

PCTE におけるツール統合のための標準データモデルへの提言\*

3K-9

梅本 肇<sup>†</sup> 山岸 利行<sup>‡</sup>

†日立西部ソフトウェア(株) ‡(株)日立製作所 ビジネスシステム開発センタ

1 はじめに

昨今、統合CASE環境としてPCTE<sup>1</sup>が注目されている。しかし、単にPCTEを用いただけではCASEツールの統合は達成できず、統合を実現するための標準データモデルが必要である。

上記目的のために、CASEツール群全体の構成に着目したSDS<sup>2</sup>と、各ツール毎のデータ構成に着目したSDSを作成することを提案する。また、各ツールをクラス分類することにより、ソフトウェア開発ライフサイクルの支援を行なうことができることを述べる。

2 標準データモデルの検討

PCTEでは、データモデルを複数のSDSを用いて表現することができる。この特長を生かすかたちで、以下のようなモデル化を考え、各々の長所短所を検討した。

- (a) 成果物すべてを一つのモデルで表現
- (b) バッチ処理用など用途に応じてモデルを作成
- (c) 成果物単位にモデルを作成

(a)のモデルは、その複雑さから設計・改良がしづらい。(b)のモデルは、成果物間の関係が見易いという特長があるが、用途がユーザ毎に異なるため標準化が難しいと思われる。(c)のモデルは、成果物の構成管理を行なうモデルを別途必要とするが、

- モデルの複雑さを軽減する
- 構成管理をするのにツールとの対応付けが可能になる(後述)
- 成果物間でのデータ共有によりデータの重複を減らせる

ことが期待できる。よって標準データモデルは成果物単位で提供されるべきであると考えた。また、個々の成果物のデータを管理するデータモデルの設計を、成果物中のデータの構成に注目して行なうことにより、前述の構成管理を行なうモデルとの視点の一貫性を保つことが可能になるという利点があることも分った。

以上より、CASEの扱うデータの構成を管理するには、成果物を管理するデータモデル(図1)と、個々の成果物中のデータを管理するデータモデル(図2)の二段階に分けて表現し、各々のデータモデルの中で用いられる各データの関係を構成リンクを用いて表現すれば良いという結論に達した。

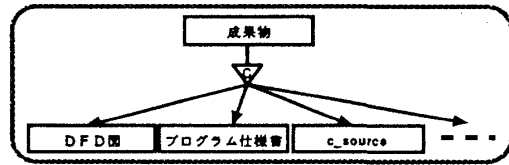


図1: 成果物の構成

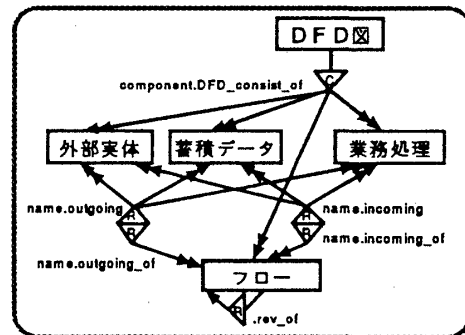


図2: DFD図のデータモデルの例

3 ツールのクラス化と制御統合のためのデータモデル

ソフトウェア開発ライフサイクルの支援は、統合CASEで実現すべきサービスである。しかし、ライフサイクルは、前述のモデル化だけでは支援できない。ライフサイクル支援は制御統合で提供すべきであると捉え、前述のモデルに付加するかたちで、その実現方法について検討した。

3.1 ツール間のデータの共有

ソフトウェア開発ライフサイクルは、情報の変換関係から見ると、

1. 前工程で作成された成果物から、現在作成している成果物に必要なデータを取り出し、
2. 現工程で新たに必要となるデータを付加する

という作業の連鎖であると見ることができ、この作業をSEE<sup>3</sup>で支援する形態として次の2つが考えられる。

- (a) データの共有
 

複数のツール間でその扱うデータ構造の一部を共有する。個々のツールは共有されていない部分すなわち前工程から受け継がれないデータを補完する作業を支援する。(図3)
- (b) ツールによる変換
 

ツール毎にデータを管理し、ライフサイクルの支援はデータ間の変換を行なうツールによって実現する。(図4)

\*Standard Data Model for Tool Integration in PCTE Environment by Hajimu UMEMOTO, Toshiyuki YAMAGISHI

<sup>1</sup>Hitachi Seibu Software, Ltd.

<sup>2</sup>Institute of Advanced Business Systems, Hitachi, Ltd.

<sup>3</sup>Portable Common Tool Environment

<sup>4</sup>Schema Definition Set

<sup>3</sup>Software Engineering Environment

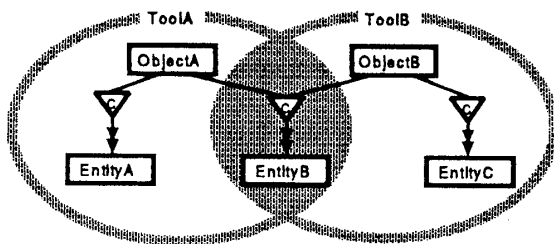


図 3: ツール間のデータ共有 (a)

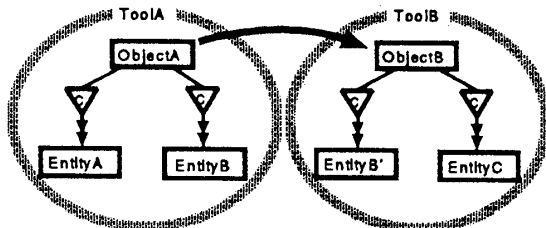


図 4: ツール間のデータ共有 (b)

(a) の方式によりツール間でのオブジェクトの共有を行なうには、個々のツールが扱うデータを、すべて別々のオブジェクトとして保持できるオブジェクトの粒度が実現され、型のインポートなどを用いてうまくモデル間で対応がとれている必要がある。しかし、現実には粒度などの問題がある上、ソフトウェア開発ライフサイクル全体を通してすべてが(a)により支援されるとは考えにくい。ゆえに、ここでは(b)の方式について検討を行なった。

### 3.2 ツールのクラスによる分類

ソフトウェア開発に使用されているツールは、それが扱っている成果物の種別と提供する機能に注目して分類することができる。この様な視点から成果物とそれを扱うツールの組として分類したものをツールクラスと定義する。

ある成果物に対してツールが行なう機能に着目すると、

- 単一の成果物をユーザが加工するためのユーザインタフェース機能
- ある成果物から他の成果物への内容の変換機能がある。これらの機能単位ごとにツール化し成果物と対応付けたものを、それぞれ編集ツールクラスと生成ツールクラスと呼ぶことにする。

ツールクラスと、編集ツールクラスおよび生成ツールクラスは IS-A の関係である。成果物は種別ごとに分類される。ツールはクラスの提供するサービス、成果物はクラスの属性に各々対応する。(図 5)

このようにモデル化すると、開発者の作業から見ると、成果物を編集ツールクラスで作成することが開発作業そのものであり、成果物間の変換は生成ツールクラスで行なうと捉えることができる。この変換関係をライフサイクルに基づいたモデルとして表現すれば、ライフサイクルに基づいた支援をすることができる。

### 3.3 ライフサイクル支援モデルとの対応

前述の成果物の構成管理を行なうモデルでは、個々の成果物間の変換関係は記述されていない。この変換

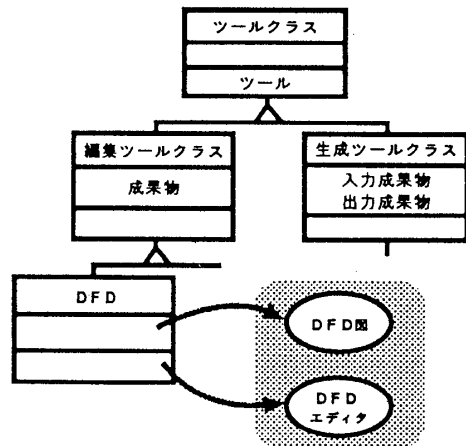


図 5: ツールクラス

関係は、成果物の構成管理を行なうモデルの個々のオブジェクトをインポートし、ライフサイクルに基づいて参照リンクで結ぶことで作成できる。

編集ツールクラスは特定のオブジェクトを操作するツールのクラスであり、生成ツールクラスはオブジェクトから他のオブジェクトへの変換を行なうツールのクラスである。ゆえに、上記方法でライフサイクル支援モデルを作成すると、オブジェクトを編集ツールクラスに、参照リンクを生成ツールクラスに、それぞれ対応付けることができる。(図 6)

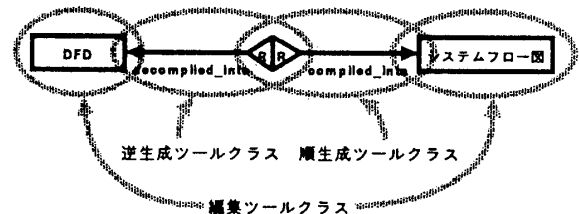


図 6: ライフサイクル支援モデルとツールクラス

## 4 おわりに

PCTE 上でデータ統合を実現するための標準データモデルを提案した。その中で、次の 2 点について述べた。

1. 成果物を管理するモデルと個々の成果物中のデータを管理するモデルに分離し、各々のモデルについて構成の観点から表現すべきである。
2. 成果物単位でデータを管理するモデルを用意することにより、成果物とそれを扱うツールの組でツールクラスを作成できる。これにより、ソフトウェア開発ライフサイクルを支援するためのモデルとツールを対応付けすることができる。

### 参考文献

- [1] (財) 日本規格協会 “ソフトウェア開発支援ツールの標準化研究” 報告書, Mar. 1993
- [2] 嵯坂恒夫 沢田篤史 満田成紀 “チュートリアルー開放型 CASE プラットフォーム”, コンピュータソフトウェア Vol.10 No.2 Mar. 1993
- [3] 嵯坂恒夫 “ソフトウェア評論 — Emerald PCTE”, コンピュータソフトウェア Vol.10 No.2 Mar. 1993