3J－7　ソフトウェア開発支援システム　W-1の開発

伊藤　昌夫　　寺井　文康　　加藤　昌也

MHIエアロスペースシステムズ(株)

1. はじめに

ソフトウェアを支援する多くのCASEツールがある[1]。しかし、それらを実際に使用しようとする場合、幾つかの問題に出会う。第一に、それらCASEは限られた特定の設計手法(通常はSA/SD)しか支援せず、出力の図式・フォームもまた限定されると云うことである。

例えば、当社では開発手法としては機能分割法を用い、図式としてはHIPOを使用しているため、それらを支援していない一般のCASEツールは使用出来ない。そうした場合、どのような対応策があるのだろうか。例えば、SA/SDの有効性は示されているから、開発手法そのものを、図式/ドキュメントを含め、ツールが支援するものに変更すれば良いと云う議論もある。しかし、本来人間(開発者)がツールに合わせるのではなく、ツールを人間に合わせるか、選択可能なだけのツールを用意すべきであろう。

次に、プロジェクト管理と云う視点から見る。CASEは管理に必要な設計者の活動を知り得ると云った点に於て、プロジェクトの実行管理を行う上での潜在能力を有しているが、その能力を生かしているツールは少ない。

本論では、今回設計製作し、社内プロジェクトに適用している開発支援ツールW-1に於ける解決方法を示す。

この支援ツールは、プログラム設計・ドキュメント作成・プロジェクト管理と云ったソフトウェア開発プロセスの多くを支援する。主な特徴を以下に挙げる。

(1)新たに設計した、コード・図式記述言語inferを用いることで、任意の図式及びプログラムコードの自動生成が出来、両者の整合性を確保することが出来る。

(2)従来のCASEでは見落とされがちだった管理的側面を支援する。

2. 作業内容の調査

設計フェーズ毎の所要工数に関するデータは多いが、実際の作業内容と云う視点で区分されたデータは少ない。本論で云う「作業内容」とは、設計文書の作成や関連文書を読む、或は打ち合わせと云った全ての開発フェーズで行われる作業である。つまり、フェーズに対して横断的に区分された作業のことである。

社内で調査したこれら各作業内容に於ける時間配分を図1.に示す。(データは、異なったフェーズにあり、ツール未使用の幾つかのプロジェクトから取得した。)

プログラミングを行う時間は、全体の1/4であるのに対し、机上検討・文書作成等に関わる時間は全体の半分を占める。

また、管理関係の時間はわずかに5%であるが、これは設計作業そのものに時間を取られて、プロジェクト管理のための各種データが取得しにくいためである。

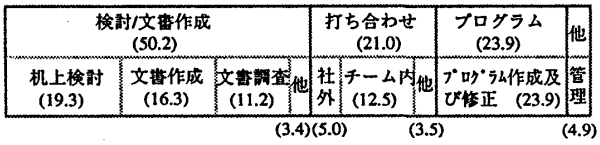

図1.　各作業内容に於ける時間配分

3. 目標

一般に支援ツールに要求される要件の他に、2章に示した現状を踏まえて、今回の開発支援ツールの開発では、特に以下のような点に着目することとした。

(1) 種々の図式を含んだドキュメントの自動生成

(2) 必要となる管理用データの自動収集と加工

なお、1章で挙げた問題点の中で、「設計手法が限定されてしまう」と云うことについては、リポジトリへのインターフェースを考慮することで、今後の拡張の余地を残すにとどめる。

4. W-1システム

4.1 ツール構成

全体構成を図2に示す。これらツール群は、ワークステーションによる分散環境の中で、互いにリポジトリを介して情報のやり取りを行う。

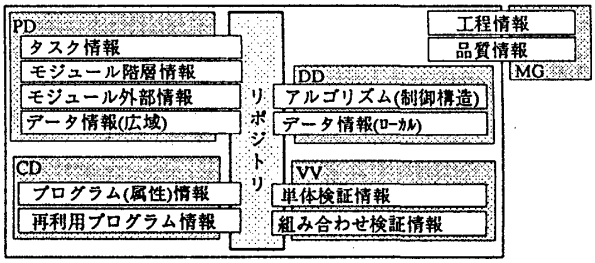

図2　W-1システム全体構成

次に各ツールの概要を示す。

(1) PD(基本設計支援)

このフェーズでは、タスク設計、モジュール外部設計、及び広域データ設計が行われる。関連するドキュメントが出力される他、入力された情報はリポジトリに保持される。

(2) DD(詳細設計支援)

このフェーズでは、次項で示すコード・図式記述言語inferを用いてモジュール内の設計が行われ、内部処理を示すIPO等のドキュメントが出力される。また、プログラムコードに関しては、PDで設計されたデータと合わせて、データ定義／宣言ファイル及び、アルゴリズム部分を含むコードのスケルトン部分が生成される。

(3) CD(コーディング支援)

このフェーズではDDで生成されたスケルトンコードをベースに直接設計者が最終的なコードを作成する。この部分は一般的な下流CASEツールをベースと

Software Development Support System: W-1
Masao Ito , Fumiyasu Terai , Masaya Kato
MHI Aerospace Systems Co.,LTD.

したものであり、W-1専用のインターフェイスにより、必要な情報はリポジトリに保存される。保存されるデータとしては、プログラム及び属性としての作成日、バージョン、コード行数等の情報である。また、再利用可能なプログラム部品に関する情報を作成したり、参照することが可能である。

(4) VV(検証支援)

単体検証についてはC0、C1レベルの検証すべきパスを自動的に判別する。対象モジュール、入力値、期待値、合否判定条件等を入力することで、自動的に判定を含めて検証を行い、成績がドキュメントとして出力される。組み合わせ検証レベルでは、実行計算機が開発計算機と異なることが多く、検証成績等のデータの解析が主たる支援である。

(5) MG(プロジェクト管理支援)

メンバーが日々入力する作業報告(作業内容、時間等)が、MGに於ける入力となる。一方、PDからVVまでのツール群より得られた情報を用いて、任意のタイミングで各成果物の状態を知ることが出来る。

従って、本ツール中で、各人の作業内容に関する入力と、その成果物を関連づける事により、動的な管理者用データを生成することができる。

つまり、管理を行う上で必要となる計量するべき事がら、つまり、「ドノ人がイツ、ナニをドノクライ実施したか」と云うことを、タスク単位／モジュール単位/検証単位等の単位毎に出力する。

4.2 コード・図式記述言語 infer

inferはコードとドキュメント(図式)を同時に生成するための言語である。図3.にinfer コードとそのinferコードより生成されたドキュメント(IPO)及びソースコードの関係を示す。

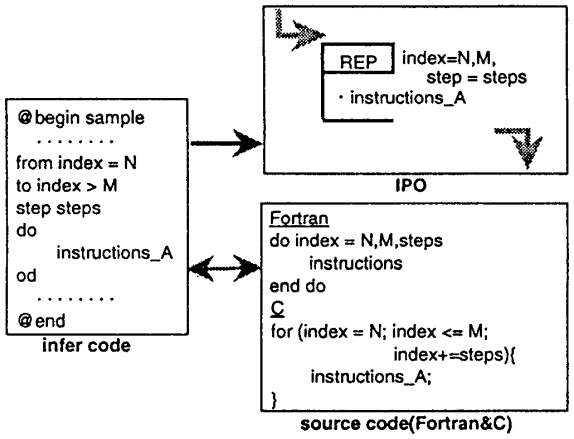

図3. inferコードと生成された図式とソースコードの関係

図式を生成する場合、一般な手段としては、テンプレートを用い、実際画面上に描画していく方法と、何らかの記述言語を用いて生成する方法があり、それぞれ一長一短がある。今回は実行計算機に依存しにくく、かつ効率が良いと云った点から後者の記述言語を用いる方法とした。

inferでは、完全なソースコードの自動生成を目指したものではない。infer言語で書かれた部分からのみ、コードが生成される。最終的な詳細なソースコードを生成するためには、実際のコードをinfer上に埋め込む必要がある。しかし仕様を限定することで、多言語に対応でき(現状はCとFortran)、また、設計者は容易にinfer言語を修得することが出来る。

なお、データに関しては、PDより入力された情報を利用し、整合性／論理的な誤り等がチェックされる。

データ、アルゴリズムの主要な部分については、言語に依存せず単一の表現となるため、プログラム言語とは独立に、inferから図式への変換ルーチンを用意することで、任意の図式(IPO、PAD等)で出力することが可能である。また、ソースコードからinferへの部分的なリバースも可能である。

5. 効果

社内で運用開始後、まだ半年程度であることもあり、効果を測定することは出来ない。しかし、単純なドキュメント作成に関しては、inferを使用することで大きな効果が上がる。入力がイメージではなくテキストデータであるため、既存データの修正に於ては、修正率にほぼ比例した作業時間となる。(手書き、ワープロ等のデータの場合、修正率以上の作業時間が必要となる。)

但し、通常文書を作成しながら設計に必要となる検討を行っており、全体としてどの程度効率が向上しているかを知るためには、更に調査を進める必要がある。

6. 今後の課題

先ず第一に、本ツールの適用による作業内容の変化がどのようになるかを詳細に調べて行きたい。

更に、本ツールの拡張を考える上で次の事がらに着目したい。

現在のCASEツールは、単独で全てをカバーするのではなく、ESPRIT[2]等で検討されていうように、ツール間のインターフェイスやリポジトリの仕様を定義することで、任意のツールを結合出来る環境を整える方向に進んでいる。

W-1のプロトタイプ版では、単独での利用となるが、SA/SD及びOOA/OODに対応した版も開発済である。そう云った意味では、1章に於て挙げた問題のうち残っている「特定の設計手法に依存している」ということは部分的には解決している。しかし、少なくともプロジェクト単位で、異なる設計手法間の情報のやり取りが可能であることが望ましい。今後、上記に示した動向を考慮しながら、設計手法の選択、図式の選択が可能となる様、本開発支援ツールの拡張を検討、実施して行きたい。

また、2章では設計作業に於て、他者との打ち合わせが全体の２０％に及ぶことを示したが、これら時間の有効活用のための方策についても考えて行きたい。

【参考文献】

(1) 原田実監修、「CASEの全て」、オーム社、1991
(2) I.Thomas,"PCTE Interfaces: Supporting Tools in Software-Engineering Environments",Software vol.6 No.6,Nov 1989