

自動ドキュメンテーション支援システム“ADCAS” —金融アプリケーションパッケージへの適用例—

Auto Documentation Aid System “ADCAS”

ソフトウェア保守作業の効率向上が、EDP部門での生産性向上の最重要課題となっている。ソフトウェアの保守では信頼性が高く、見やすいドキュメントが要求されるが、従来の手書きドキュメントでは、保守を重ねるたびにドキュメントの信頼性が低下し、保守コストを増加させる要因となっている。

この問題を解決するために、プログラムから保守ドキュメントを自動作成する自動ドキュメンテーション支援システムADCASを開発した。ADCASの出力ドキュメントは、日本語での表記とし、更にPAD図などの構造化プログラミング技法を取り入れ、プログラムの可視性を大幅に向上することができる。本論文では、金融アプリケーションパッケージへの適用例を中心に、ADCASの適用効果について述べる。

平川知親* Tomochika Hirakawa
中村康一* Kōichi Nakamura
谷部幸男* Yukio Yabe
戸金旗一** Kiichi Togane

1 緒言

ソフトウェアの開発規模が増大するにつれて、EDP (Electronic Data Processing) 部門のバックログが急増している。この背景として、既存のソフトウェアを保守する作業に多大な工数を要していることが挙げられる。この保守効率を向上させることが、ソフトウェア開発の生産性向上につながると言える。保守効率を向上させるためには、正確で見やすいドキュメントが要求されるが、プログラムの保守を重ねるたびに、保守用ドキュメントとプログラムが一致しなくなるという問題がある。この問題を解決するためには、プログラムから保守用ドキュメントを作成するか、設計(仕様記述)ドキュメントからプログラムを生成するか、どちらかの方法が考えられる。自動ドキュメンテーション支援システム“ADCAS” (Auto Documentation Aid System) は、前者のプログラムから保守用ドキュメントを自動作成するシステムである。

日立製作所では、金融機関でのソフトウェア開発の生産性向上を目的に、金融オンラインアプリケーションパッケージとして“EXPERT” (Experienced Partner for Efficient Real-Time Banking System) 及び“ESCORT” (Effective Software for Customer-service based on Online Realtime Banking System) を開発した。このアプリケーションパッケージの開発に当たり、適用ユーザーでの保守が容易なドキュメントの提供を目的としてADCASを採用した。

ADCASは導入のしやすい日本語変換方式の採用、構造化プログラミング技法に対応したドキュメントの出力が特徴である。

2 金融機関におけるソフトウェア開発の動向

金融機関では、先進都市銀行を中心とした第3次オンラインシステム開発のインパクトから地方銀行、相互銀行を中心に、勘定処理システム再構築の気運が高まっている。

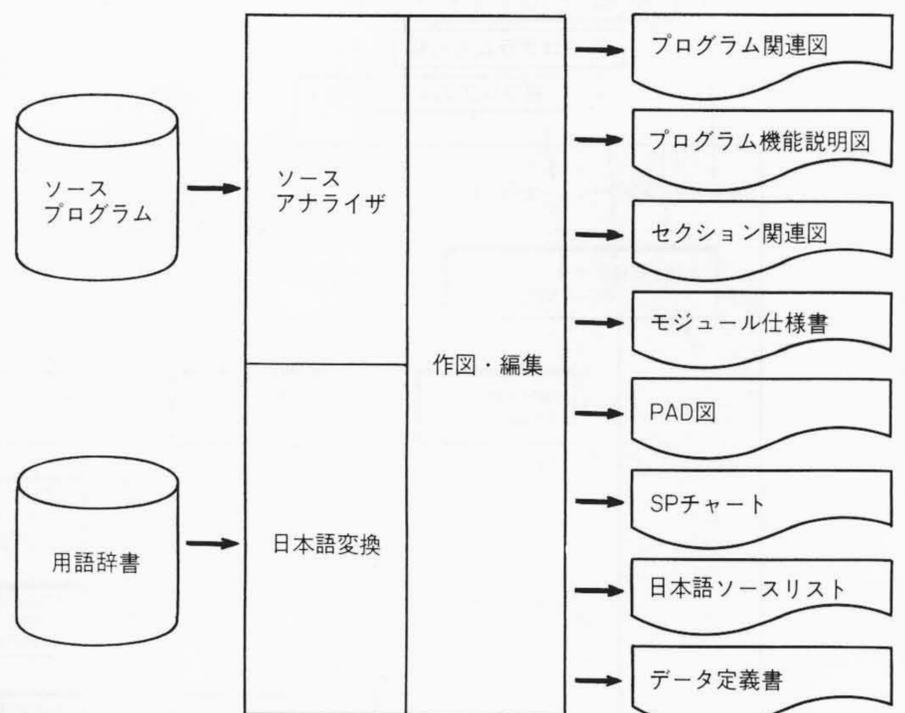
しかし、地方銀行、相互銀行などの地域金融機関では、先進都市銀行のように勘定処理システムを独自で全面再構築することは、開発要員不足の問題だけでなく、過大な投資によ

る経営圧迫の要因ともなりかねない。更に、開発要員が確保できても短期間に新システムを構築することは困難である。

日立製作所では、このような状況下で、これまでの金融オンラインシステムの経験により蓄積したノウハウを基に、金融オンラインアプリケーションパッケージ“EXPERT”及び“ESCORT”を開発した。EXPERTは地方銀行、相互銀行向けの大規模パッケージであり、ESCORTは信用金庫、信用組合向けの中規模パッケージである。

3 パッケージ適用上の問題

パッケージの適用は、パッケージの仕様を理解し、ユーザー側で容易に改造できることが前提条件である。EXPERT,



注：略語説明 PAD(Problem Analysis Diagram), SP(Structured Programming)

図1 ADCAS(Auto Documentation Aid System)の入出力構成
ソースプログラムを入力として、8種類の保守用ドキュメントを漢字プリンタに出力できる。

* 日立製作所大森ソフトウェア工場 ** 日立コンピュータコンサルタント株式会社

ESCORTは、「少ない要員」、「短い期間」でのシステム構築が目標であり、パッケージの仕様理解が容易で、継続して保守が可能なドキュメントを提供する必要がある。しかし、現実には150万ステップにも及ぶ大規模システム(EXPERTの例)について、プログラムと完全に一致した誤りのないドキュメントを作成するには膨大な工数を要し、適用ユーザーが改造した場合にも同様の保守作業(ドキュメントの修正)が発生することになる。

4 自動ドキュメンテーション支援システム“ADCAS”

4.1 ADCASにより作成できるドキュメント

ADCASはCOBOL又はPL/Iで記述したソースプログラム

を入力として、PAD図(Problem Analysis Diagram: 問題分析図)、日本語ソースリストなど8種類の保守ドキュメントを自動作成できる。図1にADCASの入出力構成を、図2に出力ドキュメントの体系を示す。

(1) プログラム関連図・セクション関連図

プログラムあるいはプログラムを構成するセクションの呼ぶ・呼ばれる関係を階層構造図の形式で出力する。プログラム関連図からプログラムの全体構造を把握することが容易となる。

ここでいうプログラムとは、ソースプログラムのコンパイル単位を指し、セクションはCOBOL言語では手続き部の節、PL/I言語では手続きブロックを指す。プログラム関連図の出

単位	種類	データフロー図 (HIPO図)	コントロールフロー図		データ一覧表	ソースリスト
			階層構造図	フロー図		
手 続 き	プログラムの集まり		プログラム関連図			日本語ソースリスト
	プログラム	プログラム機能説明図 入力 機能概要 出力	セクション関連図			
	セクション	モジュール仕様書 入力 処理 出力		SPチャート PAD図		
データ定義	データセグメント				データ定義書 	

注: 略語説明 HIPO(Hierarchy plus Input-Process-Output)

図2 出力ドキュメントの体系 任意のドキュメントを選択して出力できる。

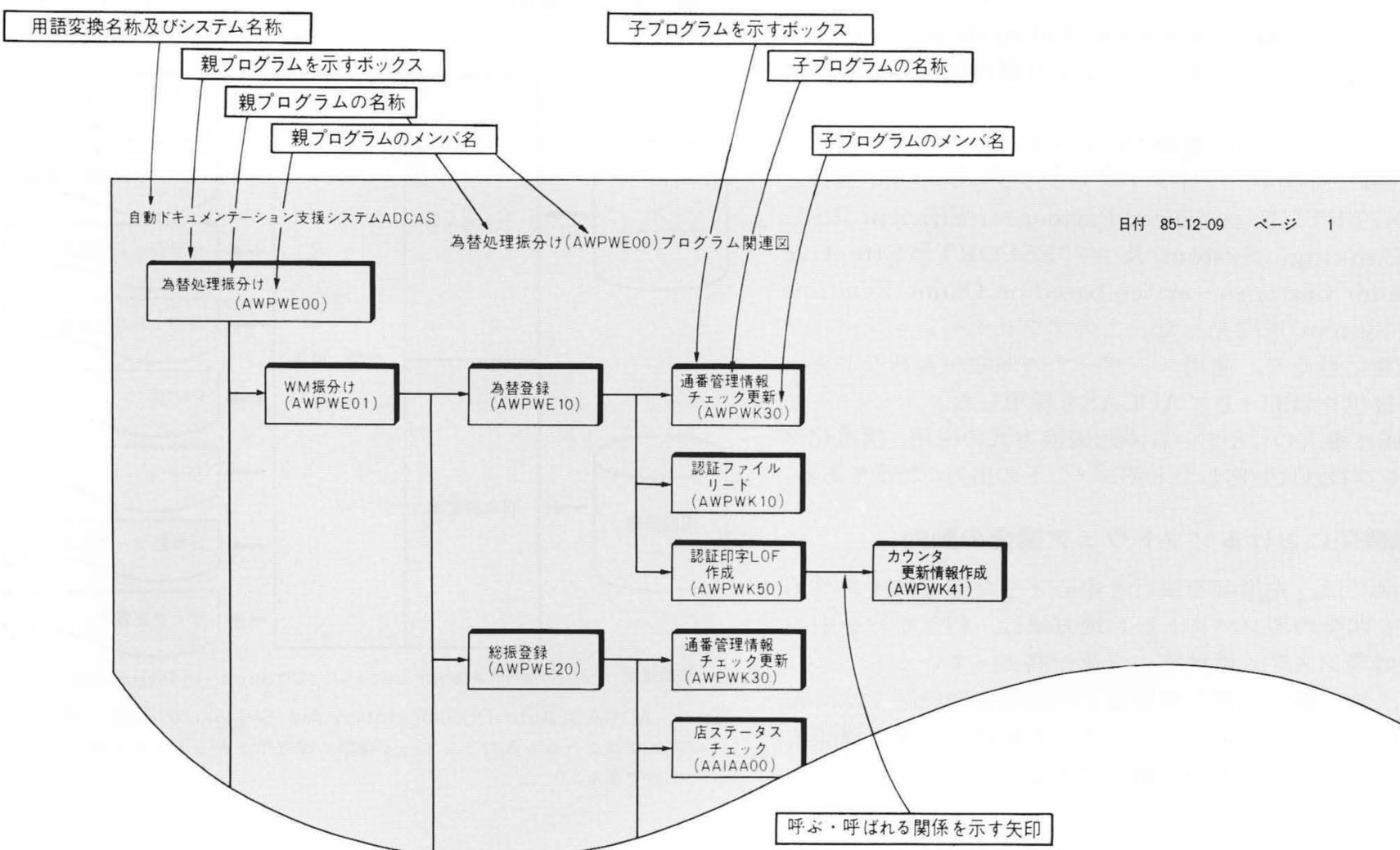


図3 プログラム関連図の出力形式 プログラムの全体構造が一目で分かる。

力例を図3に示す。

(2) プログラム機能説明図・モジュール仕様書

プログラムの機能概要あるいはセクション単位の処理概要をデータフロー図(HIPO: Hierarchy plus Input-Process-Output図)の形式で出力する。

(3) PAD図・SPチャート

プログラムを構成するセクション単位に処理構造、手順をPAD図もしくはSP(Structured Programming)チャートの形式で出力する。PAD図の出力例を図4に示す。

(4) 日本語ソースリスト

ソースプログラムの行単位に、ソースステートメントとこれに対応した日本語ステートメントを出力する。日本語ソー

スリストの出力例を図5に示す。

(5) データ定義書

データセグメント単位に、個々のデータの名称、属性、意味、用途などを一覧表の形式で出力する。

4.2 ADCAS適用時の規則

ADCASによるドキュメント作成を行なう場合、入力となるソースプログラムは、「日本語変換対象名称のコーディング規則」及び「機能コメントのコーディング規則」にのっとってコーディングする必要がある。

(1) 日本語変換対象名称のコーディング

日本語変換を行なうデータ名称、手続き名称については、以下に示す規則に従ってコーディングしなければならない。

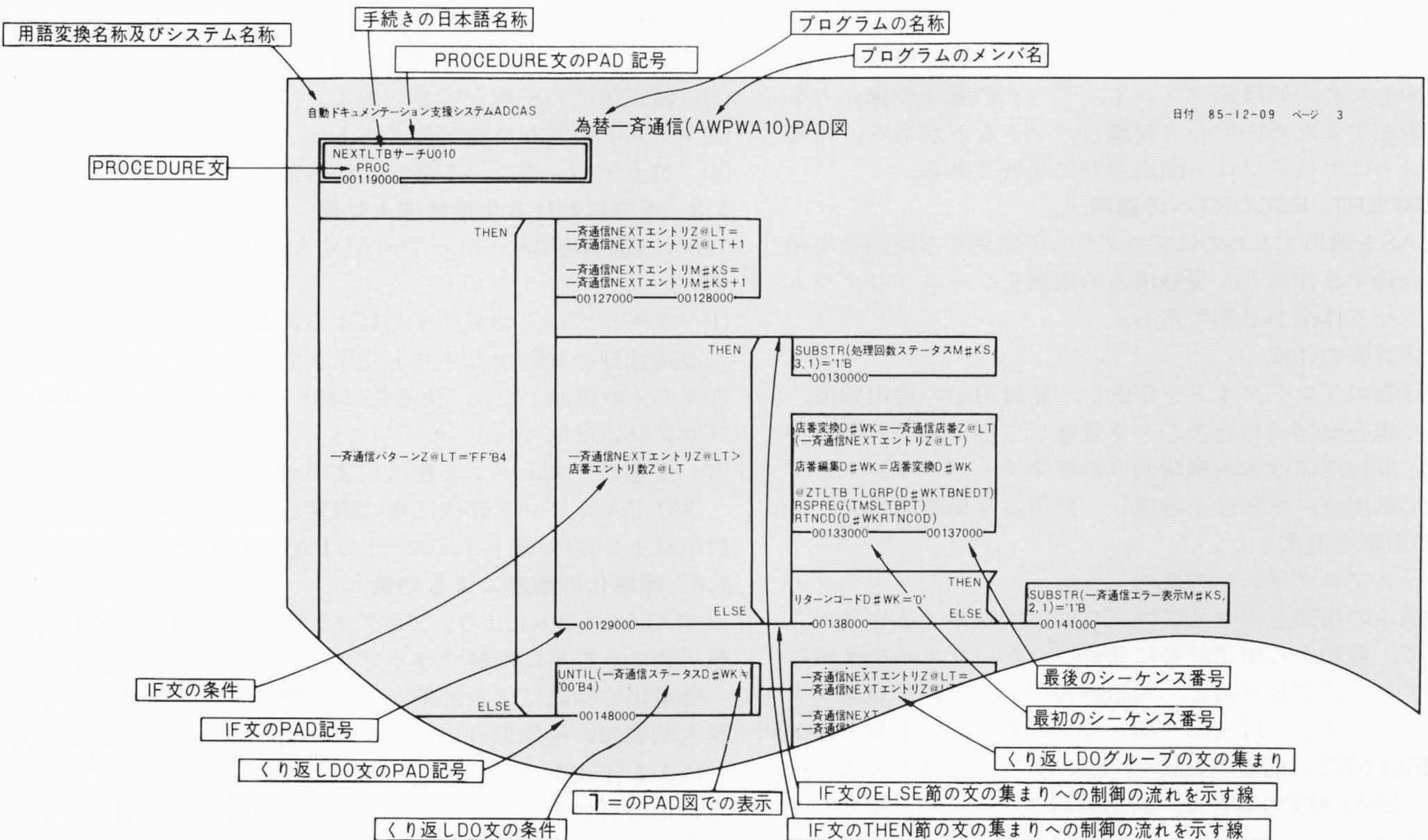


図4 PAD図の出力形式(PL/Iの例) プログラムの論理、処理の流れが一目で分かる。

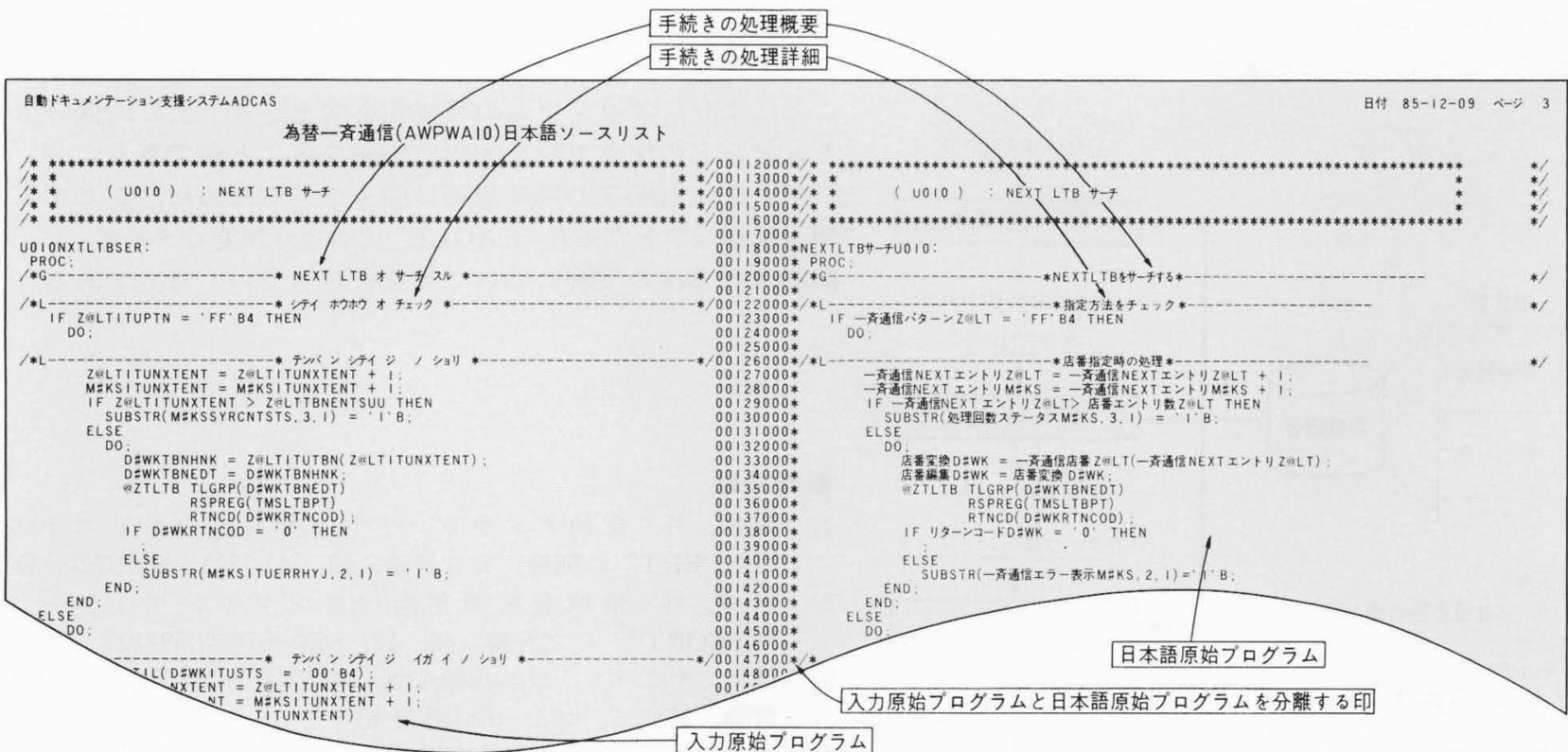


図5 日本語ソースリストの出力形式(PL/Iの例) 入力原始文と日本語表記原始文を対応させて出力する。

(a) 4-3-3の規則

先頭4文字を区分とし、以下3文字を1単語として最大5単語で日本語変換対象名称を構成する。先頭4文字はデータ区分、手続き区分として使用する。

(b) 1-3-3の規則

先頭1文字を区分とし、以下3文字を1単語として最大5単語で日本語変換対象名称を構成する。

ADCASは導入のしやすさを考慮して、上記(a)もしくは(b)のどちらかの方式を任意に選択可能であるが、更に、ユーザーOWNコーディングにより独自の日本語変換を行なうこともできる。日本語変換の例を図6に示す。

(2) 機能コメントのコーディング

機能コメントは、プログラムあるいはセクションの機能、データの意味や使い方などを保守用ドキュメントに表示させたい場合に使用する。機能コメントには、処理の概要を日本語表記するための処理部コメント、データの意味や使い方を日本語表記するためのデータ定義コメントなどがある。処理部コメントはデータフロー図出力時に必要である。

4.3 EXPERT, ESCORTへの適用

ADCASを適用するためにプログラムで使用する用語を用語辞書に登録する作業と、登録済みの用語をソースプログラムに反映させる作業が必要である。

(1) 用語辞書の作成

用語辞書のプロトタイプを作成し、登録用語の使用頻度、シンボル組合せ時のけた数などを考慮して標準用語辞書を作成した。登録用語は金融機関向けの標準テーブルウェアとするために汎用性、拡張性を考慮し、熟語より単語を優先して約3,500用語を選択した。

(2) ソースプログラムへの反映

登録済みの用語をソースプログラムに反映させる作業の方法として、最初から用語辞書に登録されたシンボルを使用してコーディングする方法と、用語辞書のシンボルを意識せずにコーディングし、最後に一括してソースに反映する方法がある。EXPERTは前者の方式を、ESCORTは後者の方式を採用した。ESCORTの場合は、新旧名称変換ツールを開発してソース変換を行なった。

5 ADCASの適用効果

5.1 ドキュメントの正確性

ソースプログラムから日本語の保守ドキュメントを自動作

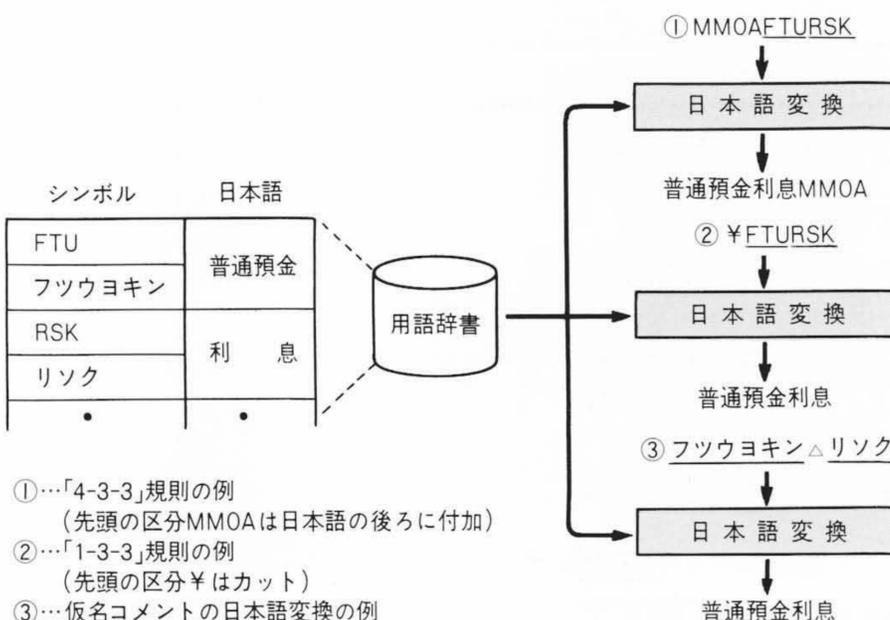


図6 日本語変換の例 ソースプログラムのシンボル及び仮名コメントを、用語辞書を使用して日本語に変換する。

成できるため、ソースプログラムと保守ドキュメントは常に一致している。ただし、EXPERT, ESCORTではデータフロー図を使用していない。これはデータフロー図に表記される機能概要、処理概要のコーディングが属人的になり、結果として信頼性が低くなると判断したためである。しかし、手書きドキュメントを削減し、正確なドキュメントを作成するために、データフロー図についても適用を検討中である。

5.2 ドキュメントの見やすさ

構造化プログラミング技法に対応した各種ドキュメントが日本語で出力されることにより、プログラムの可視性が大幅に向上する。EXPERT, ESCORTでは、主としてプログラム関連図、PAD図、日本語ソースリスト、データ定義書の使用を前提とした適用手順の標準化を行なっており、以下に述べる効果がある。

- (1) プログラムの論理把握が容易である。
- (2) 修正箇所の把握が容易である。
- (3) テスト範囲が特定可能である。
- (4) 机上デバッグでの不良摘出が容易である。

5.3 保守における生産性向上効果

EXPERT適用ユーザーでのADCAS適用効果を以下に述べる。

- (1) ドキュメントの見やすさによる効果
改造仕様の検討からテスト完了までの全体工数では、20~30%の工数削減となり、生産性は25~45%向上する(新規開発プログラムは除く)。

- (2) 手書きドキュメント削減による効果
保守ドキュメント作成工数に限定した場合、60~70%の工数削減となる(対象ドキュメントはPAD図及びデータ定義書)。

5.4 標準化の徹底による効果

ADCASの導入により、プログラミングの標準化が徹底され、第三者でも容易に理解できるプログラムが作成できる。

標準化の徹底による効果として、プログラミング段階での属人的要因の排除以外に、自動化作業範囲の拡大がある。人手による作業を自動化するには標準化の徹底が必要であり、自動化の具体例として、ソースプログラムから運用マニュアルを作成するツールなどを開発している。

6 結 言

ADCASは構造化プログラミング技法に対応したPAD図などのドキュメントをソースプログラムから自動的に作成することにより、プログラムの可視性を向上させ、更に、保守ドキュメントの作成工数を大幅に削減することができる。

プログラム保守の効率を更に向上させるために、システム開発支援ソフトウェア“EAGLE”とのより緊密なタイアップ、機能表記方法の改善について今後検討していく考えである。

参考文献

- 1) 重松, 外: 金融オンラインアプリケーションシステム“EXPERT”の開発, 日立評論, 65, 4, 301~306(昭58-4)
- 2) 岩瀬, 外: 地域金融機関向けオンラインパッケージ“ESCORT”, 日立評論, 66, 12, 865~870(昭59-12)
- 3) 前沢, 外: プログラム自動生成システム“SDL/PAD”, 日立評論, 66, 6, 463~468(昭59-6)
- 4) 葉木, 外: システム開発支援ソフトウェア“EAGLE”, 日立評論, 66, 3, 189~194(昭59-3)