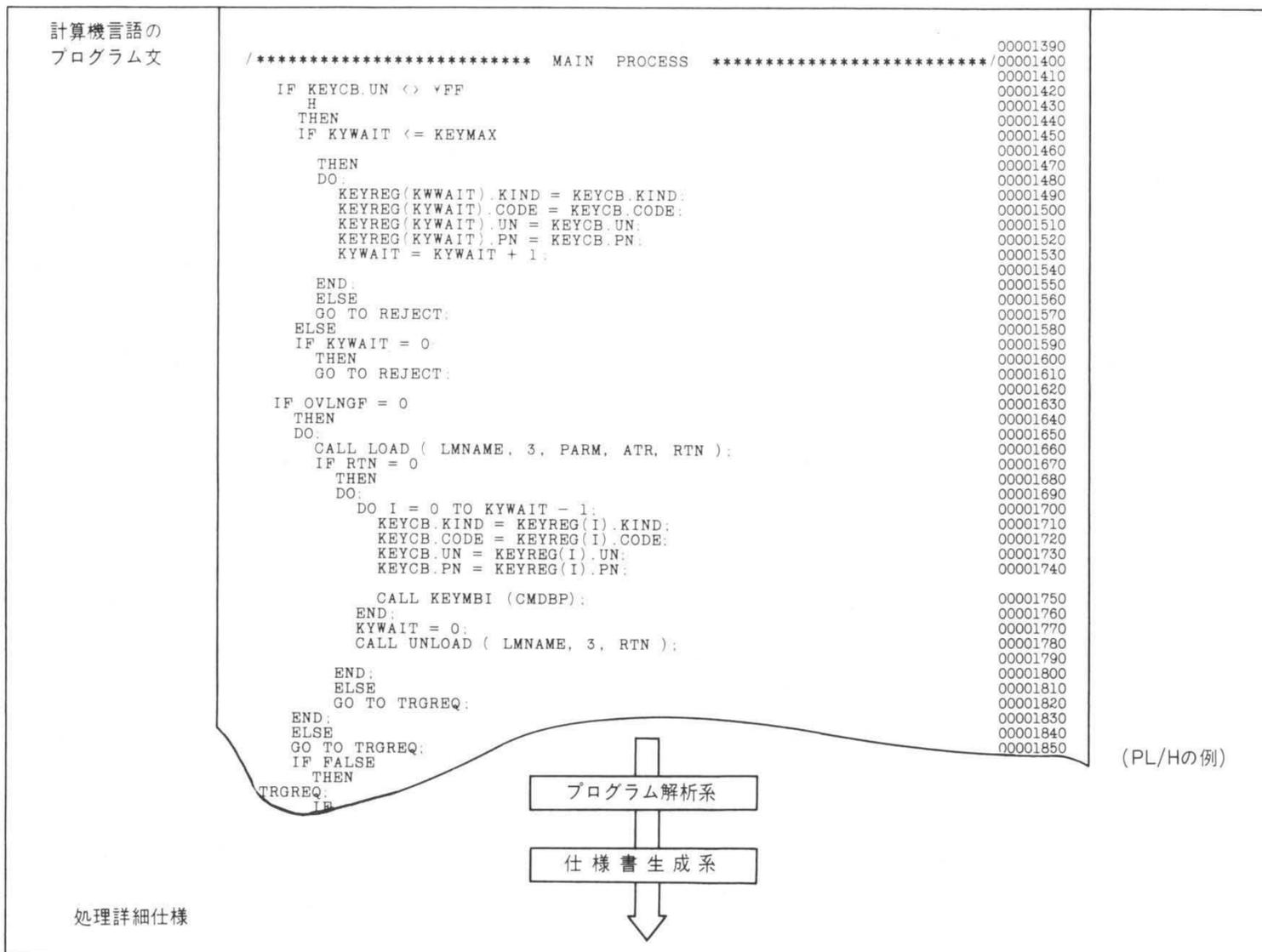


図8 計算機言語によるプログラム文から仕様書への変換 プログラム文(上記例はPL/Hのプログラム文)を入力として、図形形式の処理詳細仕様をレーザービームプリンタへ出力した例を示す。図形表現により、プログラム構造が分かりやすくなっている。

4.1 プログラムの新規開発

SDL/PADシステムを用いたプログラムの新規開発は、次のステップで行なわれる(図2参照)。(1)設計及びその結果としての仕様をSDL/PADシステムへ入力、(2)図形化された仕様書をSDL/PADシステムから出力させ、内容をレビュー、(3)仕様に誤りがあればSDL/PADシステムを利用し仕様を修正、(4)SDL/PADシステムによりプログラム文(ソーステキスト)を生成、(5)完成したプログラム文をコンパイラにより機械語

のプログラムに変換、(6)機械語への変換時にコンパイラが誤りを発見した場合は、SDL/PADシステムが誤りを仕様とともに表示するので、再び(3)、(4)、(5)のステップを繰り返す。

開発の最初のステップで、設計及びその結果としての仕様をSDL/PADへ入力することになるが、この方式としては、次の二通りが考えられる。

- (1) 設計者ないしプログラマが、最初に設計メモを手で作成し、メモを見ながら設計データを入力する方式

(2) 設計者ないしプログラマが、SDL/PADシステムからの出力画面を見ながら、対話で設計を行なう方式

以上、2方式のうち現状では、(1)が一般的と考えられる。しかし、(2)の方式は、SDL/PADシステムの誘導に従って作業を進めれば、設計技法を意識せずに設計を行なえる利点をもっている。将来は(2)の方式も有望である。ただし、(2)の方式が普及するには、対話操作が容易になること、作業員一人に1台の端末が利用できる設備面での充実が必要である。

4.2 既存プログラムの保守改良時

SDL/PADシステムを用いないで作成された既存プログラムの保守改良にSDL/PADシステムを利用する作業は、次のステップで行なわれる(図2参照)。(1)既存プログラム文を、SDL/PADシステムへ入力する、(2)SDL/PADから図形化仕様書を編集出力させる、(3)この仕様書をもとに、作業員が修正箇所を検出し、修正する。以下は、4.1で述べた新規開発と同様の方法によりプログラムを作成する。以上により、保守改良の場合にも、仕様とプログラムとの一体化を図ることができる。

5 本方式の効果

SDL/PADシステムを、本システムの開発に適用し、従来の開発結果と比較して算定した効果を図9に示す。量的な効果としては、(1)仕様記述の標準化、仕様書の自動編集により、プログラム設計及び誤り修正作業に対し20%の効率向上となる、(2)仕様の不備チェック機構、仕様のビジュアル化によるレビューの徹底により設計エラー検出時間の短縮65%、(3)プログラム仕様から、計算機言語プログラム文を自動作成する機能により、コーディング作業が不要となるため2日/kステップ削減、以上を総合すると、プログラム設計からデバッグ完了までの工数削減は従来の40%減、工期短縮は30%減が期待できる。

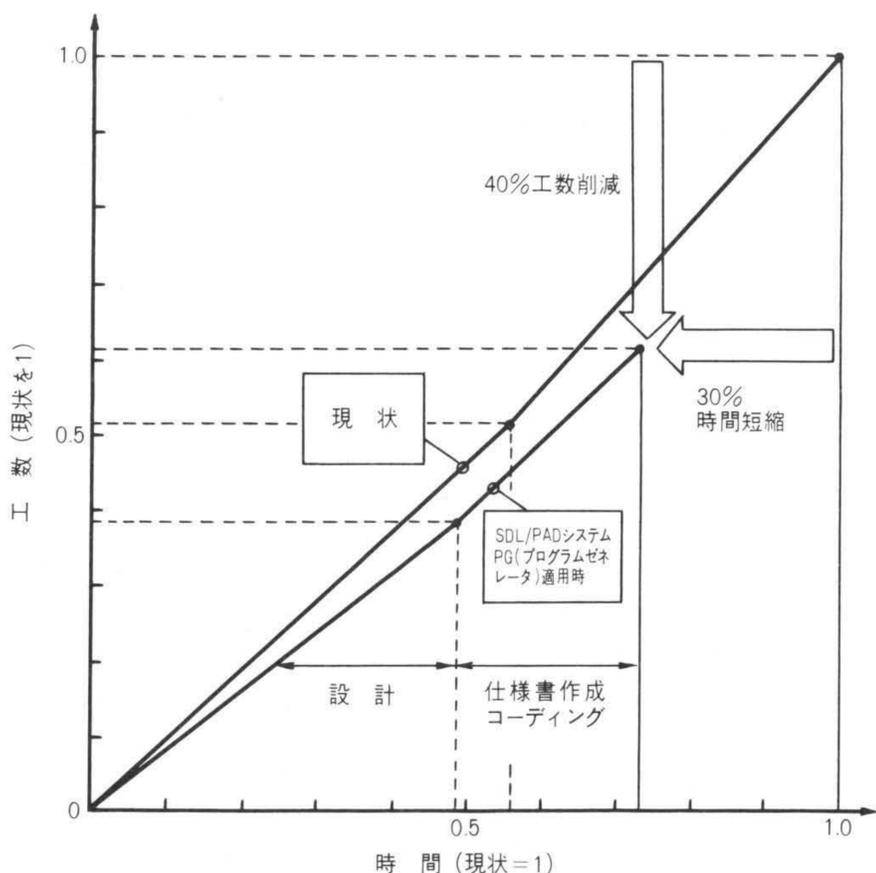


図9 SDL/PADシステムの適用期待効果 SDL/PADシステムを適用することにより、従来の人手による開発に比べて、工数については40%、工期については30%の削減が期待される。

質的な効果としては、(1)構造化設計などの新しい設計技法を取り入れた仕様記述の標準化により、経験の浅い開発者であっても、一定水準以上の品質のプログラムを開発できることが期待できる、(2)仕様書の図形化により、文書のビジュアル化が図れ、関与者間の情報伝達が正確になり、プログラムの品質向上及び開発の効率化が期待できる。

6 結 言

高品質の計算機プログラムを、効率的に開発することを支援する計算機システムとして、プログラム自動生成システムSDL/PADを開発した。このシステムは、標準化されたプログラム仕様を入力とし、プログラムを自動作成する機能と、逆に、プログラムを入力とし、図形化されたビジュアルな仕様書を自動作成する機能とをもったものである。本論文では、このシステムの機能と、代表的な利用方法とを中心に報告した。

本システムは、計算機システムへの要求を仕様に変換する技術、対話型設計技術、仕様からテストケース及びデータを作成する技術などの研究成果を取り入れ、更に総合的な計算機ソフトウェア生産支援システムへと拡張することができる。日立製作所では、この方面の研究開発も既に着手している^{11), 12)}。

参考文献

- 1) 斎藤, 外: 図形化論理仕様出力システムの開発, 情報処理学会第22回全国大会(昭56-3)
 - 2) 前沢, 外: 処理論理図自動作成システムの開発, 情報処理学会第23回全国大会(昭56-9)
 - 3) H. Maezawa, et al.: Interactive System for Structured Program Production, Proc. of 7th International Conference on Software Engineering, 1984.3 Orlando, Florida.
 - 4) 鳥居, 外: プログラミング方法論の展望, 情報処理, Vol. 20, No. 1, pp. 22~43(昭54)
 - 5) S. H. Caine, et al.: PDL-A Tool For Software Design, Proceedings of National Computer Conference (1975)
 - 6) 二村, 外: PAD(Problem Analysis Diagram)によるプログラムの設計および作成, 情報処理学会論文誌, Vol. 21, No. 4, (昭55-7)
 - 7) 国友, 外訳, G. J. Myers: ソフトウェアの複合/構造化設計, 近代科学社(昭54)
 - 8) 野下, 外訳, N. Wirth: 系統的プログラミング/入門, 近代科学社, (昭50)
 - 9) 野下, 外訳, E. W. Dijkstra, et al.: 構造化プログラミング, サイエンス社(昭50)
 - 10) HITAC自動ドキュメンテーション支援システムADCAS, マニュアル番号8090-3-322, (株)日立製作所, 昭和58年4月(第1版)
 - 11) M. Kobayashi, et al.: ICAS: An Integrated Computer Aided Software Engineering System, SPRING '83 COMPCON
 - 12) 小林, 外: ソフトウェア一貫生産システム"ICAS"の基盤技術の確立, 日立評論, 66, 3, 171~176(昭59-3)
- なお、この日立評論、66, 3(昭59-3)は、ソフトウェア生産技術特集であり、上記以外に関連論文10篇が掲載されている。