

マイクロコンピュータ用ソフトドキュメント自動生成システム“Auto-DS”

Automated Documentation Support System “Auto-DS”

既存のプログラムを修正したり、再利用する場合には、プログラムの内容を正確に記述したドキュメントが不可欠である。とりわけ、開発量の急増が著しいマイクロコンピュータの分野では、プログラムの蓄積が増えるにつれてドキュメント整備が大きな問題となっている。そのため日立製作所では、事業所ごとに個々に制定されたマイクロコンピュータ用ドキュメントの全社標準化、規格化を図るとともに、それに基づいて、標準的なドキュメントをプログラムから自動生成するシステム“Auto-DS”を開発し実用化した。プログラム再利用時に本システムを適用した結果、作成するドキュメントの8割を自動的に生成できることが確認されたほか、設計レビューやデバッグなどにも適用され大きな効果を挙げている。

橋本祐宏* *Yūkō Hashimoto*
鵜飼純一** *Jun'ichi Ugai*
増井光幸*** *Mitsuyuki Masui*
上藤博司**** *Hiroshi Kamifuji*

1 緒言

マイクロコンピュータを組み込んだシステム製品では、その製品の優位性を確保するために、基本機能に加え徐々に機能拡張を続け、ソフトウェアの規模は拡大する一方である。しかも、市場競争の激しいOA (Office Automation) 機器などでは、新機種の発表サイクルが年々短縮され、短期開発能力の確保と増大する蓄積ソフトの保守が大きな問題となっている¹⁾。

このような状況の中で、膨大なドキュメントを開発時に完全な形で整備していくことは難しいうえ、過去開発されたソフトウェアの中には、ドキュメントが不備のプログラムも少なくない。この結果、プログラムの再利用や保守をますます困難にしている。この解決策として、設計工程でドキュメントを対話的に作成するツールが開発されてはいるが、プログ

ラミング工程で発生した変更を、設計ドキュメントに確実に反映させるための現実的な解決策にはなっていない。そこで、ソフトウェア開発の最終産物であるプログラムそのものからドキュメントを自動的に生成し、プログラムとドキュメントの完全一致を図るツールの必要性が増大してきた(図1)。

これまでは、プログラムからドキュメントを自動生成するツールで出力されるドキュメントは、プログラムの詳細な論理図や構成図など出力情報種別が少ないものに限られ、プログラム理解のためのドキュメントとしては不十分であり、また、使い勝手の面でも不備な点が多かった。

日立評論、第66巻、3号(昭和59年3月25日発行)の東京大学大型計算機センター石田晴久教授 Ph.D.の巻頭言に触れられているように、「ツールの使いにくさ」問題を解決することがツール普及の鉄則との認識をもち「利用負担は最小限」、「必要ドキュメントを即時提供」を実現するシステムAuto-DS (Automated Documentation Support System)を開発した。本論文では、Auto-DSの開発方針と特長技術を中心に述べる。

2 開発の方針

Auto-DSの開発上の基本方針は、使い勝手の良いツールを作ることであり、以下に示す具体的な方針を設定し、実現を図った。

(1) ツール利用者に特別の負担を掛けない。

ドキュメントを生成するための基本的な情報は、すべてソースプログラムから入手することとし、ソースプログラム中の命令から得られない意味情報及び構造明示情報の表現にコメントを利用する。一般に、プログラム開発者はプログラムを理解しやすくするために適宜コメントを挿入している点に着目し、コメント記述の標準化を行なった(図2)。コメントの挿入位置関係とキーワードから情報を抽出可能とする最小限の記述ルールを設定し、キーワードについては、既に日立製作所内で実用されているモジュール仕様記述用言語MDL (Module Design Language)²⁾を基本とし、若干の追加を行なった。

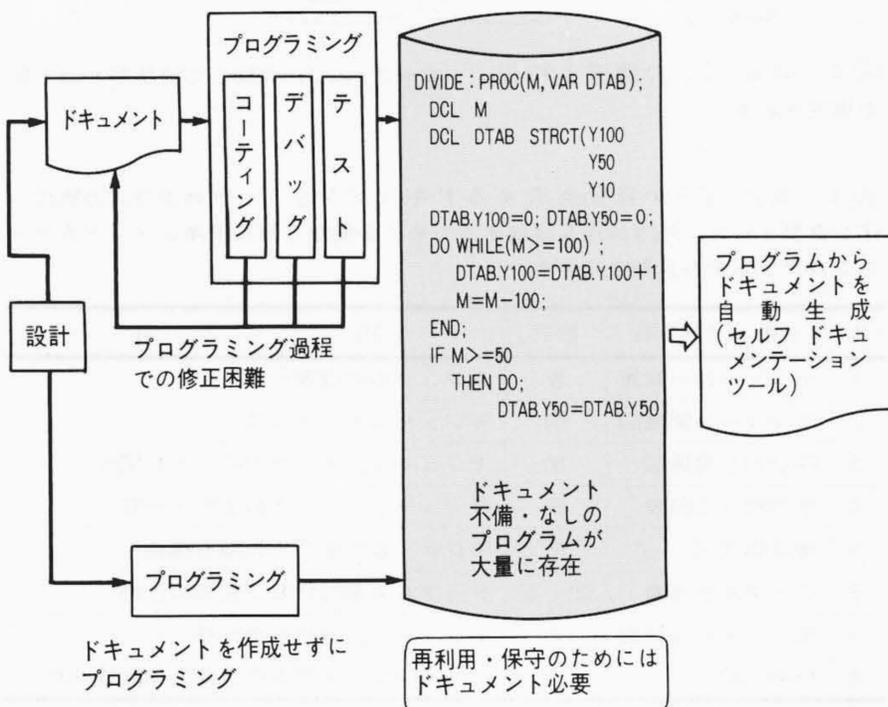


図1 ソフトウェア再利用・保守における問題点と解決策 ソフトウェアの再利用・保守のためには、プログラムの内容を正しく反映したドキュメントが必要であるが、現実にはドキュメント不備のプログラムが少なくない。

* 日立製作所システム開発研究所 ** 日立製作所生産技術部 *** 日立製作所戸塚工場 **** 日立製作所佐和工場

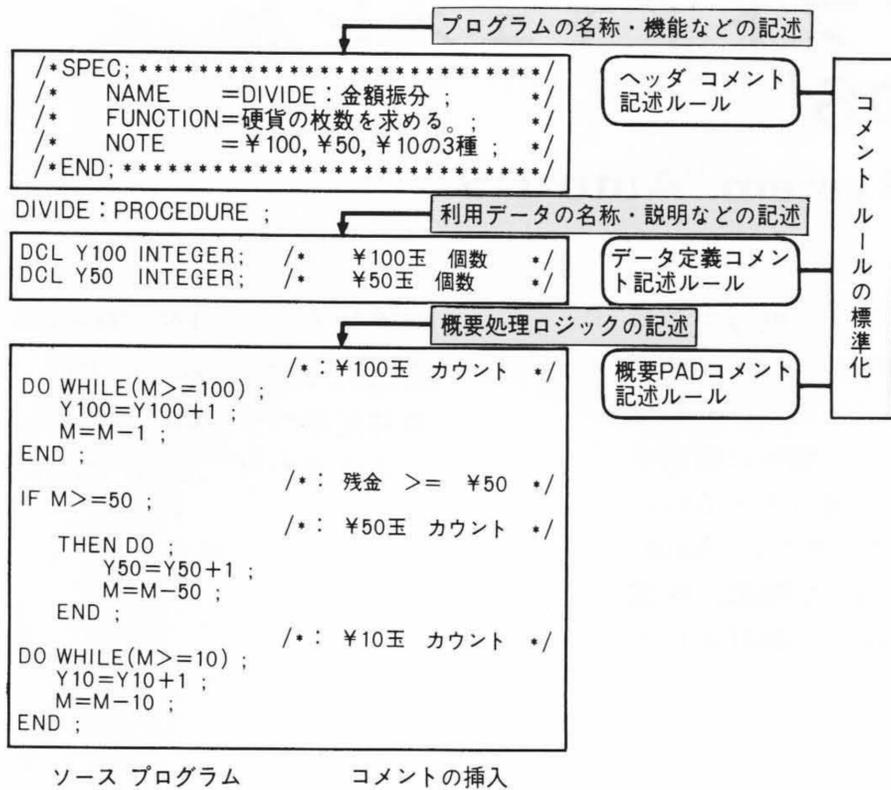


図2 コメントの必要性と標準化 プログラムを理解しやすくするために、通常挿入するコメントの記述ルールを標準化し、このコメントをドキュメント生成に利用する。

(2) 体系化されたドキュメントを作成する。

プログラムを理解するためには開発過程と同様に、プログラムの概要把握から順次詳細に入ってゆく過程が必要である。すなわち、プログラムの全体の構造及び概要を理解し、プログラムの構成要素であるモジュールの相互の関係、モジュールの入出力インタフェース及び処理論理と段階的に理解していくことになる(図3)。これに対応してドキュメントを体系的に整備しなければならない。この観点から出力対象ドキュメントを選定し、体系的なドキュメントを出力可能とした。

(3) プログラミング言語に対する適用拡大を可能とする。

マイクロコンピュータの分野では高級言語の利用も進んでいるが、性能確保などのためアセンブリ言語の使用も続くものと思われる。また、マイクロプロセッサの機種も多岐にわたり、それに対応したアセンブリ言語も多様である。これらの言語に対して順次適用拡大を図れる方式をとった。

3 システムの構成と機能

Auto-DSは、図4に示すように解析部と出力部で構成する^{3),4)}。解析部では、2章で述べたようにコメントを挿入したソースプログラムを入力として、プログラムを構成するモジュール単位に解析を行ない、解析情報ファイルに格納する。出力部では解析情報ファイル中のモジュール単位の情報からプログラムを理解するのに必要な、「プログラム構成」、「インタフェース仕様」、「処理論理仕様」の三つの側面からのドキュメント一式を生成出力する。

なお、ドキュメント性やプログラムそのものの読みやすさを向上させるためにも、できるかぎりコメントを挿入することが望ましいが、高級言語の場合には、コメントが挿入されていないものでも処理論理仕様を除く他のドキュメントはすべて出力することができる。

以下に、本システムの機能と特長について述べる。

3.1 保守ドキュメントの生成

Auto-DSは、表1に示す8種類のドキュメントを自動生成する。これらのドキュメントの特長を以下に述べる。

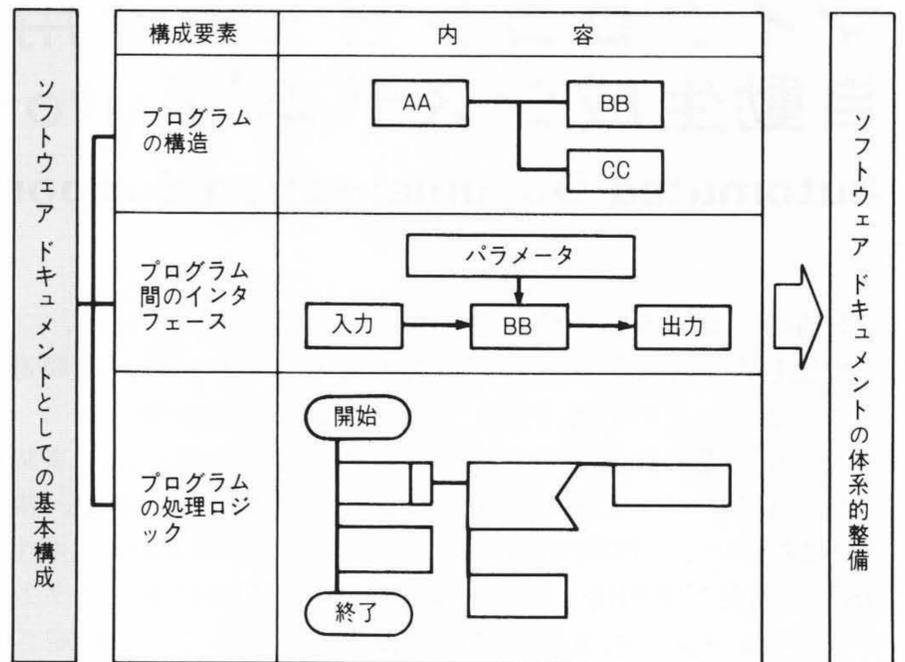


図3 プログラム理解のためのドキュメント プログラムを理解するためには、上記3要素をもったドキュメントが必要不可欠である。

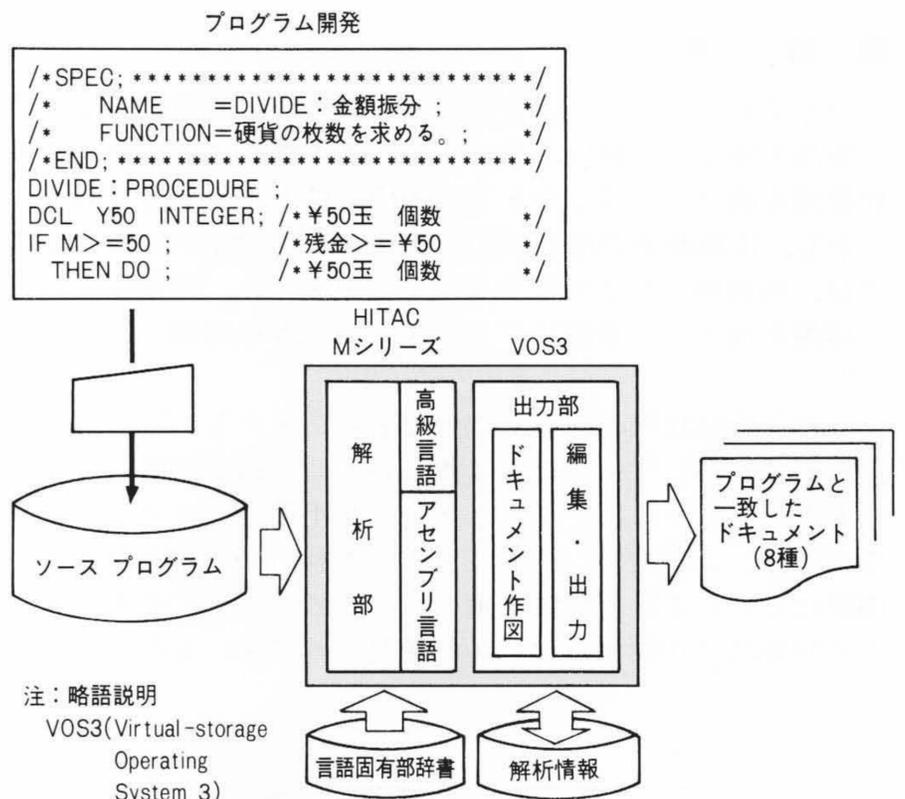


図4 Auto-DSの構成と特長 本システムは、大別して解析部と出力部で構成される。

表1 Auto-DSが自動生成するドキュメント プログラムの構成、インタフェース、処理論理を理解するために必要な8種のドキュメントをソースプログラムから自動生成する。

No.	ドキュメント名	形式	内容
1	モジュール一覧表	表	モジュールの概要一覧
2	モジュール関連図	図	モジュールの階層構造
3	呼び出し関係図	図	モジュール、手続きの呼び出し関係
4	手続相互参照表	表	モジュール、手続きの参照元一覧
5	機能構成図	図	モジュール内手続きの階層構造
6	テーブル仕様書	図・表	テーブルの割付け及び変数の仕様
7	モジュール仕様書	表	モジュール、手続きの仕様
8	概要PAD図	図	モジュール、手続きの概要処理論理仕様

注: 略語説明 PAD(Problem Analysis Diagram)

(1) ドキュメントは、図・表のイメージを重視し、LBP(Laser Beam Printer)によって出力する。なお、X-Yプロッタによっても出力できる(図5)。

- (2) 日本語によるドキュメントの出力も可能である。
- (3) 必要な種類のドキュメントだけを、分割して出力できるとともに、プログラムの単位(モジュール, 手続き)ごとに、1セットのドキュメントを配列・編集し、プログラム仕様書としての体裁を整えて出力できる。

3.2 概要処理論理仕様図の生成

プログラムの論理を木構造で表現し、構造を見やすくするプログラム論理図記述法として、日立製作所ではPAD⁵⁾(Problem Analysis Diagram)を開発し、実用化している。本システムでは、このPAD表記に従ったプログラムの概要処理論理仕様図(以下、概要PAD図という。)を生成する機能をもっている。PAD図生成を支援するツールとして、高級言語で記述されたプログラムを対象に、プログラム中のステートメントそのものを構造表現するツール(詳細PAD生成ツール)⁶⁾は既に開発済みである。しかし、開発当事者以外の者が、プログラムの処理論理の概要を理解するには詳細にすぎるきらいがある。

これに対し、本システムで出力する概要PAD図は、プログラムの理解を容易にするために、ステートメントの代わりにプログラム中に挿入された処理概要コメントを用いて表現する。すなわち、プログラムのステートメントを解析し、静的

構造を抽出し、この構造中に、プログラム中に挿入されたコメントを正しく配置することによって、概要PAD図を生成する⁷⁾。高級言語による出力例を図6に示す。

本システムでは高級言語だけでなく、アセンブリ言語についても、概要PAD図の生成ができるのが大きな特徴である⁸⁾。出力例を図7に示す。

なお、本機能を利用して、アセンブリソースプログラムの詳細PAD図を生成することも可能である。

3.3 対象言語の拡張

マイクロコンピュータの世界では、多様な言語が利用されるため、本システムでは入力の対象言語を、利用者が柔軟に拡張できるように配慮している。

- (1) 高級言語で記述された入力プログラムの解析に当たっては、構造化言語の形態をもった中間言語に基づいて解析する方式を採用しており、入力プログラムを言語種別に応じて、システム中でこの中間言語に変換している。

対象言語の拡張は、この中間言語への変換部を作成・追加することによって可能であり、現在、S-PL/H, PL/M, PL/M-86, PL/H, Cを用意している。

- (2) これに対してアセンブリ言語では、手続き定義の方式、コメント識別子、ラベル表記、分岐・リターン・データ定義

<table border="1"> <tr><td>MODULE</td><td>VML</td><td>PROC</td><td>MONEY</td></tr> <tr><td colspan="4">NAME = PRICE</td></tr> <tr><td>15</td><td>7</td><td>0</td><td></td></tr> <tr><td>(2)</td><td>(2)</td><td>(2)</td><td></td></tr> <tr><td>0</td><td>8</td><td>15</td><td></td></tr> <tr><td colspan="4">PRICE</td></tr> </table>	MODULE	VML	PROC	MONEY	NAME = PRICE				15	7	0		(2)	(2)	(2)		0	8	15		PRICE				<table border="1"> <tr><td>SYMBOL</td><td>PRICE</td><td>NAME</td><td>カカク</td><td>SIZE</td><td>2BYTE</td></tr> <tr><td>LV</td><td>NAME</td><td>ATTR</td><td>LN</td><td>BYTE</td><td>B</td><td>CONTENT</td></tr> <tr><td>1</td><td>PRICE</td><td>INTE</td><td></td><td>0</td><td>0</td><td>カカク</td></tr> </table>	SYMBOL	PRICE	NAME	カカク	SIZE	2BYTE	LV	NAME	ATTR	LN	BYTE	B	CONTENT	1	PRICE	INTE		0	0	カカク																													
MODULE	VML	PROC	MONEY																																																																							
NAME = PRICE																																																																										
15	7	0																																																																								
(2)	(2)	(2)																																																																								
0	8	15																																																																								
PRICE																																																																										
SYMBOL	PRICE	NAME	カカク	SIZE	2BYTE																																																																					
LV	NAME	ATTR	LN	BYTE	B	CONTENT																																																																				
1	PRICE	INTE		0	0	カカク																																																																				
<table border="1"> <tr><td>MODULE</td><td>VML</td><td>PROC</td><td>MONEY</td></tr> <tr><td colspan="4">NAM = COIN</td></tr> <tr><td>15</td><td>7</td><td>0</td><td></td></tr> <tr><td>(2)</td><td>(2)</td><td>(2)</td><td></td></tr> <tr><td>0</td><td>8</td><td>15</td><td></td></tr> <tr><td colspan="4">Y100</td></tr> <tr><td colspan="4">Y50</td></tr> <tr><td colspan="4">Y10</td></tr> </table>	MODULE	VML	PROC	MONEY	NAM = COIN				15	7	0		(2)	(2)	(2)		0	8	15		Y100				Y50				Y10				<table border="1"> <tr><td>SYMBOL</td><td>COIN</td><td>NAME</td><td>トウニユウ キンガク</td><td>SIZE</td><td>6BYTE</td></tr> <tr><td>LV</td><td>NAME</td><td>ATTR</td><td>LN</td><td>BYTE</td><td>B</td><td>CONTENT</td></tr> <tr><td>1</td><td>COIN</td><td></td><td></td><td>0</td><td>0</td><td>トウニユウ キンガク</td></tr> <tr><td>2</td><td>Y100</td><td>INTE</td><td></td><td>0</td><td>0</td><td>¥100</td></tr> <tr><td>2</td><td>Y50</td><td>INTE</td><td></td><td>2</td><td>0</td><td>¥50</td></tr> <tr><td>2</td><td>Y10</td><td>INTE</td><td></td><td>4</td><td>0</td><td>¥10</td></tr> </table>	SYMBOL	COIN	NAME	トウニユウ キンガク	SIZE	6BYTE	LV	NAME	ATTR	LN	BYTE	B	CONTENT	1	COIN			0	0	トウニユウ キンガク	2	Y100	INTE		0	0	¥100	2	Y50	INTE		2	0	¥50	2	Y10	INTE		4	0	¥10
MODULE	VML	PROC	MONEY																																																																							
NAM = COIN																																																																										
15	7	0																																																																								
(2)	(2)	(2)																																																																								
0	8	15																																																																								
Y100																																																																										
Y50																																																																										
Y10																																																																										
SYMBOL	COIN	NAME	トウニユウ キンガク	SIZE	6BYTE																																																																					
LV	NAME	ATTR	LN	BYTE	B	CONTENT																																																																				
1	COIN			0	0	トウニユウ キンガク																																																																				
2	Y100	INTE		0	0	¥100																																																																				
2	Y50	INTE		2	0	¥50																																																																				
2	Y10	INTE		4	0	¥10																																																																				
<table border="1"> <tr><td>MODULE</td><td>VML</td><td>PROC</td><td>MONEY</td></tr> <tr><td colspan="4">NAME = RETC</td></tr> <tr><td>15</td><td>7</td><td>0</td><td></td></tr> <tr><td>(2)</td><td>(2)</td><td>(2)</td><td></td></tr> <tr><td>0</td><td>8</td><td>15</td><td></td></tr> <tr><td colspan="4">RETC</td></tr> </table>	MODULE	VML	PROC	MONEY	NAME = RETC				15	7	0		(2)	(2)	(2)		0	8	15		RETC				<table border="1"> <tr><td>SYMBOL</td><td>RETC</td><td>NAME</td><td>リターンコード</td><td>SIZE</td><td>2BYTE</td></tr> <tr><td>LV</td><td>NAME</td><td>ATTR</td><td>LN</td><td>BYTE</td><td>B</td><td>CONTENT</td></tr> <tr><td>1</td><td>RETC</td><td>INTE</td><td></td><td>0</td><td>0</td><td>リターンコード</td></tr> </table>	SYMBOL	RETC	NAME	リターンコード	SIZE	2BYTE	LV	NAME	ATTR	LN	BYTE	B	CONTENT	1	RETC	INTE		0	0	リターンコード																													
MODULE	VML	PROC	MONEY																																																																							
NAME = RETC																																																																										
15	7	0																																																																								
(2)	(2)	(2)																																																																								
0	8	15																																																																								
RETC																																																																										
SYMBOL	RETC	NAME	リターンコード	SIZE	2BYTE																																																																					
LV	NAME	ATTR	LN	BYTE	B	CONTENT																																																																				
1	RETC	INTE		0	0	リターンコード																																																																				
LINK	MODULE	PROC	DWN	ヒタチ タロウ	60/05/08	TITLE	Hitachi, Ltd.	SYSTEMS DEVELOPMENT LABORATORY DWG. NO.																																																																		
VX	VML	MONEY	CHKD			TABLE SPEC LIST	Tokyo Japan	11- 1																																																																		
			APPD																																																																							

図5 出力ドキュメント例(テーブル仕様) テーブル仕様書は、割付図とデータ要素の一覧表で構成する。

```

DIVIDE : PROCEDURE(M, VAR DTAB) ;
DCL M      INTEGER ;
DCL DTAB
    STRCT(Y100  INTEGER,
           Y50   INTEGER,
           Y10   INTEGER) ;
DTAB. Y100=0 ; DTAB. Y50=0 ; DTAB. Y10=0 ;
/* : ¥100ダマ カウント */
DO WHILE(M>=100) ;
    DTAB. Y100=DTAB. Y100+1 ;
    M=M-100 ;
END ;
/* : ノコリ >= ¥50 */
IF M>=50
/* : ¥50ダマ カウント */
THEN DO ;
    DTAB. Y50=DTAB. Y50+1 ;
    M=M-50 ;
END ;
/* : ¥10ダマ カウント */
DO WHILE(M>=10) ;
    DTAB. Y10=DTAB. Y10+1 ;
    M=M-10 ;
END ;
END DIVIDE ;
                
```

ソース プログラム(S-PL/H)

概要 P A D

図6 高級言語の概要PAD生成例 プログラムの構造に基づいて概要PAD図を生成する。

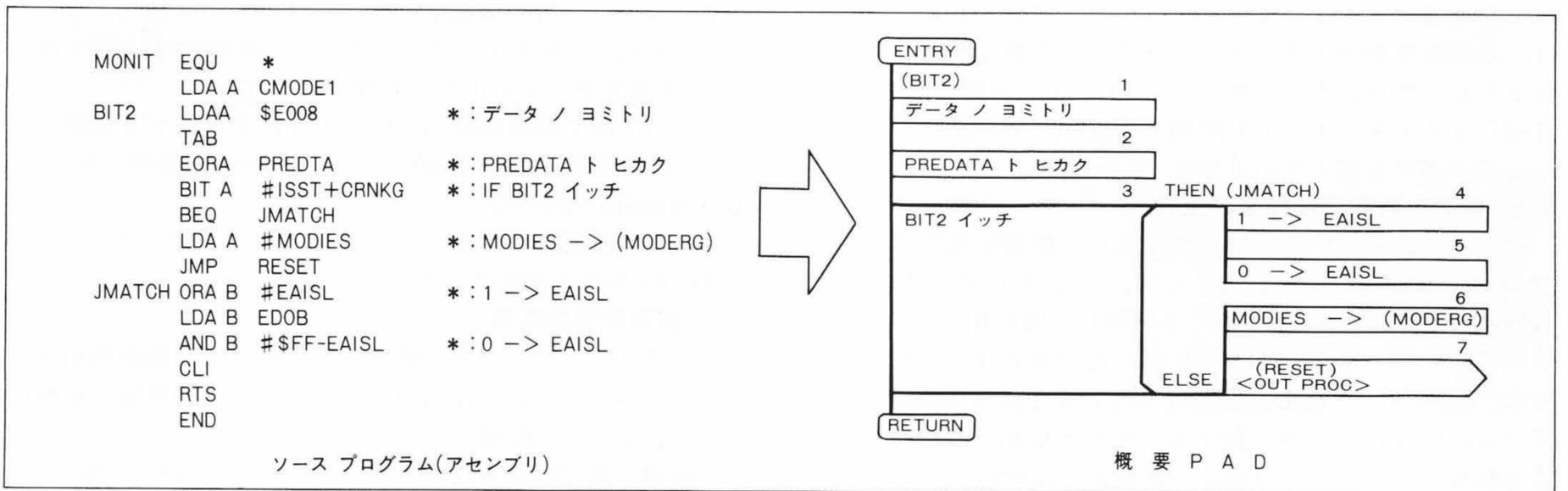


図7 アセンブリ言語の概要PAD生成例 プログラムの静的構造を解析し構造化を図り、概要PAD図を生成する。

などに該当するオペレーションコード、オペランド構成などが、各言語仕様間で異なっている。

本システムでは、対象言語の拡張性を実現するために、上記の項目に関する情報を、アセンブリ言語仕様ごとにシステムに登録する方式を採用している。

解析の対象とするソースプログラムの種別の指定により、システムは該当言語仕様の登録結果に基づいて入力プログラムの解析を実施し、ドキュメントを生成する。

現在、日立、モトローラ、インテル、ザイログの主要なアセンブリ言語仕様を登録しているが、登録は利用者が簡単に行なえるように、定形化されている。登録の追加によって、容易に新規言語によるプログラムの入力が可能である。

4 適用評価

本システムは昭和59年4月、基本機能を開発し、以降順次機能拡大を図り、現在もなお利用に伴う意見を反映した改良を継続している。これまで、OA機器関連ソフトウェア、電子交換機ソフトウェア、マイクロコンピュータ用基本ソフトウェア、銀行端末ソフトウェアなど広い分野にわたって300kS/月のプログラムに適用され、大きな成果を挙げている。以下に適用評価について述べる。

(1) 高級言語プログラムへのコメント挿入

本システムで設定したコメント記述ルールによらない既存のプログラムに対して適用評価を行なった。これらのプログラムに挿入されているコメントのうち、実行文中に挿入されたコメントに対して概要PADコメント識別子を追加するだけで、すべてのドキュメントを生成することができた。この結果は、既存プログラムの再利用促進に対する強力なツールになると確信できるものであった。

(2) アセンブリ言語プログラムへのコメント挿入

本システムでは、アセンブリ言語プログラムの構造を忠実に反映したPAD図を自動生成するので、プログラムの構造を意識しないでコメントを挿入すればよい。

したがって、PADによる設計に基づいてプログラミングした場合には、コメントの挿入も容易であり、理解しやすいPAD図が生成される。PAD以外の手法によった場合には、構造化の度合いは低く、理解しにくいPAD図が生成される。

理解しやすく保守しやすいプログラムを作成するためにも、PADによる設計を進めることが必要であり、それによって、本システムの機能を十分に活用することができる。

(3) その他

本システムの対象となる言語拡張の要望が強く、現在まで

に高級言語では5種、アセンブリ言語では7種の言語仕様への対応を可能としている。特にアセンブリ言語については、利用元で言語仕様の登録が行なわれており、言語拡張性が立証された。また、利用形態もプログラム再利用、共通サブルーチンの活用促進、プログラム開発時のレビュー、プログラムのデバッグ、保守と多様な目的に広げられ適用性が十分認識された。

なお、本システムは社内14事業所の協力と、日立ソフトウェアエンジニアリング株式会社の支援を得て開発したものである。

5 結 言

マイクロコンピュータ用ソフトドキュメント自動生成システムAuto-DSについて、その機能概要と特長を中心に述べた。

本システムはソフトウェア一貫生産システムICAS (Integrated Computer Aided Software Engineering System)⁶⁾のドキュメント支援機能に位置づけられるものであり、特に、ツール利用時の簡易性、各種言語に対する適用拡大の容易性という特長によって、実用性の高いシステムとして日立製作所内で広く利用されている。

今後は、ツール利用のノウハウを支援機能に更に反映し、より良いシステムへと改良を続けてゆく考えである。

参考文献

- 1) 青山, 外: ソフトウェア生産の諸問題とその生産技術の動向, 日立評論, 62, 12, 847~852(昭55-12)
- 2) 鈴木, 外: ソフトウェア構造設計支援システム“ADDS”, 日立評論, 66, 3, 185~188(昭59-3)
- 3) 橋本, 外: ソフトウェア保守支援のためのドキュメントツール Auto-DS(機能と実現方式の概要), 情報処理学会第29回全国大会論文集, 447~448(昭59-9)
- 4) 峯尾, 外: ドキュメント作成支援システム(Auto-DS), 電子通信学会通信部門全国大会論文集, 439~440(昭59-11)
- 5) 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, 21, 4, 259~267(昭55-7)
- 6) 小林, 外: ソフトウェア一貫生産システム“ICAS”基盤技術の確立, 日立評論, 66, 3, 171~176(昭59-3)
- 7) 鶴飼, 外: ソフトウェア保守支援のためのドキュメントツール Auto-DS(概要論理仕様図生成方式), 情報処理学会第29回全国大会論文集, 449~450(昭59-9)
- 8) 鶴飼, 外: ソフトウェア保守支援のためのドキュメントツール Auto-DS(アセンブリ・プログラムの概要論理の構造表現), 情報処理学会第30回全国大会論文集, 557~578(昭60-3)