

汎用オブジェクト指向CASE

2K-4

高館 公人 千葉 寛之 秋庭 圭子 島田 直樹 柳町 美帆 青山 和之

(株)日立製作所

1. はじめに

様々なオブジェクト指向方法論に対応できる、クラス間の静的な関係を定義するオブジェクト図のエディタを開発した。様々な方法論に対処可能とするために、方法論によって大きく異なる仕様の表現を扱う層と、方法論によって違いの少ない仕様の意味を扱う層を分離した。意味を扱う層は、様々な方法論が提供する意味を全て包含するようなモデルを扱う。表現を扱う層は、インタプリタ言語で記述し、実行時に解釈することによって、新たな方法論にも対処できる。

2. 方法論における違い

2.1 モデルの比較

現在までに様々なオブジェクト指向方法論が提案されている。分析工程のみを支援する方法論、分析からテスト工程までを支援する方法論など様々な方法論があるが、方法論で提供されているモデルを分類すると、表1に示す6つに分類できる。

表1 オブジェクト指向方法論のモデルの分類

表現範囲	モデル種別	内容
全体	静的仕様	クラス間の構造的な関係。
	動的仕様	クラス間のメッセージ通信。
クラス	静的仕様	クラスの有する属性とメソッド。
	動的仕様	クラスの状態遷移。
メソッド	静的仕様	メソッドのインタフェース。
	動的仕様	メソッドのアルゴリズム。

上記分類に従って、各方法論で提供されているモデルを分類すると表2のようになる。[1, 2, 3]

表2 オブジェクト指向方法論の仕様書

範囲	種別	Coad&Yourdon	OMT	Booch
全体	静的	OOA図	オブジェクト図	クラス図
	動的		インタレース図	オブジェクト図
クラス	静的	テンプレート	クラス仕様	テンプレート
	動的	状態遷移図	状態遷移図	状態遷移図
メソッド	静的	-	メソッド仕様	テンプレート
	動的	サービス図	データフロー図	状態遷移図

Generic Object-Oriented CASE;

Masato TAKADACHI, Hiroyuki CHIBA, Keiko AKIBA,
Naoki SHIMADA, Miho YANAGIMACHI, Kazuyuki AOYAMA
Hitachi, Ltd.

各方法論で呼び方は異なるが、クラス間の関係を示すオブジェクト図と、クラス内の状態遷移を示す状態遷移図は全ての方法論で用意している。クラスの静的構造、メソッドの静的/動的構造を記述するモデルを用意しているか否かは、方法論によって異なる。特にメソッドの動的構造を表現するモデルは、Coad&Yourdonではフローチャートの様なサービス図、OMTではデータフロー図、BoochのOODでは状態推移図であり、方法論によって全くモデルが異なっている。

2.2 記法の比較

3つの方法論で用意されているオブジェクト図の表現方法の一部を比較した結果を表3に示す。同じ意味でも方法論によって表現方法は異なる。例えば、クラス間の包含関係を、OMT、Coad&Yourdonでは、ツリー構造で表現するが、BoochのOODではクラス的位置関係で表現している。表現している意味の数も方法論によって違いがある。しかし、表現している意味の数は、全ての方法論をあわせても、10程度である。

表3 オブジェクト図の記法

方法論	Coad&Yourdon	OMT	Booch
クラス			
インスタンス			
汎化/特化			
包含関係			

2.3 用語の比較

意味に対する呼び方が方法論によって異なる。方法論による用語の違いを表4に示す。例えばクラス間の包含関係を、OMTでは"Aggregation"と呼び、Coad&Yourdonでは"Whole-Part Structure"と呼ぶ。日本語訳も"集約"、"集成"、"包含"と様々に訳されている。

表4 方法論で用いる用語

Coad-Yourdon	OMT	Booch
Class	Class	Class
Object	Object	Object
Gen-Spec Structure	Generalization	Inherits
Whole-Part Structure	Aggregation	-

この他にも、方法論によって、支援する工程の範囲や手順等、異なる点は数多く存在するが、1つのエディタを汎用化する観点からは無関係なので省略する。

3. 汎用化方式

2章で記述した3つの項目を利用者が定義可能とすることで、エディタを様々な方法論に対処可能にする。

3.1 モデルの汎用化

状態遷移図やデータフロー図等のモデルを編集するエディタと、仕様書の種別の対応関係を利用者が自由に定義可能とした。これによって、メソッドを定義した仕様書を状態遷移図でもデータフロー図でも記述できる。

さらに、オブジェクト図のエディタは、仕様書の編集中に、クラスやメソッドの詳細を定義するエディタを起動できる。詳細を定義するエディタは方法論によって異なるので、クラスやメソッドを選択した際に起動するエディタを、利用者が定義できるようにした。

3.2 記法の汎用化

記法を扱う層をインタプリタ言語で記述し、実行時に解釈する方式とした。記法を扱う層を表現層と呼び、意味を扱う層を意味層と呼ぶ。

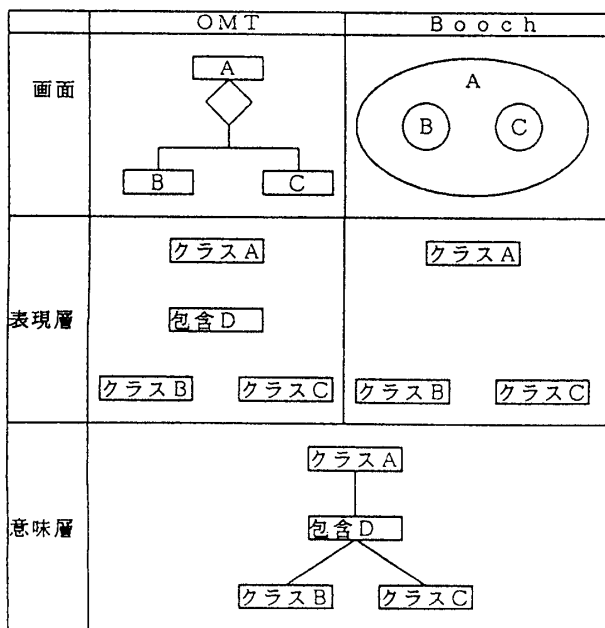


図1 記法の汎用化方式

意味層は、方法論で用意しているオブジェクト図の概念を全て包含するような総括的なモデルを扱う。

表現層は、画面に表示されるオブジェクトの大きさ、座標等を管理する。意味層は、オブジェクト間の関係を管理する。現状では、表現層の情報と意味層の情報は別々のファイルに格納され、リポジトリに格納される。将来的には、意味層の情報はリポジトリにデータモデルとして展開する。

3.3 用語の汎用化

標準用語を用意し、標準用語に別名を設定できるようにした。エディタの使用中表示されるメッセージの中に標準用語が含まれていて、それに対して別名が定義されている場合には、メッセージは別名に置き換わって表示される。

4. 評価

4.1 汎用性

エディタが扱う意味を利用者が定義可能とすることにより、モデルの汎用化が実現できる。つまり、オブジェクト図、状態遷移図にも対応可能なエディタとなる。

今回開発したエディタでは、オブジェクト図の汎用化を実現した。汎用化のレベルは1段低いですが、クラス間の汎化関係を包含関係に置換する機能等、オブジェクト図固有の機能を実現できた。

4.2 性能

対話性能に問題は無いが、エディタの起動時の性能が悪い。これは、起動時にインタプリタ言語で記述された表現層のファイルを読み込み、メモリに展開しているためである。

5. まとめ

様々な方法論に対応するオブジェクト図のエディタを開発した。現在までにOMTとCoad&Yourdonのためのプレゼンテーション層をインタプリタ言語で記述した。今後、Booch等様々な方法論毎にプレゼンテーション層を記述するとともに、状態遷移図エディタ、イベントトレース図エディタ等を開発していく。

6. 参考文献

- 1) G.Booch "Object-Oriented Design with Application" The Benjamin/Cummings 1991
- 2) P.Coad, E.Youdon "Object-Oriented Analysis" Yourdon Press 1991
- 3) J.Rumbaugh 他 "Object-Oriented Modeling and Design" Prentice-Hall 1991