

ソフトウェア構造設計支援システム “ADDS”

Software Design Support System “ADDS”

ソフトウェアの大規模化、複雑化に伴い、ソフトウェア開発での生産性及び信頼性の向上が重大な課題となっている。日立製作所では、この課題を解決するために、ソフトウェア開発支援システム(CASDシステム)を開発し運用してきている。特に設計段階では、構造設計支援システムADDSによって設計作業を総合的に支援し、生産性及び信頼性の向上を図っている。

ADDSの入力は、構造化設計法に基づいたモジュール仕様記述言語MDLである。設計者は、MDL言語に対応した会話形ワークシート画面から設計仕様情報をADDSに入力して、仕様の解析及び設計ドキュメントの作成を自動化し、設計作業を効率的に進めることができる。

鈴木啓之* Hiroyuki Suzuki

橋本忠雄* Tadao Hashimoto

桜井春雄* Haruo Sakurai

野木兼六** Kenroku Nogi

1 緒 言

従来日立製作所では、ソフトウェア生産技術を生産管理技術及び生産支援技術の両面から改善してきた。ソフトウェア開発の近代化のため、総合的なCASDシステム(Computer Aided Software Development System:ソフトウェア開発支援システム)¹⁾を開発し、日立製作所工場で運用してきている。CASDシステムはソフトウェア生産の全体を支援するために設計支援、プログラム支援及びテスト支援の各サブシステムで構成し、各サブシステムは支援治工具で実現している。

ADDS(Automated Design and Documentation System:構造設計支援システム)は、このCASDシステムの設計支援サブシステムを構成する治工具である。

本論文ではADDSについて、その概要、ADDSで用いられる設計仕様記述のための言語、設計仕様情報データベースへの入力機能、各種設計ドキュメントの自動作成機能などを中心に紹介する。

2 ADDSの概要

ADDSは、MDL(Module Design Language:モジュール仕様記述言語)と図1に示す処理系で構成している。設計仕様

情報は、MDLの記述又はワークシート画面によってデータベースに入力、蓄積する。ADDSはこの蓄積した設計仕様情報を解析し、設計レビューや保守に有用な各種設計ドキュメント及びソースプログラムのコメントにできるMDLを自動作成する。ADDSは、これらの機能によって構造設計工程を総合的に支援している。

3 モジュール仕様記述言語

ADDSでは、構造化設計法²⁾に基づいて設計した設計仕様情報をMDLで表わす。このMDLを用い、文章や図表であいまいに表現されていた設計仕様情報が、漏れの無い正確なものとなり、仕様解析、設計ドキュメントの自動作成が可能となる。

このMDLは表1に示すように五つの仕様から構成され、モジュールと呼ばれるコンパイル単位ごとに記述する。MDLの文法はキーワード方式であるが、特に内部仕様では段階的詳細化が容易にできる擬似コードを用いている。また、MDLはソースプログラムのコメントとして展開できるように工夫しており、設計仕様情報とソースプログラムの対応付けが可能となっている。この一体化によって保守時の設計仕様情報の

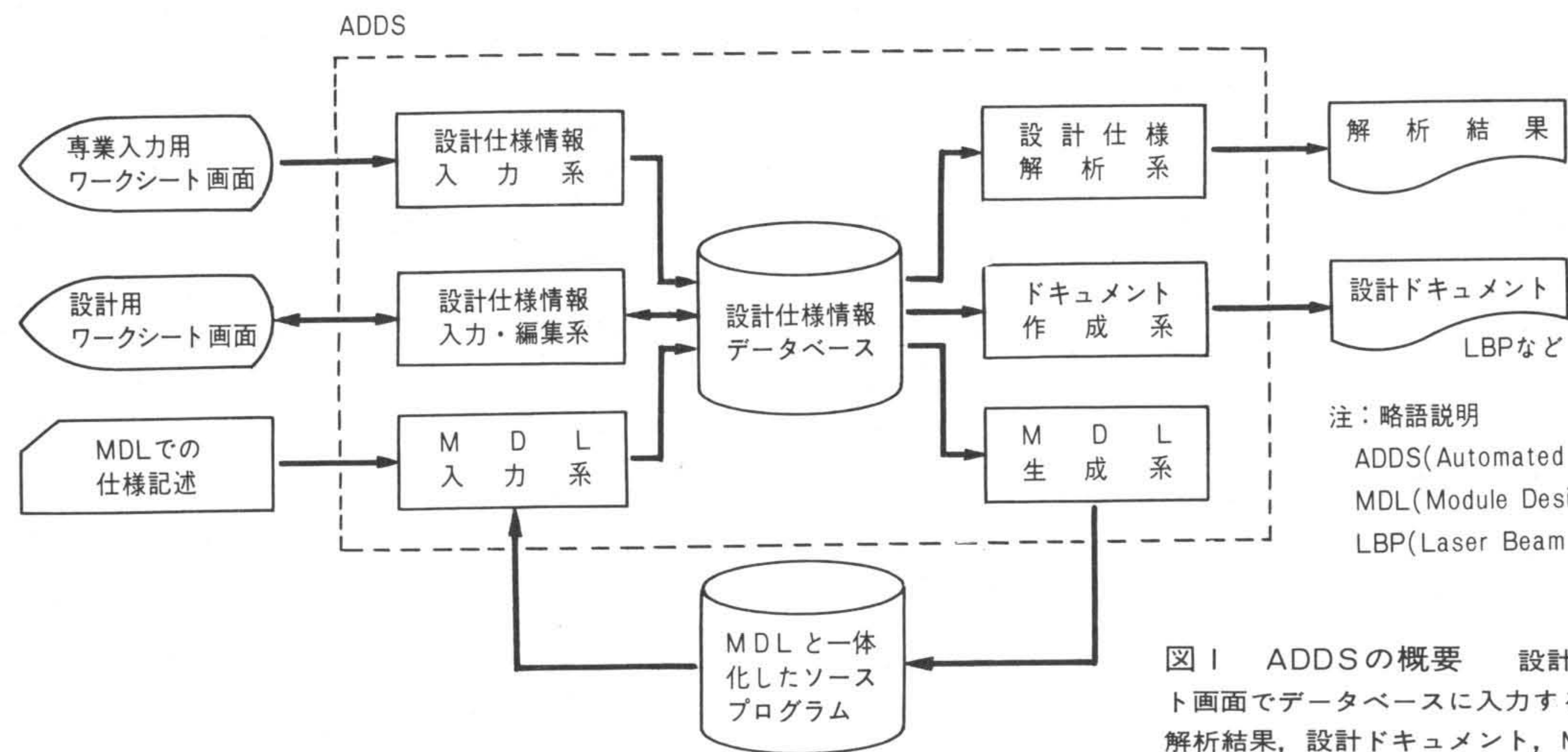


図1 ADDSの概要 設計仕様情報はMDLの記述又はワークシート画面でデータベースに入力する。データベースの情報から設計仕様の解析結果、設計ドキュメント、MDLが得られる。

* 日立製作所ソフトウェア工場

** 日立製作所システム開発研究所

表1 MDLの構成 MDLは五つの仕様から構成される。

項 目	内 容
概 略 仕 様	モジュールの名称、機能など
管 理 仕 様	モジュールのシステム名、作成日付など
関 係 仕 様	モジュールの階層関係、他モジュールとの関係
外 部 仕 様	モジュールのインタフェース情報
内 部 仕 様	モジュールの処理又はデータ構造の擬似コード

更新漏れを防ぐことができる。

3.1 外部仕様などの記述

MDLでは、モジュールの概要、他モジュールとの関係などを、概略仕様、管理仕様、関係仕様及び外部仕様で記述する。概略仕様では、モジュール名称、機能概要などの設計仕様情報をNAME、FUNCTIONなどのキーワードで記述する。管理仕様では、モジュールが所属するシステム名称、作成日付、変更履歴などのモジュールの管理に必要な設計仕様情報をSTATUS、DATE、HISTORYなどのキーワードで記述する。関係仕様では、モジュール間の関係を記述する。すなわち、モジュールの階層関係、他モジュールとの制御関係、データの参照関係などをCHILDREN、EXTERNAL、RELATIONなどのキーワードで記述する。外部仕様では、他モジュールが当該モジュールを呼び出すときの条件、インタフェース情報及び入出力データなどをENTRY、LINKAGE、INPUT、OUTPUTなどのキーワードで記述する。

3.2 内部仕様の記述

MDLでは、モジュールの内部処理構造を内部仕様で記述する。構造設計法では、モジュールの内部処理構造設計に段階的詳細化の手法を用いる。人間の能力の範囲は限られており、一度に多くの事柄を考えると誤りが増すという特性がある。段階的詳細化では、モジュールの内部処理構造設計で、この特性を十分に考慮し、一度に思考する範囲を制限する。小さな範囲を段階的に詳細化して、誤りのない設計を進めてゆく

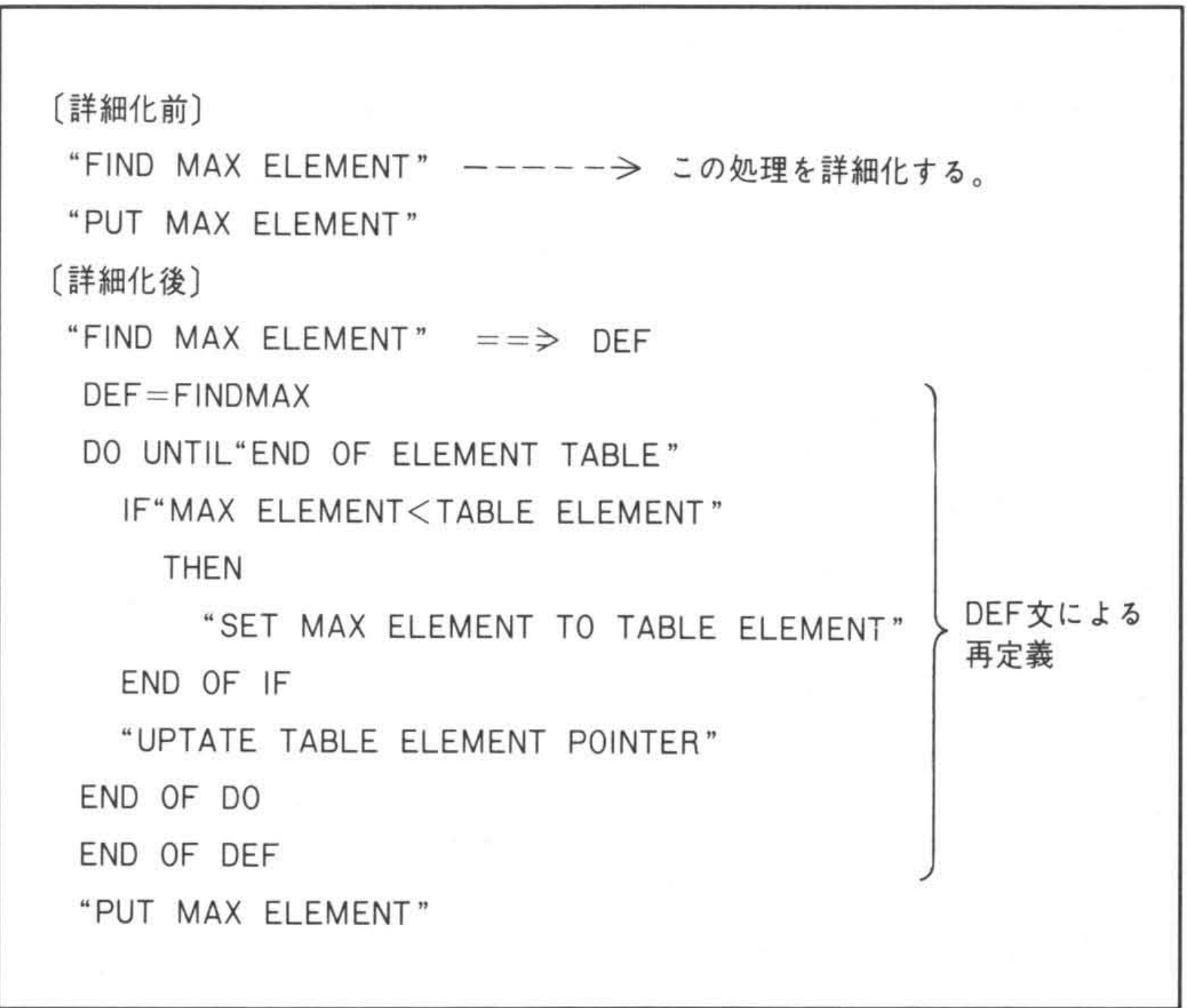


図2 MDLの擬似コードによる段階的詳細化 モジュールの内部処理構造設計では擬似コードを用い、段階的に詳細化してゆく。

ことができる。このように、段階的詳細化は人間の特性に合った優れた手法であるが、有効な支援環境がなかったため、適用が徹底できなかった。

しかし、MDLでは段階的詳細化を擬似コードで、効率的に進めることができる。図2に示すように段階的詳細化は、擬似コードの一つである再定義文(DEF文)で容易に実現できる。このDEF文で段階的に詳細化した部分を挿入しただけで、履歴情報を残しながら設計を進めてゆくことができる。

この擬似コードを用い、ADDSでPAD(Problem Analysis Diagram)³⁾を自動作成し、設計レビューをしながら、段階的詳細化によるモジュールの内部処理構造設計を効率良く進めることができる。

4 設計仕様情報の入力・編集

ADDSでは設計仕様情報をデータベースに蓄積してゆくが、この設計仕様情報を容易に入力し、編集できるように各種機能をサポートしている。

4.1 バッチ処理による入力

ADDSはMDLをバッチ処理で読み込み、設計仕様情報をデータベースに蓄積できる。このバッチ処理は、主にコメントとしてソースプログラム中に展開したMDLを入力したり、擬似コードで記述した内部仕様の入力に用いる。

4.2 会話形ワークシート画面による入力・編集

設計の初期段階では、設計仕様でMDLを記述し、設計レビューの結果を反映して、速やかに設計仕様情報を修正しなければならない。このMDLの新規作成入力及び設計仕様情報の修正を効率的に支援するため、ADDSは会話形ワークシート画面で設計仕様情報の入力・編集機能をサポートしている。

この会話形ワークシート画面による入力・編集機能の特徴は次のとおりである。

- (1) MDLの概略仕様、管理仕様、関係仕様及び外部仕様に対応するワークシート画面をサポートしている。
- (2) ワークシート方式では、必要なキーワードなどの項目をあらかじめ表示しているので、画面の空白を埋める単純な作業だけで、効率的に漏れの無い仕様を記述できる。
- (3) TSS(Time Sharing System)端末で、設計仕様情報を容易にかつ迅速に修正できる。
- (4) あらかじめ定めた用語は画面の下部にあるガイダンス番号を指定するだけで、設計情報として選択することができる。
- (5) 漢字の入力・編集も可能であり、日本語を設計仕様情報とすることができる。

図3にワークシート画面の例を示す。

4.3 専門オペレータによる入力

ADDSでは会話形ワークシート画面をサポートしているため、設計仕様情報が効率的に入力できる。しかし、日本人は端末キーボード操作が不慣れなため、大量入力にかなりの工数を要する。そこで、専門オペレータが設計仕様情報を入力し、効果的に設計作業を支援することができる。ADDSは特に専門入力に適したワークシート画面をサポートしている。

例えば、専門入力を利用する場合の設計手順は次のようになる。

- (1) 設計者は、構造設計法で設計し手書きワークシートに記入する。
- (2) 専門オペレータは、手書きワークシートを用い、入力専用ワークシート画面から設計仕様情報をデータベースに入力する。
- (3) 設計者は、ADDSの解析機能及びドキュメント作成機能

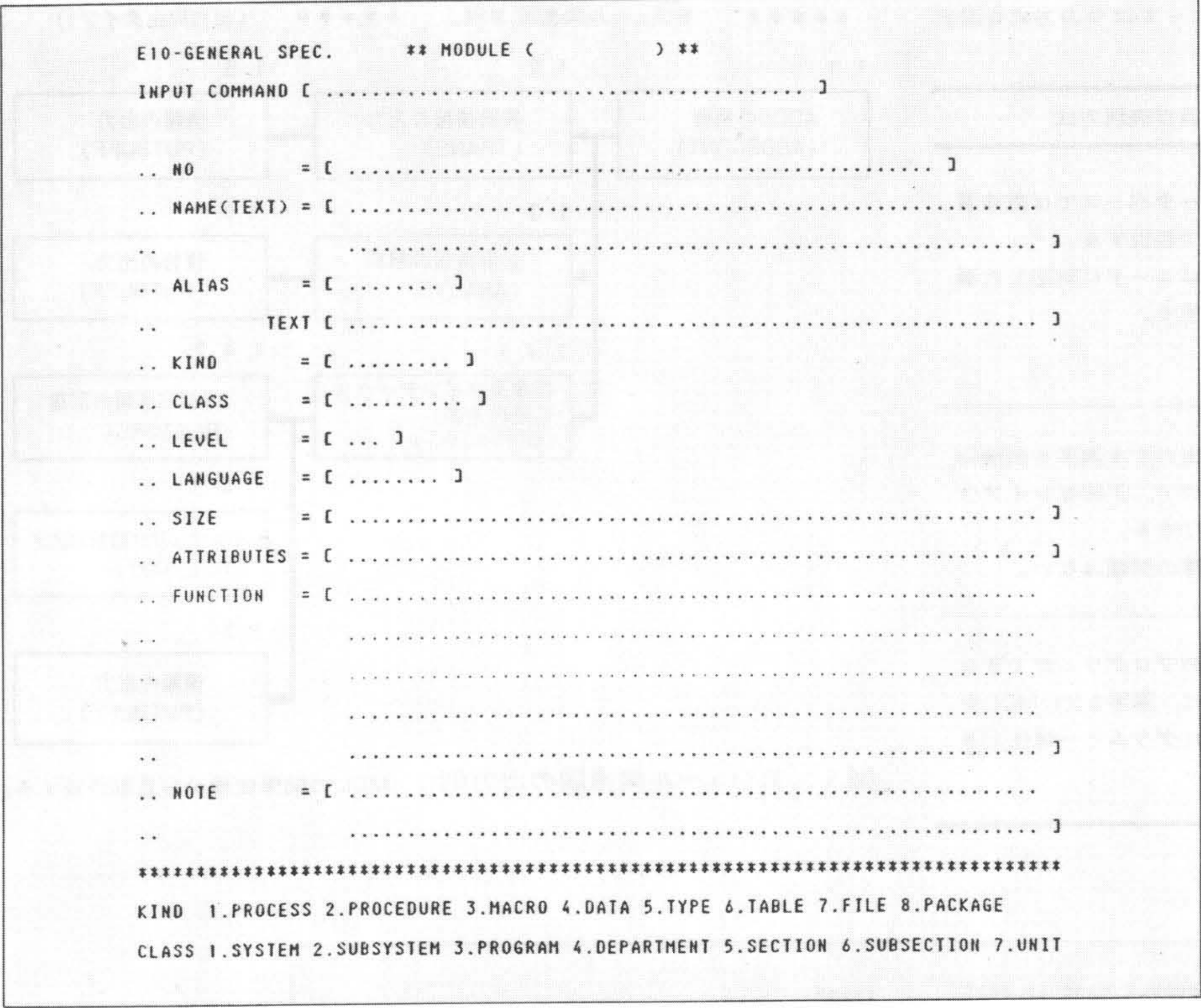


図3 ワークシート画面の例 設計者はワークシート画面を利用して、簡単に設計仕様情報を入力できる。

- で作成した資料を使い、設計レビューをする。
- (4) 設計者は、レビュー結果で、ユーザー入力・編集用会話形ワークシート画面の設計仕様情報を修正する。
 - (5) 以上の(3)、(4)を繰り返しながら設計を進める。
 - (6) 設計が終わった段階に、MDL生成機能でソースプログラ

ムのコメントを生成する。
専門入力をする場合の設計環境を図4に示す。

5 設計ドキュメントの作成

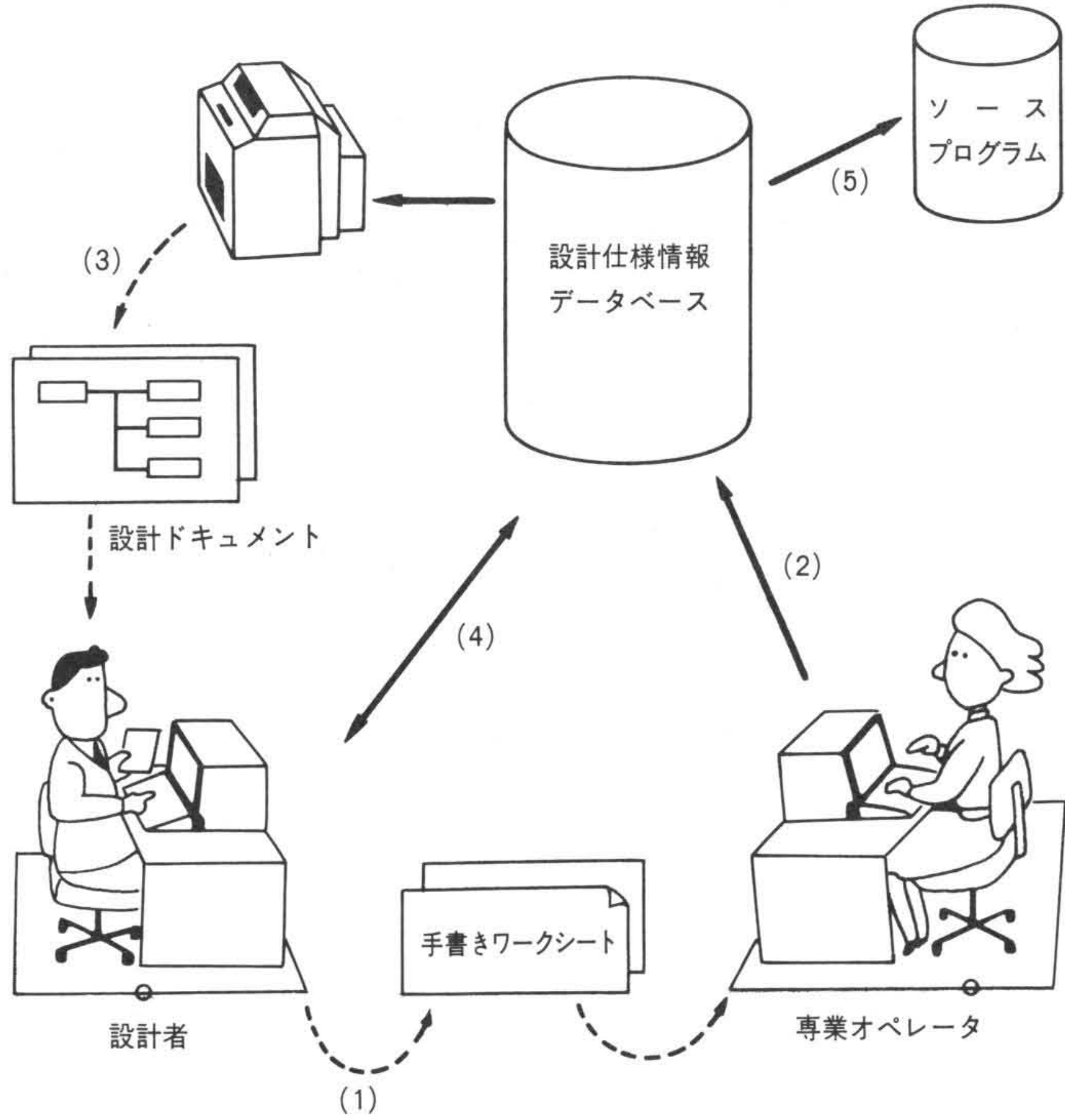
ADDSでは、前述のように各種設計ドキュメントを自動作成できる。

表2に各種ドキュメントの一覧を示す。
これらのADDSが自動作成するドキュメントの特徴は次のとおりである。

- (1) 高度に図形化してLBP(Laser Beam Printer)から出力する。
- (2) 設計仕様情報データベースから即時に自動作成できる。

表2 ADDSが自動作成する設計ドキュメント 設計者はこれらの設計ドキュメントを、設計レビューに活用する。また、保守書としても利用できる。

No.	ドキュメント名	形式	内 容
1	機能階層構造	図・表	各モジュールの階層構造
2	モジュール仕様	表	各モジュールの仕様
3	テーブル仕様	表	各テーブルの仕様
4	ファイル仕様	表	各ファイルの仕様
5	モジュール関連	図・表	モジュール間の制御関係
6	モジュール一覧	表	モジュールの概要一覧
7	機能一覧	表	モジュールの機能一覧
8	変更履歴一覧	表	モジュールの変更履歴の一覧
9	データフロー	図	モジュールの入出力と処理手順の図示
10	P A D	図	モジュールの処理手順の構造的な図示



注：(1)～(5)は設計手順(本文参照)

図4 専門入力をする場合の設計環境 設計者は大量に入力するときには専門オペレータを利用する。専門入力用ワークシート画面は、入力専用として簡素化している。

表 3 ADDSの日本語出力方式 利用状況によって二つの方式を選択できる。

方式	仮名表現方式	直接表現方式
特徴	MDL, データベースでは従来のコード体系である仮名文字で表現する。 出力時に仮名漢字変換をし, 漢字で出力する。	MDL, データベースでは直接漢字コードで表現する。 出力時にはコードに対応した漢字を出力する。
利点	MDLが従来のコード体系だけで表わせるため, 通常のプログラム言語との一体化ができる。	入力時に出力する漢字を直接指定できるので, 正確なレイアウト指定ができる。 同音異義語の問題はない。
欠点	MDLと出力ドキュメントでのレイアウトが一致しない。 同音異義語の選択に工夫が必要である。	日本語でのプログラムができない言語では, 漢字を含むMDLをソースプログラムと一体化できない。

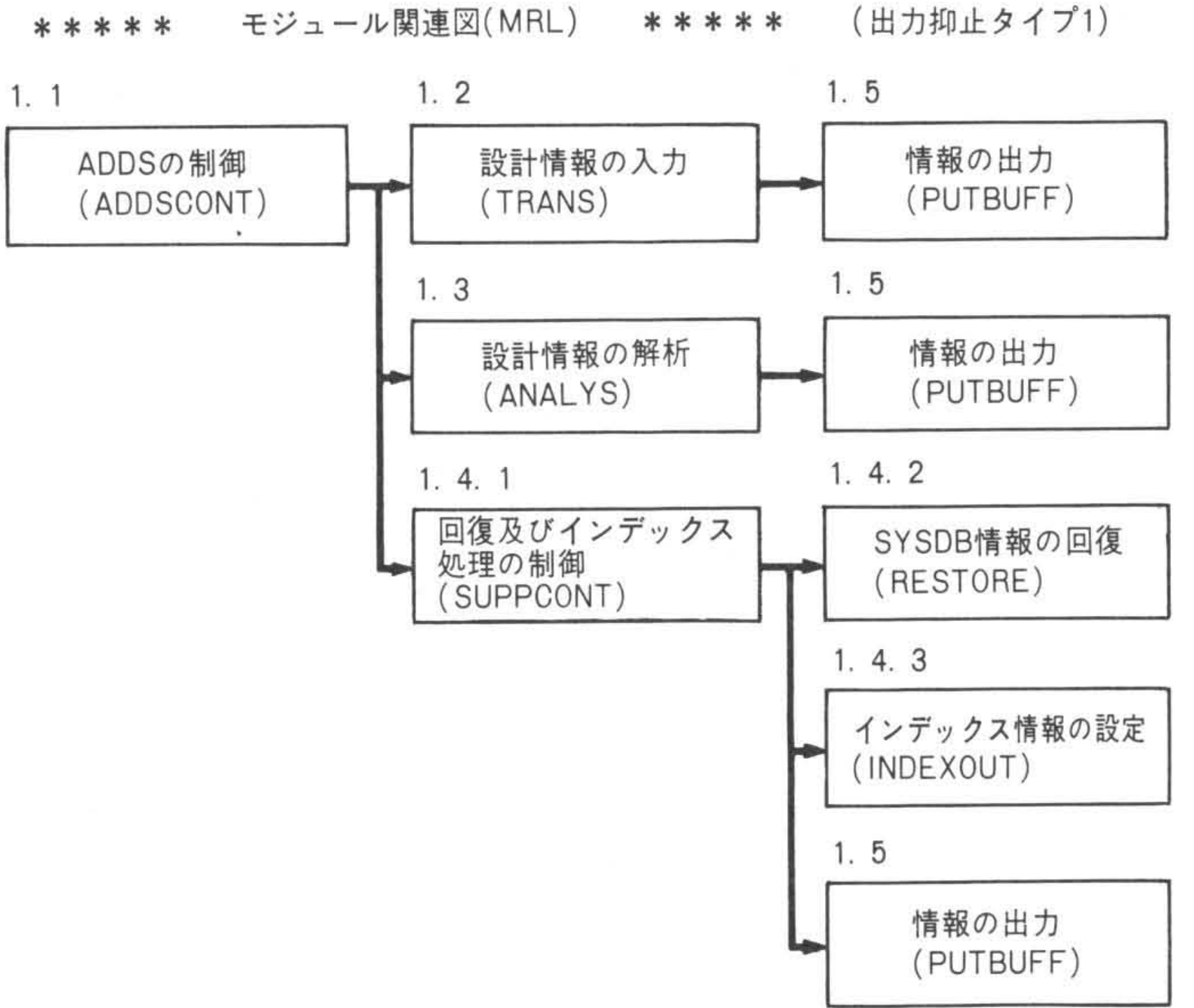


図 5 モジュール関連図の出力例 MDLの関係仕様から自動作成する。

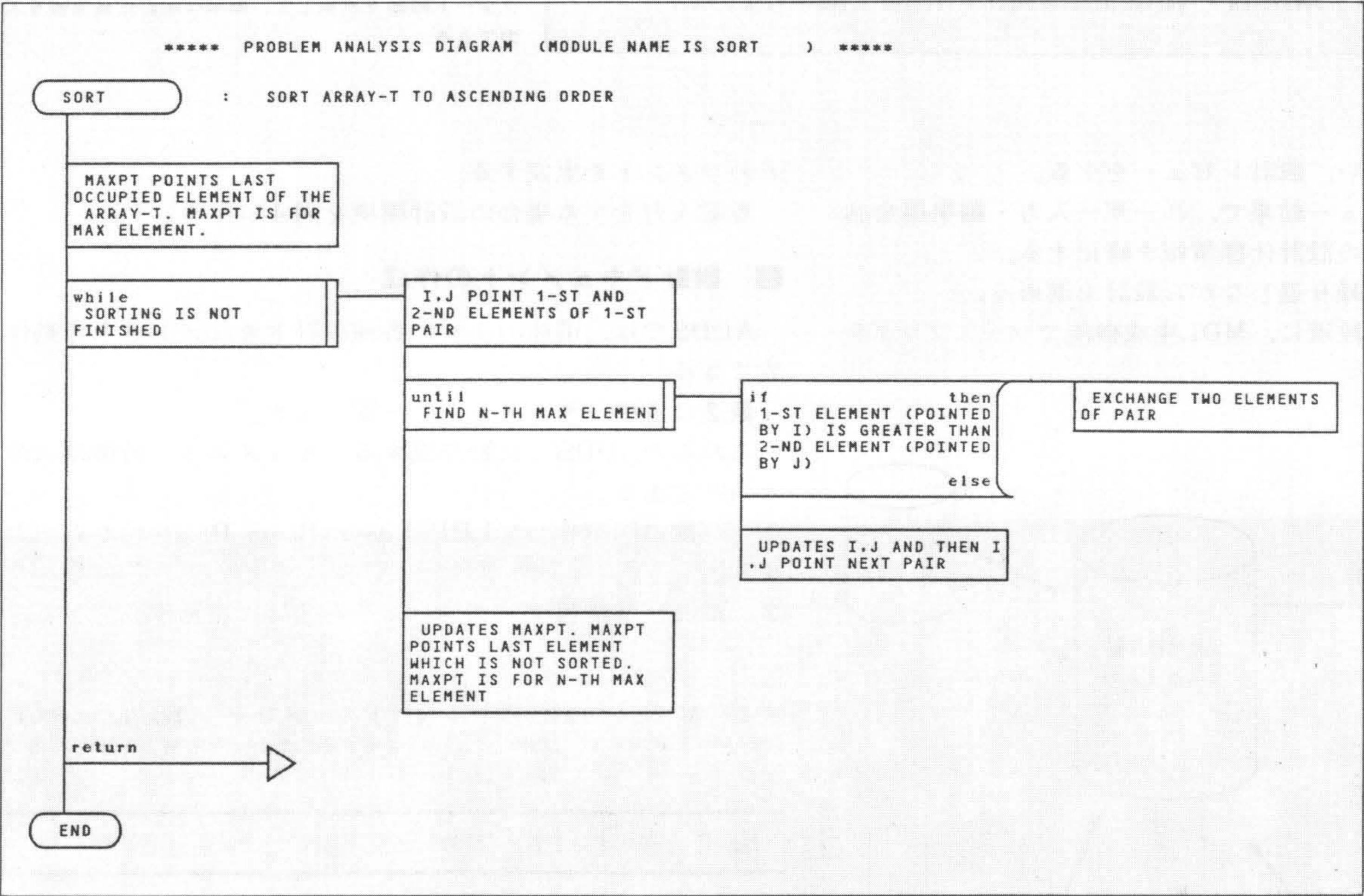


図 6 PADの出力例 MDLの内部仕様から自動作成する。

(3) 英語及び日本語で出力できる。特に日本語設計ドキュメントの出力方法は、表 3 に示すように使用条件によって、仮名漢字表現方式又は直接漢字表現方式が選択できる。

モジュール関連の図形形式の出力例を図 5 に、PAD出力の例を図 6 に示す。

6 結 言

ADDSは設計支援治工具であり、大規模なシステム開発に適用できる。ADDSを用い設計工数を半減することができた。また、設計仕様情報を更新するときにもその工数を大幅に削減し、信頼性を向上できる。

今後、内部処理構造設計の会話形支援機能の充実及び段階的詳細化手法のデータ構造設計の拡張を検討してゆく予定である。

参考文献

- 1) 片岡, 外: ソフトウェア開発支援システム(CASDシステム), 日立評論, 62, 12, 879~882(昭55-12)
- 2) 片岡, 外: ソフトウェア構造設計技法, 日立評論, 62, 12, 853~856(昭55-12)
- 3) 二村, 外: プログラムの木構造化図面“PAD”, 日立評論, 62, 12, 871~874(昭55-12)