

# “RC<sup>2</sup>E”（リポジトリ中心のCASE環境）普及の価値と課題

## The value and tasks for diffusion of “RC<sup>2</sup>E (Repository Centric CASE Environment)”

三輪一郎<sup>†</sup>

Ichiro Miwa<sup>†</sup>

<sup>†</sup>株式会社プライド

<sup>†</sup>K.K. Puraido

### 要旨

CASE (Computer Aided Software Engineering) という概念が示されて 30 年が経つが、エンタープライズ系の IT 産業では、手作業によるプログラミング工程を経てソフトウェアを製造する手法が主流をなしている。

そこで、リポジトリ (Repository : Upper CASE ツールと Lower CASE ツールが共有する設計情報データベース) をシステム開発の中心に置き、そこに蓄積された設計情報を活用することで人手を介さないソフトウェア製造工程を実現する “RC<sup>2</sup>E (Repository Centric CASE Environment : リポジトリを中心とした CASE 環境)” の利活用を提唱するとともに、取り組むべき課題について考察する。

## 1. はじめに

本稿は、“試作型生成ツール” (試作型の下流 CASE ツール) との連携が加わることで、“RC<sup>2</sup>E” のレベルが上がり、導入容易性が高まる可能性を示すとともに、普及に向けた課題を考察するものである。

## 2. CASE 環境導入の課題

本稿では CASE ツールを活用するシステム開発環境を“CASE 環境”と呼ぶ。一定の価値が見込まれ、また、技術的には実用化に至っているにもかかわらず、実際には CASE 環境の導入が大きく加速するには至っておらず、普及に向けた何らかの課題が存在することが考えられる。

CASE 環境の導入に向けた課題について、まず、これまでに整理されてきた「一般的な課題」の面から考察を加える。

### 2.1. CASE 導入の一般的な課題

FFIEC(米国連邦金融機関検査協議会) が公開している“CASE”の解説ページ[1]では、「Common CASE risks and associated controls (CASE 環境導入の一般的なリスクと対策)」の例として、「Inadequate Standardization (不十分な標準化)」、「Unrealistic Expectations (過度な期待)」、「Quick Implementation (急激な導入)」、「Weak Repository Controls (リポジトリ運用の不徹底)」の 4 点を挙げている (邦訳は筆者)。本稿ではまず、これらを CASE 環境導入の一般的な課題として認識し、考察を加える。

「過度な期待」では、コスト削減効果を見込む際に、初期費用を勘案することを勧めている。また、「急激な導入」では、ツール操作に習熟する期間を見越して、短納期案件にいきなり適用しないことを勧めている。但しこれらは、一般的なツール導入にも当てはまると認識し、本稿では、「不十分な標準化」と「リポジトリ運用の徹底」を考察の対象とする。

### 2.2. 不十分な標準化

「不十分な標準化」は、異なるベンダが提供する上流 CASE ツールと下流 CASE ツールの連携を実現するために考慮すべき、大きな課題である。この課題に対する解決策として、OSLC の活動が挙げられる。

OSLC (Open Services for Lifecycle Collaboration) は、ソフトウェア開発ツールの相互運用性を向上するという共通の目標を持つ企業・組織・個人から成るオープンコミュニティであり[2]、既に多くの企業が参

加して、ソフトウェア開発ライフサイクルのデータを連携するための共通の仕様を定めるために活動を始めている。これは、課題である CASE ツール間連携のための標準化を、企業間の協力体制を築くことで実現しようとするものである。

標準化を期待する背景には、CASE ツールを導入する組織がベンダロックインを避けたいという意向がある。効果的な CASE ツールを導入しても、提供するベンダの経営環境の変化などにより継続的な利用が望めなくなった場合、過去に蓄積した設計情報を他の CASE ツールに移行することができなくなり、過去の投資が失われるだけでなく、生産性の大きな低下が懸念されるからである。OSLC の活動が進むことにより、複数のツールが比較・選択可能でかつ一定の対策を講じれば他の CASE ツールに設計情報を移行できることが保証された“高度な CASE 環境”が提供されれば、“買い換え可能な市場”が提供され、導入側がベンダロックインに陥るリスクも低減されることが期待できる。

### 2.3. リポジトリ運用の徹底

「リポジトリ運用の徹底」に関しては、IT 業界全体が CASE 環境の運用に関する一般的なルールと効果を理解し、リポジトリ活用の目的と方法を意識してその効果を損なうようなルール違反を犯さないように人材教育が徹底されることで、リスクを回避できるものと考察する。

また、“リポジトリ”として整備される「設計情報データベース」が完成すると、利用者の知的資産になるとともに、他の下流 CASE ツールを利用する際にも継続的に利用することができる。これは、上流 CASE ツールが整備を促す項目の多くが、一般的な下流 CASE ツールが要求する「設計情報」の項目と一致するためである。上流 CASE ツールの中には、リポジトリを介して既に複数の下流 CASE ツールとの連携を実現している製品もあり、効果を意識したリポジトリの運用について、国内の IT 業界内にも経験とノウハウが蓄積されつつあることが推測できる。

## 3. CASE 環境の現状と課題

CASE の概念は広範にわたるが、市販に至っている CASE ツールは主に、上流 CASE ツールと下流 CASE ツールに分類できる。それぞれについて、現状を整理して課題を認識する。

### 3.1. 上流 CASE の課題

上流 CASE ツールは、下流工程に伝達すべき「設計情報」を、整合性を保ちつつ漏れなく整備することを助けるものであり、単体で工数削減や期間短縮の効果はない。但し、下流 CASE ツールとの連携を想定した CASE 環境の利用者が、早期に漏れなく「設計情報」を整備するために、また、下流 CASE ツールに伝達する「設計情報」をデータベース化し、コンピュータ上で効率的に伝達するために、上流 CASE ツールを利用する事例がみられるようになった[3]。

上流 CASE ツールを利用して蓄積される「設計情報」は、“リポジトリ”として整備され、下流 CASE ツールを効果的に利用するための準備作業も併せて実施することを促してくれるため、下流 CASE ツールを利用してコスト削減と期間短縮の効果を得ようとする利用者が構築すべき“RC<sup>2</sup>E”の一部として位置づければ、高い利用価値が見込まれる。しかし単体では大きな効果が得られないため、“RC<sup>2</sup>E”の中に位置づけて運用することが課題になる。

### 3.2. 下流 CASE の動向

近年、下流 CASE ツールという言葉はあまり使われていないが、代表的な機能の一つであるソフトウェアの自動生成機能は実用性が高まっており、いくつかの製品が日経コンピュータ誌[3]などで取り上げられる機会が増えている。取り上げられる事例には、従来主流だった、徹底した設計情報の整備とイン

プットによって完成形に近いソフトウェアの生成を目指す“完成型生成ツール”だけでなく、より少ない設計情報からとりあえず実用に耐えるレベルの試作品に近いソフトウェアの生成を目指す“試作型生成ツール（試作型の下流 CASE ツール）”が加わっている。以後、前者を“完成型生成ツール”，後者を“試作型生成ツール”と呼ぶこととする。

“試作型生成ツール”は優れた GUI を持ち，“完成型生成ツール”が要求する文字や特定の図化技法による定義よりも直観的な，図示や画面での描画による定義を優先してインプット情報を得ることに特徴がある。“完成型生成ツール”は要件定義や基本設計の結果を文字や特定の図化技法で表現できるプロの SE が使用することを想定して開発されているのに対し，“試作型生成ツール”はエンドユーザが直接利用して，試験的な運用を含めた業務改善を自ら行うことを視野に入れて開発されている。

エンドユーザは“試作型生成ツール”を手軽に使うことができ，手作業で行われている業務や新しく発生した業務を自らコンピュータ化していくことができる。

しかし現在，“試作型生成ツール”は“RC<sup>2</sup>E”と連携しておらず，自動生成されるソフトウェアも単独で存在するのみである。

### 3.3. RC<sup>2</sup>E

ソフトウェア工学におけるリポジトリの機能もまた広範にわたるが，上流工程と下流工程の連携を目的とする CASE 環境の実用化に際しては，上流工程のアウトプットである「設計情報」を下流工程に伝える「設計情報データベース」としての役割が期待される。本稿では，この目的と役割の範囲で“リポジトリ”という用語を使うこととする。

上流・下流の両 CASE ツールを利用する環境における設計情報の連携は，上流 CASE ツールのアウトプットが「リポジトリ」を介して下流 CASE ツールのインプットとして引き継がれることで実現される。

この，“リポジトリ”を中心として上流 CASE ツールと下流 CASE ツールが高度に連携する CASE 環境」を「Repository Centric CASE Environment」と呼び，“RC<sup>2</sup>E”と表記する。

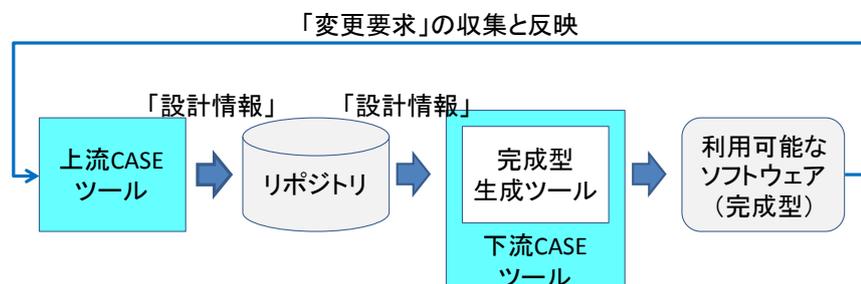


図1 “RC<sup>2</sup>E” (Level-1) の概念図

図1は，“RC<sup>2</sup>E” (Level-1) の概念図である。完成型ソフトウェアの提供を目指す従来型のシステム開発案件では，「上流 CASE ツール」から「リポジトリ」を介して「完成型生成ツール」を利用することで「利用可能なソフトウェア（完成型）」に至る青い太矢印のルートをとる。次期システムの開発は青い細矢印のルートで行い，リポジトリを介さないプログラムの改編は行わない。

だがこれは“RC<sup>2</sup>E”としては Level-1 のモデルである。従来からあるこのモデルには課題がある。システムを新規開発する機会が減り，保守が中心となっている現在，下流 CASE ツールを活用して効果を得るためには，現在のシステムの仕様を，一度全て上流 CASE ツールを使ってリポジトリ内に再現しなければならないが，これは容易ではない。この最初のハードルが，FFIEC(米国連邦金融機関検査協議会)の指摘から漏れている，CASE 環境普及のさらなる課題となっている。

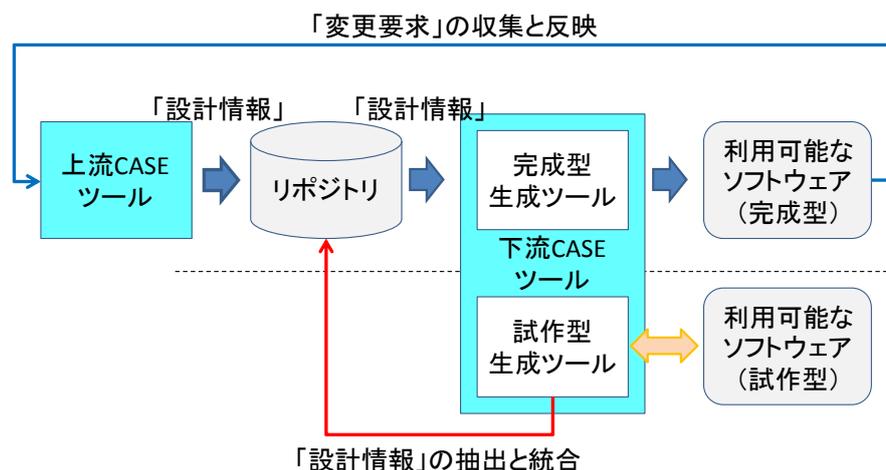


図2 “RC<sup>2</sup>E” (Level-2) の概念図

図2は、目指すべき“RC<sup>2</sup>E” (Level-2) の概念図である。Level-1 のモデルに，“試作型生成ツール”と設計情報の抽出ルート（赤矢印）を加え、リポジトリ環境への新しい合流ルートを作る。

ユーザによる業務改善型のシステム開発では，“試作型生成ツール”と“利用可能なソフトウェア（試作型）”の間を行き来しながら業務への適合度を高めると共に、ツール内にあるモデルも完成度を高めていく。業務運用が安定した段階で「設計情報」の抽出と統合を行い、「リポジトリ」にその設計情報を取り込んだ上で，“完成型生成ツール”を利用して“利用可能なソフトウェア（試作型）”の機能を“利用可能なソフトウェア（完成型）”の一部に統合し、廃止する。以後は、Level-1 の“RC<sup>2</sup>E”と同様である。エンドユーザはまた、新しい業務やコンピュータ化されていない業務を対象に“試作型生成ツール”を利用して、コンピュータの利用範囲を徐々に広げていくことができる。

### 3.4. RC<sup>2</sup>E (Level-2) 普及の課題

現段階では，“試作型生成ツール”のベンダは単独で機能することを意図しており，“設計情報”をアウトプットすることを想定していない。そのため、図2の上半分と点線以下の下半分の世界は断絶している。

そこで、下流 CASE ツールに引き渡す必須のリポジトリ項目を認識するために、すでに複数の下流 CASE ツールとの連携を果たした上流 CASE ツールベンダを巻き込み、基本骨格とすべき連携項目を標準化し、この連携項目を“試作型生成ツール”にも適用する。こうして“試作型生成ツール”内の「設計情報」を抽出してリポジトリに連携できるようにした Level-2 の“RC<sup>2</sup>E”を構築することができれば、CASE 環境導入のハードルが下がり、さらに利活用を促進することができると考察した。

この標準は、ベンダロックインを避けたい利用者の意向を反映し、オープンな環境で複数ベンダもしくはCASE ツール業界全体で共有しなければならない。

## 4. おわりに

本稿では、リポジトリを中心として高度に連携する CASE 環境を“RC<sup>2</sup>E”と定義し、その普及に向けた課題について考察した。但し本稿は、未検証な仮説に基づいてまとめられたものであり、後続の研究による実証を大いに期待するものである。

### 参考文献

- [1] <http://ithandbook.ffiec.gov/it-booklets/development-and-acquisition/development-procedures/software-development-techniques/computer-aided-software-engineering.aspx?prev=1>
- [2] <https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/OSLC%20Japan%20Wi>

ki/page/1.OSLC%20%E3%81%A8%E3%81%AF

[3] 井上英明, “日経コンピュータ 2012-3/15 特集「超高速開発」が日本を救う”, 日経 BP, 2012, pp.28-45.