

# 3U-2 CASE構築のための ソフトウェアデータモデル

永松 祐嗣  
(株)日立製作所 システム開発研究所

## 1. はじめに

近年、ソフトウェアライフサイクル全般を効果的に支援する統合型CASE環境が盛んに提案されている[1]。CASEを実用的にするためには、開発プロジェクトに柔軟に適合するCASE環境を容易に構築(カスタマイズ)できる環境(メタCASE)が必要となる。

メタCASEの要件として、ソフトウェア仕様(プロダクト)やソフトウェアの開発プロセス、プロジェクトの管理手順を形式的に扱う必要がある。従来提案されているリポジトリデータモデルでは、開発プロジェクトに応じて柔軟にモデルをカスタマイズするだけの十分な表現力がない。本論文では、ソフトウェア開発に関わる生産物(仕様書)、開発手順、管理手順を形式的かつ統一的に表現できるデータモデルの1手法について述べる。

## 2. 基本モデル

ソフトウェアデータモデルの骨格として我々は、図1に示すものを設定した。

標準モデルは、プロジェクト共通のモデル、プロジェクトモデルは、標準モデルをカスタマイズしたプロジェクト固有のモデルを扱う。さらにプロジェクトモデルのプロダクトモデルに基づいて、具体的なデータを設定した仕様書のモデルをプロダクトインスタンスモデルとする。

プロダクトモデルは、ソフトウェア開発に関わる生産物のモデルであり、ソフトウェア設計/テスト仕様だけでなく、変更管理、進捗管理等の管理データも扱う。

プロセスモデルは、プロダクトモデルのソフトウェア設計/テスト仕様に対する開発手順のモデルである。また、プロジェクト管理モデルは、プロジェクトの人、物(設備)、金、期間に関する管理情報のモデルである。

プロセスモデルは、プロダクトモデルをデータとして操作演算方法を表現していると捕らえることができ、その意味でプロセスモデルは、プロダクトモデルのメタモデルである。

同様に、プロジェクト管理モデルは、プロダクトモデルと、プロセスモデルのメタモデルと考えられる。

一方で、標準モデルは、プロジェクトモデルのテンプレート(型)であり、プロジェクトモデル内のプロダクトモデルは、プロダクトインスタンスモデルの型と考えることができる。オブジェクト指向モデルでは、このメタと型の2種類の関係を、メタインスタンスの1種類の関係で表現しおり非常に混乱する場合がある。ここでは、以下の関係を設定する。

- (1)メタリンク：プロダクトモデル、プロセスモデル、プロジェクト管理モデル間の結合を表現するリンク
- (2)テンプレートリンク：標準モデル、プロジェクトモデル、プロダクトインスタンスモデル間の結合を表現するリンク

## 3. 詳細モデル

2で述べたプロダクト、プロセス、プロジェクト管理モデルの具体的な表現として、オブジェクト指向モデル[2][3]をベースにした。オブジェクト指向モデルでは、プロダクト(仕様)の意味モデルとして、以下の3つのモデルを設定する。

- (1)情報モデル：情報の主体とそれらの関係(構造)モデル
- (2)状態遷移モデル：状態と状態の遷移を起こすイベントのモデル
- (3)データフロー(関数)モデル：データとデータの変換操作(プロセス)のモデル

我々は、上記モデルをプロダクトモデルだけでなく、プロセス、プロジェクトの意味モデルにも用いることにする。即ち、開発(管理)手順自体を1つのオブジェ

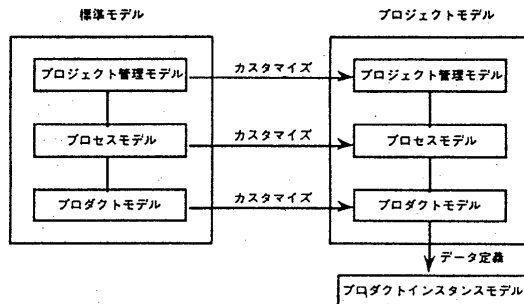


図1 ソフトウェアデータモデルスケルトン

クト(エンティティ)として、その関係や構造を情報モデルで表現する。また、各開発(管理)手順がどのように同期をとりながら実行していくか時間的な振る舞いを状態遷移モデルで表現し、機能的な振る舞いをデータフローモデルで表現する。

このような情報モデル、状態遷移モデル、データフローモデルの統合的な表現として、以下を設定した。

情報概念の記述は、既存のE-Rモデルにオブジェクト指向のIS-A,HAS-A,INSTANCE-METAの関係を表す特別な記法を用意した。このINSTANCE-METAの関係は、同一モデル内の関係であり、2章で述べたメタリンクとは異なる。また、エンティティの属性やエンティティ自身に、直積,直和,列や再帰による構造表現も用意した。

状態概念とプロセス概念を表現するため上記のE-Rモデルの中に以下のトークンモデルを導入した。即ち、エンティティを状態、あるいはプロセスとみなす。エンティティとエンティティを結ぶ線として、関係を表す線とは別に、データ線と、コントロール線を設定する。各エンティティには、属性として、トークンを複数個保持でき(デフォルトは1つ)、複数個トークンを保持するエンティティは、データ蓄積を行うエンティティと解釈できる。トークンは、データか、イベントであり、イベントは構造を持たないアトムデータである。トークンは1個づつ、データ線か、コントロール線を流れる。トークンは、AND/ORの分配エンティティと、結合エンティティにより流れを制御される。また、特別なエンティティとしてトークンのSOURCE/SINKを設定した。

4. モデル例

図2にプロジェクトモデルにおけるソフトウェアデータモデルの例を示す。ここで表現されていることは、以下のような事柄である。

- (1)Product Modelとして、E-Rモデル仕様があり、E-Rモデル仕様は、Entityとその間のRelationを要素として、複数回適用することにより構成される。
- (2) Process Modelにおいて、分析作業の中にE-Rモデルを設定する作業があり、E-Rモデルを設定する作業は、Entityを定義して、その後にRelationを定義する作業を複数回行う。
- (3)Process ModelにおけるEntityを定義する作業は、Product ModelにおけるE-Rモデル仕様中のEntityを定義することである。Relationを定義する作業も同様(メタリンク関係により結合している)。
- (4)Project Management Modelにおいて、人員計画作業があり、人員としてA,Bがいる。
- (5)Project Management Modelにおいて、人員Aは、Process ModelにおけるE-Rモデル設定作業を担当して、人員Bは、Process Modelにおける分析作業を

担当している。

5. おわりに

CASE構築のためのソフトウェアデータモデルについて述べた。本モデルは、データベースにおける概念スキーマに相当する。今後内部スキーマ、外部スキーマを設計し、さらに、データ定義/操作言語を開発することにより、メタCASEシステムを実現していく予定である。

[参考文献]

- [1] M.Chen, R.J.Norman: A Framework for Integrated CASE, IEEE Software, Vol.9, No.2, 1992, pp18-22.
- [2] J. RumBaugh, et. al: Object-Oriented Modeling and Design, Prentice-Hall, Inc 1991.
- [3] S. Shlaer, S. J. Mellor: Object Lifecycles Modeling the World in States, Yourdon Press, 1992.

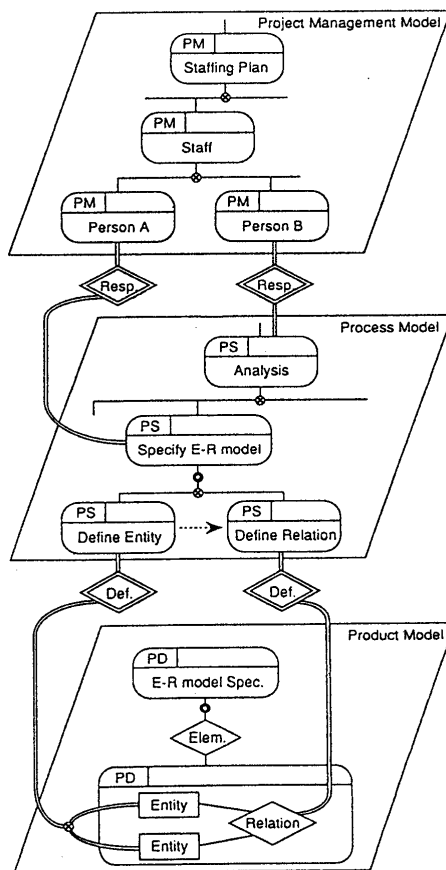


図2 ソフトウェアデータモデル例