

ソフトウェア エンジニアリング アーキテクチャ SEA/I

松本 正雄

中田 修二

日本電気 (株) ソフトウェア生産技術研究所

1. はじめに

ソフトウェア エンジニアリング アーキテクチャ SEA/I (以下では SEA/I と記す) は、オフィス情報システムなどの各種アプリケーション・ソフトウェアを良質かつ効率的に生産、サポートするための最新鋭自動化システムである。このために SEA/I は開発対象システムのプロトタイプ作成から設計、ソフトウェア生成、実証、立ち上げ、フォローアップまでの全工程を一貫して支援する連動ツール体系を構成している。

本資料では SEA/I のツール体系、SEA/I によるソフトウェア開発の持長と実際の開発手順を中心に SEA/I を紹介する。

2. SEA/I のツール体系

SEA/I は、システム・エンジニア (SE) のアプリケーション・ソフトウェア開発経験の蓄積である経験情報ベースと、この経験情報ベースの上に構築されたツール系とから構成されている (図1)。さらに各ツール系にはいくつかのユニットツールが用意されている。

ソフトCAPツール系：従来システム稼働時まで待たなければ得られなかったコンピュータ出力、マン・マシンインタフェースといった人間/機械系境界条件を開発作業の初期段階において、対象を視覚化し、素早く製作してしまう。

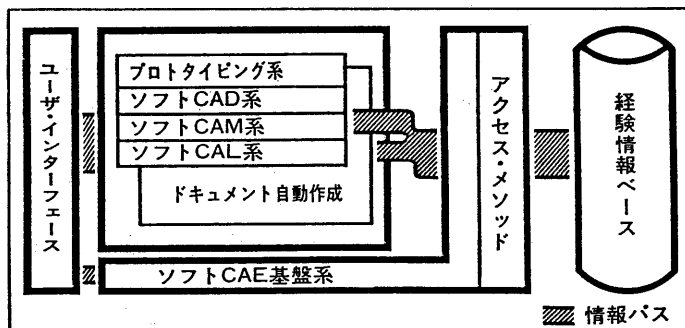
ソフトCADツール系：設計を詳細にっめていくのを大幅に助ける。オフィス情報システムで重要なフォームの設計、システム構造を段階的に階層化して設計し、更に個々のプログラムを整合構造で論理設計する事ができる。

ソフトCAMツール系：ソフトCADツール系での設計結果を利用してプログラム部品を生成または作成できる。さらにこれらプログラム部品を元にプログラムを合成できる。

ソフトCALツール系：実世界条件に照らした品質保証のためのテストベッドは SEA/I が準備する。さらにアプリケーションシステムの立ち上げも SEA/I で処理される。

アプリケーションの開発ライフサイクルを工程順にカバーする上記のツール系の他に、①自動化が進んでもなお

図1 ソフトウェア エンジニアリング アーキテクチャ SEA/I



依然として必要なドキュメントは S E A / I が作り出す、②保持したい経験情報は S E A / I が経験情報ベースにデータベース化するのて、再活用が容易になる、などのツール群が存在している。

3. S E A / I の特長

2章で概説した機能を S E A / I に実装せしめた考え方として、下記の各点が特長として挙げられる。

(1) 設計対象の視覚化

入出力画面や帳表などのフォームはこの実現イメージを端末の画面上で直接設計するフォーム・イメージ設計によって設計される。

開発対象アプリケーションシステムのシステム階層や構成単位間の関係づけなどは種々の図表を活用して行なう。

また、プログラム論理の設計にはプログラムの論理仕様を木表現プログラム構造で端末画面に表わし、対話的に設計を行える。

上記いずれの場合においても設計対象の実現イメージを視覚化しているので、設計内容の理解、確認、改善などが的確かつ容易に行える。

(2) プログラミングの自動化

イメージによる設計の結果は、設計書などのドキュメントとなるだけでなく、S E A / I の部品生成機能によりプログラム部品に自動的に変換される。さらに、それらプログラム部品は S E A / I 合成機能によって設計内容と一致したプログラムに自動合成される。

このように、S E A / I においては従来のプログラマの人手によるプログラム製造工程に変わって、設計結果に基づいてプログラミングの自動化が行われている。

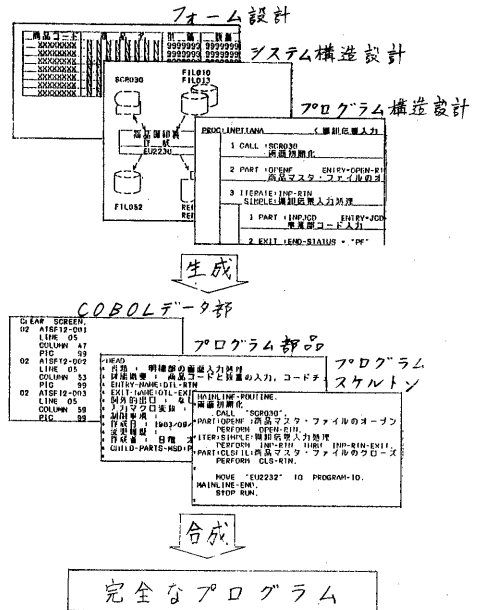


図2 S E A / I における視覚化と自動化

(3) ソフトウェア資産の利用

イメージ設計の結果や自動生成されたプログラム部品は、S E A / I の経験情報として体系的に分類管理される。

新たなアプリケーションシステムの開発には経験情報ベースに蓄積されているソフトウェア資産を有効に利用できる。また設計対象が実現イメージに視覚化されているので、直感的な判断が可能で、従来の課題であった「ソフトウェア資産の再利用できないかの判断の難しさ」も解決されている。

さらに S E A / I 経験情報の流通により、ソフトウェア資産を容易に活用し合える。

(4) 総合生産システム

開発対象アプリケーションのライフサイクルを一貫して支援するツール体系を構成している。これにより設計データの整合性や、ソフト開発工程間で

のSEA/Iツール同志の円滑な連動などは特に配慮されている。

4. ソフトウェア開発手順

図3は、SEA/Iのツール群とITOSユーティリティの間で受け渡されるソフトウェア情報に注目して、SEA/Iでのソフトウェア開発手順を表わしている。以下では図3に添って各ツールの機能を中心にソフトウェア開発手順を述べる。

4.1 プロトタイプング支援

アプリケーション開発の難しさは、利用者とシステム作成者との間で実現されるであろうアプリケーションについての共通の理解が得にくいことである。

SCAPのプロトタイプングツール(PROTOS)を使えば、アプリケーション完成時に出される帳表、伝票入力や問い合わせのための画面などの概略フォームを計算機の支援によって、実イメージを画面上に見ながら容易に作成することができ(図3A)。また一度作成したものの手直しも容易に行える。したがって開発対象アプリケーションの要求の明確化をスムーズに、かつ确实に行うことができる。

また、システム構造設計を支援するツール(SYSDES)を使えば、利用者から見たアプリケーションのシステム階層構造を容易に作成あるいは変更ができ、アプリケーション全体の機能やデータの流束についての共通の理解を深めることができる。

4.2 設計作業支援

プロトタイプング作業の後をうけて作成するアプリケーションシステムを具体化する作業が、設計作業である。

ソフトCADツール系(SCAD)は、

(注) ITOS: 本報告のSEA/Iが稼動するオペレーティングシステム

- (1) システム構造設計ツール(SYSDES)
- (2) フォーム設計ツール(FOMDES)
- (3) プログラム構造設計ツール(PRGDES)

の3つのツールから構成されている。

システム構造設計ツール(SYSDES)はアプリケーションのシステム構造の設計をインタラクティブに支援する。また、本ツールはアプリケーションの設計インプリメントの過程で生成される各種ソフトウェア情報の体系的な管理の手助けをする(図3H)。

フォーム設計ツール(FOMDES)は、三種類のフォーム——帳表フォーム、画面フォーム、ファイルフォーム——のそれぞれに対応したユニットツールから構成されている。設計スタイルは、PRGDESと同じように実イメージを画面上に見ながらインタラクティブに行う。帳表フォームと画面フォームについては、PROTOSで作成した概略フォームを利用して詳細フォーム(図3C)を設計できる。作業の重複がなく効率的である。

プログラム構造設計ツール(PRGDES)はプログラム内部の機能構造を具体化し、プログラム部品を検索してこの機能構造の適当な位置に組み込む機能により、プログラム構造の設計を行なう(図3B)。

また設計情報をもとにCOBOLのプログラムスケルトンを作成する。これは、“プログラム合成指示”としてCOBOLプログラムランセサイザへの入力となる。

4.3 インプリメント作業支援

ほとんどの業務用アプリケーションシステムはCOBOLにより記述されている。SEA/IのソフトCAMツ

ール系 (JCAM) は COBOL によるアプリケーション開発を支援するものであり、

- (1) 書式制御パラメータ生成ツール (FMPEGEN),
- (2) COBOL データ部生成ツール (CODGEN),
- (3) COBOL プログラムシンセサイザ (COB SYN),
- (4) プログラム部品作成・管理ツール (METMNG),

の 4 つのツールから構成されている。

FMPEGEN と CODGEN は、フォーム設計支援ツール (FORMDES) が作り出した詳細フォーム情報から、自動的あるいは半自動的にそれぞれ書式制御のための #JFGEN パラメータ、#FORM のパラメータ (図 3 ㉔)、および COBOL データ部のファイル節・画面節・報告書節のソースコード (図 3 ㉕) を生成する。

COB SYN は、プログラム構造設計支援ツール (PRGDES) が作り出した COBOL のプログラムスケルトン (図 3 ㉖) を入力し、このプログラム合成指示に従ってプログラム部品を集め、1本の COBOL プログラムを合成する。マクロ記述を含む汎用性の高い部品を使用している場合には、合成後インタラクティブにマクロパラメータを与えてやり、マクロ展開を行う。マクロ展開後の COBOL ソーステキストは、シンタクスチェックを行える。こうして作り出された COBOL プログラムは構文エラーのない完全なもので、即コンパイル、リンク、実行テストが行える。

COB SYN には以上のプログラム自動合成機能のほか、シンタクスチェックをかけながらインタラクティブに効率よく COBOL ソーステキスト

をエントリする構文指向エディタ機能をもっている。

METMNG はプログラム部品の生成と管理を支援するツールである。METMNG の作り出したプログラム部品はプログラム構造設計ツールで参照し、COB SYN がプログラム合成に使用する。プログラム部品の作成においてはマクロ記述が可能であり、汎用性の高い部品を定義することもできる。

また、テストインストール作業支援

アプリケーション開発の最後の作業は、インプリメントの完了したプログラム/システムが正しく機能することを確認し、こうして品質保証の完了したアプリケーションを利用先の計算機システム上にインストールすることである。ソフト CAL ツール系 (JCAL) はこの作業の手助けをする。

業務用アプリケーションシステムの品質点検作業では、マスタファイルやトランザクションファイルのデータを用意するという、手間のかかる面倒な作業が常につきまとう。テスト支援ツール (JTEST) はこのようなレコード構造をもった大量、または複雑なテストデータの生成を支援する (図 3 ㉗)。

インストール作業では、完成したアプリケーションシステムのロードモジュール、メニューパラメータ、実行制御用のジョブ制御言語などを格納するためのファイル、およびアプリケーションが使用する各種データファイルの作成が非常に大きなウェイトをしめる。この作業では、おのおののファイル容量算定とその算定結果に基づくファイルアロケート用の JCL の作成が中心となる。運用環境自動設定ツール (ENVGEN) はこのような作業を支援する。

5. おわりに

ソフトウェア エンジニアリング
アーキテクチャ S E A / I の概用を
紹介した。S E A / I ではオフィス情
報システム開発上の設計対象（フォー
ム，システム構造，プログラム構造な
ど）を視覚化し，ソフトウェアを構成
するプログラムの部品化を進め，また
ソフトウェア製造工程の自動化を行っ
ている。またこれらの機能を駆使して
開発したアプリケーションの設計情報
の重要部分は経験情報ベースに蓄積さ
れ，さらに利用することができるので
ソフトウェアの生産・品質向上の面
での寄与が大きい。今後は S E A / I の
機能と利用技術の一段の拡充を図りた
い。

最後に，日頃より御指導いただく当
研究所の藤野所長はじめ開発・利用に
おいて御指導いただいている方々に深
謝致します。

参考文献

- (1) 松本他「ソフトウェア C A S E
ファシリティ：S E A / I」
情報処理学会第27回全国大会論文集
PP 513 ~ 514 (1983)
- (2) 松本他「S E A / I におけるイ
ンタラクティブなシステム構造設計支
援 (S D A / 1)」
情報処理学会第27回全国大会論文集
PP 515 ~ 516 (1983)
- (3) 松本他「ソフトウェア エンジ
ニアリング」アーキテクチャ S E A
/ I (2) ドキュメント自動生成ツ
ール」
情報処理学会第28回全国大会論文集
PP 489 ~ 490 (1984)