

機能展開知識を基にしたソフトウェア設計支援システムの検討

石川 和彦 蓮田 広保
(株) 東芝 システム・ソフトウェア技術研究所

ソフトウェア生産工業化システム I M A P ではあらかじめ整備された部品を組み合わせてアプリケーションプログラムを合成することを目ざしている。我々は本システムの中核技術である部品化設計支援ツール群の開発を進めている。この中には設計書エディタ、部品検索ツール、部品管理ツール、ソースコードジェネレータが含まれる。機能展開知識を用いながら、設計技術法 T F F に従って、トップダウンに設計を進めていく。この際、あらかじめ用意した部品群をできるだけ利用できるように設計者を誘導するところがポイントである。

本方式による設計支援システムは特定の分野だけを対象とせず、幅広い問題領域におけるプログラム開発に有効であると考えられる。

Software design support system based on the " function breaking down knowledge "

Kazuhiko Ishikawa, Hiroyasu Hasuda

System and software engineering laboratory, Toshiba corporation
70 yanagi-cho saiwai-ku kawasaki, kanagawa, 210 Japan

I M A P aims to synthesise the application programs by assembling the software components that are guaranteed their quality.

We are developing the software design support system which includes T F F editor, module retrieving tool, module managing tool and source code generator.

This system, which is a subsystem of I M A P has the knowledge which breaks down software modules' function.

This knowledge helps software designer to design based on guaranteed components This system supports software designers in many application fields.

1. はじめに

我々はソフトウェア生産工業化システム I M A P (Integrated software management and production support system)を開発してきた。I M A Pにおけるソフトウェア生産の工業化とは生産過程を工程に分割し、各工程内の作業基準を規定し、かつ品質基準を満たした製品を指定期間内に一定工数で生産できるようにすることである^[1]。

これを実現するために

- (1) 管理手法の確立
 - (2) 作業手順の明確化と分業化
 - (3) 部品化・再利用の促進
 - (4) 生産環境の確立
 - (5) 技術者育成のための教育体系の確立
- を重視している。

この内、特に部品化・再利用の促進についてはハードウェアと同様に、ソフトウェアにおいてもあらかじめ整備された標準部品の存在が工業化の前提となる。

2. I M A Pにおける部品とは

同じ分野の類似ソフトウェアの生産において既存ソフトウェアの流用あるいは繰り返し利用される部分を切り出して再利用することは従来から行なわれてきた。特に同一組織による類似ソフトウェア作成の場合、60%以上再利用可能なシステムもある。しかし、切り出された部分を他の組織においてソフトウェア生産に再利用しようとする、切り出された部分の理解や修正に手間取って新規に生産するよりも工数がかかったり、トラブル時に対応できないことが多い。

I M A Pにおける標準部品とは、既存のソフトウェアからの切り出しではなく、再利用することを前提にした部品を標準部品規定に基づき事前に作成し、品質を保証し、正式な手続を踏んで認知された部品のことである。

この標準部品規定には部品仕様記述法、品質基準と検査方法は当然として標準部品を認知する正式機関の位置付け、構成メンバーの運用規定も含まれている。

3. ソフトウェア設計のプロセス

ソフトウェアの設計技法としてはトップダウン設計法、データ抽象化設計法、構造化設計法等いく通りか存在する。我々はこのうち、部品再利用による設計に適した技法としてトップダウン設計法に焦点を当てた。この設計プロセスを図1-aに示す。

設計者はまず要求される仕様を機能として表現し、次にこの機能を実現するためのいくつかの下位機能に展開する(機能展開)。この機能展開をくり返していき、その結果得られる個々の要素機能に合致する部品が存在するかをチェックする。

このとき展開された下位機能群と個々の部品との対応付けが必要になり、部品検索機能が要求されるが、機能展開は一般に部品の存在を意識することなく実行されるため、部品の再利用がなかなか進みにくい。

すなわち、アプリケーションプログラムを設計してか

らその仕様にあった部品をさがすために、なかなか合致する部品が得られないのである。

図1-bにI M A Pにおけるソフトウェア設計のプロセスを示す。機能展開の段階から、あらかじめ整備された部品との対応付けがとれるように設計支援システムが誘導していくことにより、部品の再利用率を高めることができる。このためにはシステムに機能展開の知識を持たせることが必要になる^[2]。

4. 部品仕様記述方式

トップダウンに設計していく際に、既存の部品を効果的に再利用するためには仕様記述の方法、つまり部品の機能仕様及び設計者の要求仕様を正確に記述することも重要なポイントである。

従来型の自然言語によるソフトウェア仕様記述では次のような問題がある。

- (1) 一般に部品の機能仕様は各部品の作成者が記述するが、これらは表現が不統一でわかりにくい。
- (2) 部品の利用者、すなわちアプリケーションプログラムの設計者が必要とする部品の機能を表現する際に用いるキーワードは、過去の経験の違いによりかなり異なったものになる。

つまり部品利用者側の機能仕様記述と部品作成者側のそれには大きな違いがある。これを言葉の言い換えなどの複雑な交換規則を順次適用して検索を行っていた。表現の自由度が大きく、利用者にとっても作成者にとっても負担が大きいのということがこのような結果を生んでいた。

我々は両者双方が部品機能仕様を統一した概念で扱うことができる記述形式を検討し、3対象1操作モデルとしてまとめた^[3]。

本モデルの特徴をまとめると次のようになる。

- (1) 部品利用者から見て
 - ・部品の機能に対して“何をどうするのか”が明確であり、類似部品があっても相互の違いが明確な記述形式である
 - ・直観的に見ることができるよう、入力/出力データやパラメータが明示されている
 - ・要求レベルに応じた鮮明度で部品機能仕様が参照できること
- (2) 部品仕様記述者から見て
 - ・部品機能仕様を記述する時に、曖昧さ・記述不足が生じないように、記入の指針がある
 - ・経験の差異が出ないように、複数の記述者が同一概念で統一した表現ができる
 - ・利用者側の要求する詳細度レベルに対応できるように、詳細情報を段階的に記述できる

ポイントは、部品が扱う対象(データ)に対する操作を中心に据えたことである。フィルモアの格文法^[4]の概念を基にして3対象1操作モデルフォームシート(図2)にまとめた。仕様はこのシートに従って上から下へ順次、部品設計者が書き込んでいく。

対象1、対象2、対象3はそれぞれ、入力データ、作用・対比データ、出力データのことである。これらに操

作を加えて機能の概要を表わす。

更にこの4つのキーワードの詳細情報として各属性を記入する。

5. 機能展開知識

3章で述べたような設計プロセスを実現するために必要となる知識である。すなわち、ある機能抽象度のレベルで表わされたひとつの機能を、これを実現するためのいくつかの要素機能群に展開する知識である。

既に整備されている部品群が備えている機能を意識しながら機能を展開するところにポイントがある。

各機能は3対象1操作モデルに従って表現する。機能展開知識の1例を図3に示す。

本知識には展開された各要素機能をどう組み合わせたら元の機能を実現できるのかを示す情報も持たせる。つまり各要素機能をつなぐ順序及びそのつなぎ方（シーケンシャルにつなぐのか、くり返しが入るのか、分岐させるのか）である。

例ではグラフの座標軸の部分を作成し画面に表示するという機能を実現するためには、まず、座標軸の目盛幅や高さ等のパラメータを計算によって求め、次に、座標軸の各端点を表わす座標値の点列（2次元）を作成し、最後に、点列を2点ずつラインで結ぶことを繰り返すことが必要になることを表わしている。もちろん、座標軸の作り方は他にも存在する。一般に元の機能を展開する案は複数存在する。事前に個々の要素機能に対応するような部品が用意されているかが大きく影響する。

展開された1つの要素機能がさらにもう1つ下のレベルのいくつかの要素機能群に展開できることもある（図4）。

例えば、機能Aが機能B、C、D、に展開されるとする。機能BとCには対応する部品が存在するが、機能Dは部品として用意されていないというケースもありえる。あらかじめ用意された部品だけであらゆる機能を実現させるのはむずかしいからである。機能Dがさらに下位の要素機能EとFにより実現されている。但し、機能Eに対応する部品は存在しても機能Fに対応するものはないということもある。このようなケースでは、機能Aを実現したい設計者としては、Dにあたる機能を自らコーディングするか、機能Eに対応する部品を使って機能Fの部分コーディングするという2通りの方法がある。

展開する前の元の機能（機能A）の抽象度の最上位のレベルとして想定しているのは、1つのタスクレベルである。言い換えれば、機能展開知識によって支援するのはタスクの設計及びタスク内の各モジュールの設計である。

6. インプリメントの方法

6. 1 3対象1操作モデル

4章で述べたフォームシートによって示される3対象1操作モデルによる機能仕様表現を設計支援ツールとしてどのように実現するかについて述べる。要求仕様表現

と部品仕様表現のうち、後者を中心に説明する。

3対象1操作モデルは部品の機能仕様と設計者の要求仕様を形式的にかつ宣言的に記述するものである。形式的、宣言的に記述するにはフレーム型知識表現が適している。そこでエキスパートシステム構築ツールを用いて部品仕様表現フレーム群と要求仕様表現フレームを作成し、これらの中に3対象1操作モデルの概念をできるだけ盛り込んだ。これらのフレームを作成するにあたっては以下の点に留意した。

- (1) 個々の部品毎に部品仕様をフレーム化する。
- (2) 1. によって得られる各フレームをグルーピングし、共通な情報をくり出してフレーム群を体系化する。フレーム群の管理をしやすくするためである。
- (3) 各フレーム毎に持っている操作と対象1～3及び各属性の値を一括して管理するしくみを設ける。

図5は部品仕様フレーム群の階層関係を表わしている。4種類のフレーム群を用意した。各フレーム群はいずれもis-a関係で結ばれている。

(1) 部品仕様フレーム

最上位に存在するのは、部品仕様フレームである。このフレームは、3対象1操作モデルの枠組を表現したものである。3つの対象とその属性及び操作とその属性それぞれにスロットを割り当てる。但し、スロット値は、まだ入らない。これらのスロット名は、下位の3種類のフレーム群全てに継承される。

(2) 代表語別フレーム群

部品仕様フレームの1つ下位に位置するのが、代表語別部品仕様フレーム群である。3対象1操作モデルでは、1つの操作と最大3つの対象を用いて部品の仕様を表現する。実際には、操作名が定まればこれが持つ対象のパターンも一意に定まる。

そこで同じ対象パターンを持つ操作名をグルーピングし、各グループ毎に1つの代表語を定めてこの下に各操作名を関連付けた。従って代表語別部品仕様フレームでは、操作名スロットに代表語が入る。対象パターンは、本フレームより下位のフレーム全てに継承される。

代表語別部品仕様フレーム群の1つ下位に位置するのが操作別部品仕様フレーム群であり、更にもう1つ下、つまり最下位には個別部品仕様フレーム群が存在する。

(3) 個別部品仕様フレーム群

個別部品仕様フレームには、個々の部品仕様を、3対象1操作モデルの枠組に従って記述されている。

(4) 操作別部品仕様フレーム群

機能概要のうち操作名に注目して、同じ操作名を持つフレーム群を1つのグループにまとめ、これらの子フレームとする親フレームを操作名ごとに1つずつ設けた。これが操作別部品仕様フレーム群である。（図6）。このフレームには、対象1から対象3の各スロットと各属性値スロットに関して値とデーモンがセットされている。このうち値は、それぞれの子フレーム群の、対応するスロットに入っている値を集計し、一番頻度の高い値をdefault値として設定したものである。

以上のフレーム群を設計支援に具体的にどう役立たせるかについては7章に述べる。

6. 2 機能展開知識のインプリメント

次に、5章で述べた機能展開知識を具体的にどのように表現し、設計支援ツールにインプリメントするかについて述べる。

本知識は、最終的にはプログラムとして実現される各種の機能仕様のうち、互いに関連するものをグルーピングして表現したものであると見ることができる。

従って宣言的な記述形式がやはり、適している。

そこで機能展開フレーム群を作成し、この中に3対象1操作モデルの枠組を表現した。

図7にこの一例を示す。図3に表わした内容をインプリメントしたものである。

このフレームの構成を説明する。図7中、左のフレームは、フレーム名が“機能022”となっている。フレーム名は、展開する前の1つの機能を表わすidである。スロットは3種類ある。

1つめは、展開する前の機能を表わす対象名、操作名を値として持つスロット群である。機能022フレームでは“座標軸”を“表示する”がこれにあたる。

2つめは、展開した後各要素機能を表わすid及び各要素機能をつなぐ順序とそのつながり方（シーケンシャルにつながるのか、繰り返しが入るのか、分岐させるのか）を教えるスロット群である。

機能022フレームでは、1番目の要素機能は機能113というidによって表わされ、これはシーケンシャルにつながるということの意味している。このフレームと全く同じ枠組を持つ機能113というフレームが別に存在する。このフレームのスロット値を見れば、フレームは“目盛パラメータ”を“算出する”という機能を表わしていることがわかる。

3つめのスロットaboveは5。のところではふれなかった。機能展開知識をインプリメントする際に新たに付け加えたものである。次のような意味を持つ。

スロットaboveは、展開する前の機能を要素機能群の一部とするような、より抽象度の高い機能を表わすidを値として持つスロットである。機能022フレームでは、機能008が入っている。機能088フレームを見れば、“グラフ”を“表示する”であることがわかり、この機能の要素機能として、機能022の他に機能013と機能035があることもわかる。

以上でわかるように機能展開フレームでは、スロット値をキーとしたフレームの連鎖によって機能展開知識を表現している。機能展開フレームを設けたことによって得られる効果をまとめると次のようになる。

(1) 要素機能群の組み合わせ方を明確に表現することができる。

(2) 要素機能群の周りの機能とのつながり方を知ることができる。すなわち、要素機能群に対応する部品群を自らの設計に生かす際に、これらの組み合わせ方を知るとともに、周りの機能部分との関係が必要となる前後処理や押さえておくべき条件をつかむことができる。

7. 設計支援システムの構想

以上説明してきた機能展開知識及び3対象1操作モデルによる機能仕様表現法の他にも我々は部品化指向のソフトウェア設計記述法TFFについて研究してきた^[5]。

TFFの思想を盛り込んだツールTFFEDという設計記述用エディタの開発・改良を現在進めている。

又、TFFEDによって表わされた設計結果を設計者の力を借りながら、ターゲット言語のソースコードに変換するツールTCGENについても開発を進めている[6]。

これら諸々の技術を統合し、ソフトウェア設計支援システムとして1つにまとめることを検討している。ここではこのシステムの構想についてふれる。設計者が設計作業を進めていく際に、このシステムが行なう支援について、設計者とシステムの対話を通して述べる。

<対話内容>

| | | |
|---|----------|---|
| a | 起動 | まず設計者は本システムを起動し、TFFエディタCシート編集画面にする。この時点で設計者はこれから設計する処理型モジュールあるいはタスクの仕様（要求仕様）の概要はつかんでいるものとする。 |
| b | 処理ボックス作成 | 1 設計者は要求仕様をトップダウンに展開し、まず1つめの処理ボックスの作成にかかる（図8-b-1）。 2 システムは現在、登録されている操作名のリストを提示し、設計者に選択させる。このために部品仕様フレームの子フレーム名スロットに入っている値（代表語フレーム名のリスト）を用いる（図8-b-2）。 3 設計者は代表語を1つ選択する。システムはこの値に該当する代表語別フレームをアクセスし、子フレーム名スロットに入っている値（セレクトされた代表語別フレームを親フレームとする操作別部品仕様フレーム名のリスト）を表示する。 4 設計者は操作名を1つ選択する。選択したい値がない場合は設計者が単語を入力する。システムは5。と同様に該当フレームをアクセスし、対象1から対象3のスロットに入っているそれぞれのデフォルト値を順に提示し、設計者にこの値をそれぞれ対象名としてよいか確認する。否定された場合システムはデーモンを起動し、このフレームのフレーム群の対応するスロット値を取り出し、対象名リストを作成する（図8-b-4）。そして設計者に選択させる。選択したい値がリスト中になければ設計者が単語を入力する。以上で処理ボックス内の操作名と各対象名が定まる。この結果は3対象1操作モデルに従って一定の制約をつけた単文で処理ボックス内に表現される。 |
| c | 機能展開&検索 | 1 システムは定まった値をキーにして子フレーム群を検索し、マッチしたフレーム（個別部品仕様フレーム）の名を候補部品として処理ボックスのとなりに表示する。候補部品が存在しない場合は機能展開知識を利用する。 すなわち、先に定まった操作名と対象名を基に機能展開フレーム群をサーチし要素機 |

| | | |
|--------------------------------|---|---|
| | | <p>能群のフレーム名を得、それぞれの操作名、対象名を取り出す。これらの値を用いて先の処理ボックスを要素機能を表わすボックス群に展開して画面に表示する。この時、各要素機能を並べる順序や制御構造もTFFの記法に従って表現する(図8-c-1)。</p> <p>2 表示された展開の仕方ではいかを設計者に確認した後、各ボックスに該当する部品をそれぞれ検索し、候補部品の番号(フレーム名)として表示する。機能展開案は一般に複数存在する。確認時に設計者が否定した場合は第二案、第三案……として表示する</p> <p>3 展開案の中には要素機能群の一部に、対応する部品が存在しないものもある。この場合、候補部品番号は当然表示されない。その要素機能を更に展開する案が存在することがある。存在する場合には設計者に対し再度、その部分を展開するか否か選択させる。YESならば展開案を示し、候補部品を検索する。NOを選択した場合はこの要素機能は設計者自ら展開するか、このレベルで展開を止めておくかのどちらかになる。</p> <p>4 c-1において入力した操作名、各対象名で表わされる機能に対応する部品が存在せず、該当する機能展開案も存在しないケースもある。この場合にも、その機能を設計者が自らブレークダウンしていくかこれ以上展開せずこのレベルで止めておくかのどちらかになる。後者の場合はb-1に戻る</p> |
| d 仕様 確認 & 絞り 込み | 1 | <p>c-1、c-2、c-3のそれぞれにおいて該当する機能の候補部品として検索された部品の仕様を確認する必要がある。画面に表示されている候補部品番号を設計者がマウスで指定することによりシステムがサブウィンドウを1つ開きこの中にその部品仕様を表わすTFFBシート、Cシートを表示する(図8-d-1)。</p> <p>2 候補部品が多数存在することもある。この場合いちいちウィンドウを開くのはたいへんである。従って、システムが候補を絞り込むための質問を設計者に与えることになる。</p> <p>すなわち候補部品群に対応する機能を表わす操作及び各対象が持っている属性値をメニューにして表示し、設計者に選択させる(図8-d-2)。各候補部品の個別部品仕様フレームの各種属性スロットから値を取り出し、同じスロット名毎にリストにしてメニューとする。設計者が選択した値を用いて候補部品を絞り込む。</p> |
| e | | <p>以上の作業を繰り返し行ない、要求仕様を全てTFPを用いて記述する。</p> |

| | | |
|--|---|---|
| f コー ディ ング シ ート 作 成 | 1 | <p>TFP記述が全て終了した時点でシステムはTFP図と構造がまったく同じコーディングシートを作成し、表示する(図8-f)。</p> <p>設計者は先の設計結果を基にして、コーディングシートの各処理ボックスの中をソースコード(現時点ではC言語がメイン)でうめていく。使用する部品群の引数名のチェック、引数値の設定、リターン値の扱い等に注意しながらソースステートメントを作成する必要がある。該当する部品が存在しなかった部分の処理ボックスの中は全て設計者がコーディングする。</p> |
| | 2 | <p>コーディングシート作成が終了すると、次に関連データシートを作成する。詳細については省略する。</p> |
| g コー ド 生 成 | | <p>以上の作業が全て終了するとシステムは設計対象であるモジュールあるいはタスクのソースコードを生成する。この際使用する部品のソースコードはシステムが部品データベースから部品番号によって取り出す。生成ののちコンパイルしてオブジェクトモジュールを作る。タスクの場合は更にリンクしてロードモジュールを生成する。</p> |

8. 考 察

8. 1 比 較

7章で述べた設計支援システムは3対象1操作モデルに基づく部品仕様フレーム群と機能展開知識及びTFPを用いてソフトウェア設計・製造作業を効果的に支援するものである。

事務処理分野においてはKIPS^[7]やARIES^[8]あるいはPAPS^[9]といった部品合成システムが成功事例として知られている。これらのシステムと比較しながら本システムの特徴をまとめると表1の通り。

この表から、本システムが他のシステムと異なる大きな特徴は、分野を特に限定しないということ及び部品群を組み合わせる際に機能展開知識を用いることが上げられる。処理パターンの中に部品を埋め込んでいくという方法は確かに有効であるが、事務処理分野に依存した部分が多い。経験上、いくつかの処理パターンが存在すること、扱うデータの構造の種類が限られていること等この分野に固有な事情がかなり影響している。事実、事務処理以外の分野においては、処理パターンを基にした自動生成ツールの例は比較的少ない。

そこで我々は他の分野においても適用できる部品合成の方法について検討してきた。その結果、1つのやり方として機能展開知識をキーとしたTFPによる設計手法をまとめた。機能展開知識は機能抽象度のレベルとして上はタスクのレベルから下は文字コード変換サブルーチンのレベルまで幅広くカバーすることができる。個々の展開知識を次々に連鎖させて階層化していくことも可能である。ベテランの設計者が持っている機能分割・詳細化能力を表現する強力な手段となり得る。もちろん、本

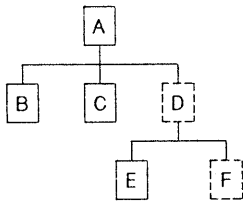


図4 機能展開知識の連鎖

部品仕様フレーム

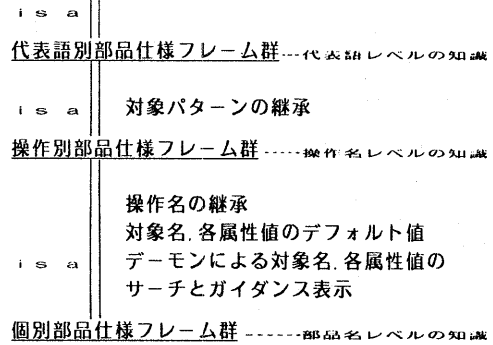


図5 部品仕様フレーム群

FRAME 部品コピーする

```

SLOT IS-A
FACET value 部品移動する
SLOT IS-A-CONV
FACET value 部品 bbn00006
          部品 bbn0134
          部品 bbn0028

SLOT 対象1
FACET value demon srh-a
FACET value ファイル
SLOT 対象2
FACET value NONE
SLOT 対象3
FACET value demon srh-a
FACET value ファイル
SLOT 表現形式1
FACET value demon srh-c
FACET value ランダムファイル

SLOT 操作
FACET value コピーする
SLOT 作用開始1
FACET value demon srh-d
FACET value ファイル先頭
  
```

図6 操作別部品仕様フレーム群

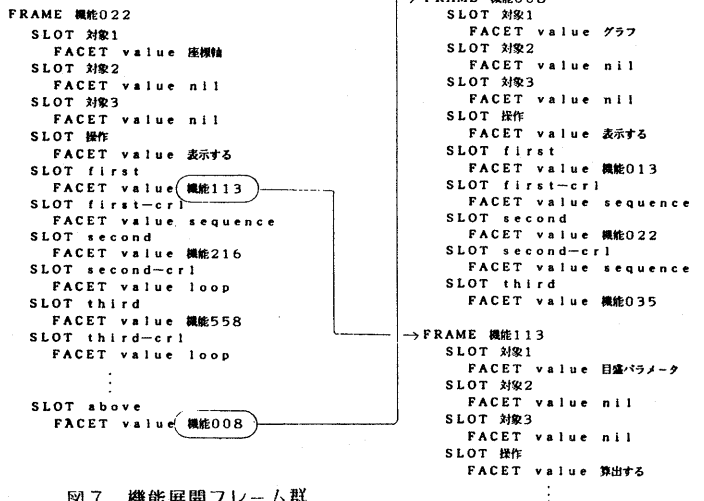
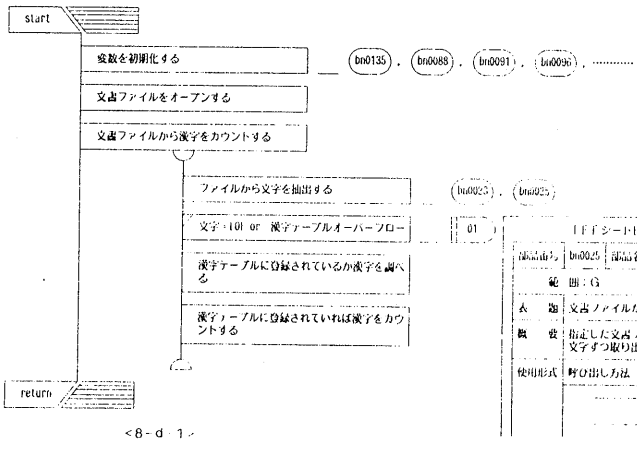
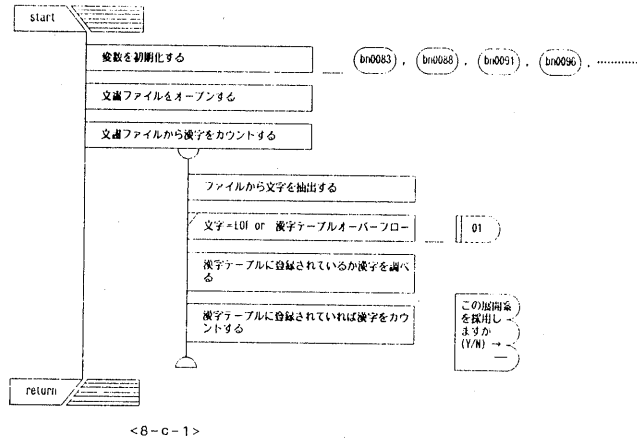
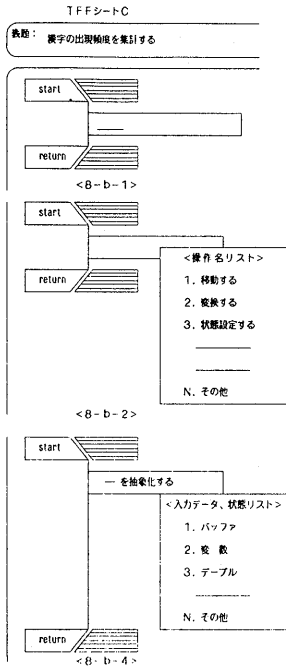


図7 機能展開フレーム群

| システム | SEA/T | PAPS | ARIES | 本設計支援システム |
|-----------------|----------------------|----------------------|----------------------|---------------------------------|
| 項目 | | | | |
| 対象分野 | 事務処理 (バッチ処理) | 事務処理 (バッチ処理) | 事務処理 (バッチ処理) | 分野を限定しない |
| 要求仕様の与え方 | 階層メニュー | 階層メニュー | 制限された自然言語又は階層メニュー | TFF+階層メニュー |
| 採用しているモデル | 特になし | 部品表現モデル | データ導出モデル | 3対象1操作モデル |
| 用いる部品等 | 詳細部品 (処理パターン) + 部分部品 | 詳細部品 (処理パターン) + 部分部品 | 詳細部品 (処理パターン) + 部分部品 | 標準部品群 |
| 部品の検査機能 | 人間系で行なう | ルールの形で持つ | フレームに見出し情報を持つ | 仕様表現モデル同志をマッチングする |
| 部品の組み合わせ方 | 処理パターンの中に部品を埋め込んでいく | 処理パターンの中に部品を埋め込んでいく | 処理パターンの中に部品を埋め込んでいく | 標準部品群を機能展開知識を利用してTFFエディタで組み合わせる |
| 部品のカスタマイズ機能 | 適合化ルールの形で持つ | 適合化ルールの形で持つ | 適合化フレームの形で持つ | なし |
| 生成されるプログラムの記述言語 | COBOLが主 | COBOL | COBOL | C言語が主 |

表1 他システムとの比較



TFFシートB

| | | | |
|------|-----------------------|-----|---------|
| 部品名 | bn0025 | 部品名 | getword |
| 範囲 | G | | |
| 主題 | 文書ファイルからの文字抽出 | | |
| 機能 | 指定した文書ファイルから1文字ずつ取り出す | | |
| 使用形式 | 呼び出し方法 | | |

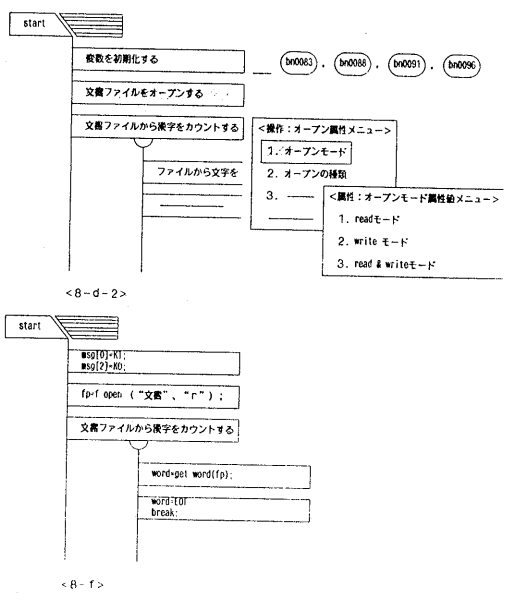


図8 設計支援システムとの対話画面