

# CASEツールの統合インタフェースモデルの一考察

1R-4

岩田誠司、松村一夫、大筆豊、高橋生宗

株式会社東芝システム・ソフトウェア技術研究所

## 1. はじめに

ソフトウェア開発において、生産性・信頼性・保守性の向上はソフトウェア技術者の慢性的不足に伴い最重要課題となっている。このような問題を解決するために様々な設計方法論、言語、テスト・管理方法がこれまで研究開発されてきた。そしてこれらの方法論、言語等を計算機を用いてソフトウェア開発のライフサイクル上の特定フェーズあるいは複数フェーズを支援する環境が構築されており、こうした支援環境は現在、CASE (Computer-Aided Software Engineering) 環境として注目されている。[1], [2]

本稿では特に、個々のCASEツールを統合しライフサイクルの複数フェーズを支援する環境を構築する場合の問題点を明かにし、統合化に関する議論のベースとなるインタフェースモデルを提案する。また提案したモデルに基づき統合化の指針を示すとともに、現在のCASEツールを考察し、一貫CASEツールの実現方針を述べる。

## 2. CASE環境(ツール)

現在、CASEツールという用語は広く使われており、人によって指し示している範囲にも違いがある。本節ではCASE環境(ツール)に対し定義を与え、以降の議論の枠組みを与える。

まず狭義にはCASEツールを、

“ソフトウェア工学における設計手法、管理手法等を計算機を用いて実現・支援するツール”

と解釈できる。つまり、グラフィックエディタ、マウス等を利用してこれらの方法論を計算機上で実現するツールを指している。狭義の解釈ではソフトウェア工学上の方法論を基本としていることがCASEツールの条件である。

一方、広義には、

“計算機を用いてソフトウェアの設計・管理等を支援するツール”

と解釈できる。つまり、構造化分析・設計ツールだけでなく、言語指向環境、構造化指向エディタ、ツールキットなども含まれる。すなわち広義には、コンピュータ支援ソフトウェア開発環境を指している。

本稿ではCASEツールを広義の意味で定義する。

## 3. CASEツールの統合と一貫CASE環境

現在、多くのCASEツールは単体CASEツール(ソフトウェアライフサイクル上の特定の1フェーズを支援)として提供されている。しかし、単体CASEツールには以下のような問題があげられている。

- ・CASEツール間で成果物の流用が難しい(開発工程を断片的にのみ支援)
- ・CASEツール間で開発工程(入力作業)の重複がある
- ・ツールによってユーザインタフェースが異なる
- ・個々の単体CASEツールが主機能(EX. 構造分析)の他に副機能(EX. 版管理等)を持っている為、他ツール(EX. 版管理システム)との併用が難しい

従って近年、これらの単体CASEツールを統合して隣接するフェーズ(要求定義と基本設計、詳細設計とコーディング等)を支援する統合CASE環境の開発が行なわれている。しかし、要求分析から保守までのソフトウェアライフサイクル全体に渡って支援する環境で完成されている例は少ない。本稿ではライフサイクル全体を支援し、統一されたインタフェースを提供する環境を一貫CASE環境として位置づける(図1参照)。

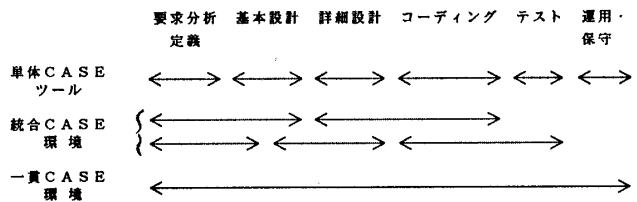


図1 CASEツール・環境の分類

このような単体CASEツールを組み合わせる統合CASE環境、あるいは一貫CASE環境を構築していくには、次のような統合方法が考えられる。

- ・成果物(ドキュメント)、運用による統合  
例) ドキュメント化された要求仕様書を見ながら設計を行なう等の人間系による統合
- ・ユーザインタフェースを統一することによる統合  
例) 単体CASEツール上に統一されたインタフェースを与える操作的な統合
- ・共通のツール環境を使用することによる統合  
例) 共通のウィンドウシステム、あるいはツールキットを利用してCASEツールを構築する
- ・共通DBによる統合  
例) 個々のCASEツールが扱うデータを共通のデータ構造で管理し、他ツールへデータ変換を行なう
- ・ファイル(コード)レベルで統合  
例) システム上のテキストコード、バイナリコード等のファイルレベルでデータの変換、補完を行なう
- ・ネットワーク等で異機種間のツールを統合  
例) MS-DOS上、UNIX上の同一種類CASEツールを結合

またソフトウェアライフサイクルに注目すると、

- ・同一フェーズを支援するCASEツールを統合  
表現形式、手法間の変換を行なう
  - ・異なるフェーズを支援するCASEツールを統合  
前段階のフェーズから必要な情報を次フェーズで利用する
- といった2種類の統合が考えられる。

## 4. CASEツール間インタフェースモデル

まずCASEツール間の統合の為のインタフェースとして以下の7階層を提案する(図2参照)。

- ・システム運用層・人間、組織によって個々のCASEツールを利用・管理するレイアで、広くは社会システムまで含める。
- ・ユーザインタフェース層・個々のCASEツールとユーザとのインタフェースとなるレイア。ツールの違いを意識しなくてもよいユーザインタフェースが要求される。

- ・ **ツール機能層** ・ 個々のCASEツールのプログラム本体があるレイア。ツールの機能を実現している。
- ・ **ツール環境層** ・ 個々のCASEツールに対し、ウィンドウシステム、ツールキット等を提供しているレイア。
- ・ **データ層** ・ データベース管理機能を提供するレイア。
- ・ **OS層** ・ 計算機のOSレベルで提供されているレイア。特にファイル管理機能、あるいはテキストコード、バイナリコードといったデータレベルを扱う。
- ・ **物理層** ・ CPU、メモリ、通信制御等の機能を提供するレイア。

システム運用層
ユーザインタフェース層
ツール機能層
ツール環境層
データ層
OS層
物理層

図2 CASEツールのインタフェースモデル

次に各レイアにおける統合化の指針を示す。

- ・ システム運用層では、組織、作業環境等の違いやツール間の結合（成果物の利用）を人間系および計算機系での運用を定め、CASEツール群を利用していく。この層では方法論・規定の設定や、ツールの運用管理サービスが特に重要となる。
- ・ ユーザインタフェース層では、個々のCASEツール毎に異なるユーザインタフェースに対し統一されたユーザインタフェースを提供し、操作方法や画面表示等のユーザと接する部分を統合化する。
- ・ ツール機能層では、サブツールの組み込み（エディタにバージョン管理ツールを組み込む）といった機能的な統合を行なう。各ツールの支援範囲・機能が明確に分離されなければならない。
- ・ ツール環境層では、共通のツールキット、あるいは基礎環境といったプラットフォームを与え、それらを利用して新規ツールを開発することにより統合を行なう。
- ・ データ層では、SEDB (Software Engineering Data Base) といったDBにより個々のCASEツールの概念レベルのデータをレポジトリとして管理し、新規あるいは既存のCASEツールの統合をデータによって行なう。
- ・ OS層では、テキストコード、バイナリコードといったデータの物理レベルでOSが提供している機能を利用してデータ変換を行ない統合する。あるいは同一のOS上にツールを構築する事によりOS自体を統一する。
- ・ 物理層では、CPU、メモリ、ネットワーク機能といった最もプリミティブなレベルで異機種間のCASEツールの統合を行なう。

5. モデルの考察

- ・ **単体CASEツール組合せの場合** (図3参照)  
単体ツールの多くは固有の操作法を持ち、自ツール内にデータ管理機能まで含んでおり閉じた環境となっている。つまりユーザインタフェース層からデータ層までが個々のCASEツールに分離されている(図中の1重点線)。従って、この場合にはシステム運用層、OS層、物理層で人間系による運用やデータ変換によって統合、管理が行なわれる(図中の2重点線)。しかし、システム運用層で明確な方法論や規定がないため組み合わせる事による利点が活かされていない点も問題である。
- ・ **統合CASE環境の場合** (図4参照)  
現状の統合CASEツールの多くにおいて、統合化はウィンドウシステム等の共通環境上にツールを構築し、共

通データを管理し、運用方法を確立することにより行なわれている。しかし共通に管理されるデータは特定の範囲に限られており、ユーザインタフェースも独自の形式のものも多く、各層間の分離も明確には行なわれていない(図中の1重点線)。また個々のCASEツールにおいて、統合化(機能拡張)の指針がない事も問題である。

・ **理想の一貫CASE環境の場合** (図5参照)  
理想的な一貫CASE環境を構築するには、ツール環境層、データ層だけでなくユーザインタフェース層をも含めた統合が要求される。また各層も明確に分離され独立していなければならない。統合CASE環境との大きな違いはソフトウェアライフサイクル全体を通じて標準のデータ構造を定義しDB管理機能を提供すること、およびユーザインタフェースを統一する事である。

6. おわりに

本稿ではそしてCASEツールを統合していくための議論のベースとして7階層のモデルを提案した。CASEツールの統合は各階層において行なわれる事が重要である。またCASEツール、環境について定義し現在の問題点を提案したモデルに基づいて述べた。

今後は提案したモデルに基づき各層の統合化指針を詳細検討し、我々が開発を行なっているソフトウェア生産工業化システムIMAP(Integrated Software Management and Production Support System)<sup>[3]</sup>の一貫CASE環境への拡張を進める。第一段階としてCASEツール間の基本データ要素を明確にしデータ層の統合化モデルを検討する。

参考文献

- [1] E.J.Chikofsky (eds), "Computer-Aided Software Engineering(CASE)", IEEE Computer Society Press Technology Series
- [2] A.S.Fisher, "CASE: Using Software Development Tools", John Wiley & Sons, 1988.
- [3] 大筆 他, "IMAPシステム(1)-(10)", 情報処理学会第31回全国大会予稿集, pp.489-508.

ツールA		ツールB	
システム運用層	システム運用層	システム運用層	システム運用層
ユーザインタフェース層	ユーザインタフェース層	ユーザインタフェース層	ユーザインタフェース層
ツール機能層	ツール機能層	ツール機能層	ツール機能層
ツール環境層	ツール環境層	ツール環境層	ツール環境層
データ層	データ層	データ層	データ層
OS層	OS層	OS層	OS層
物理層	物理層	物理層	物理層

図3 単体CASE環境の場合

ツールA		ツールB	
システム運用層			
ユーザインタフェース層	ユーザインタフェース層	ユーザインタフェース層	ユーザインタフェース層
ツール機能層	ツール機能層	ツール機能層	ツール機能層
ツール環境層			
データ層	データ層	データ層	データ層
OS層			
物理層			

図4 統合CASE環境の場合

ツールA		ツールB	
システム運用層			
ユーザインタフェース層			
ツール機能層	ツール機能層	ツール機能層	ツール機能層
ツール環境層			
データ層			
OS層			
物理層			

図5 理想の一貫CASE環境の場合