

OMTにおけるモデルの一貫性について

波瀾 由布子[†] 合原 正男[†] 池辺 八洲彦^{††} 蔡 東生[†] 宇都宮 公訓[†]

4R-2

[†]筑波大学 電子・情報工学系 ^{††}会津大学 コンピュータ理工学部

1 はじめに

ソフトウェア開発においては、オブジェクト指向パラダイムへ、着実にパラダイムシフトしつつある。いくつかのオブジェクト指向ソフトウェア開発技法が提案されているが、中でも、注目されている技法の一つに、J.Rumbaugh等によるOMT(Object Modeling Technique)がある。OMTのための上流CASEツールとして、StP/OMTなどがあるが、著者等は、さらに強力なCASEツールの開発を目指して、OMTモデルの一貫性チェックの支援に関して検討を行なったので、報告する。

OMTでは、開発対象システムを、オブジェクトの静的な関係、オブジェクトの動的な振舞い、データ間の依存関係の三つの側面から、それぞれオブジェクトモデル、動的モデル、機能モデルと呼ばれる三つのモデルに分けて、記述する。思考の生産性のため、三つのモデルに分割してモデル化するが、これら三つのモデルは独立ではなく、相互に関係し合っており、一つの対象システムの意味的に統合されたモデルとして、それらの間で、一貫性が保たれていなければならない。もちろん、三つのモデルそれぞれとしても、一貫性が保たれていなければならない。

OMTでは、分析フェーズで作られたモデルが、そのまま、設計フェーズ、実装フェーズに引き継がれる。例えば、設計は、分析フェーズの(出力としての)モデルを詳細化することによって行なう。その意味で、各フェーズごとのモデルが存在するが、ここでは、分析フェーズで作られたモデルを対象にしている。一貫性チェックはできるかぎり自動化すべきであるが、自動化が難しいことも多い。その場合は、コンピュータでチェックリストを生成し、分析者にチェックをしてもらうアプローチを探ってい

る。2節、3節に主要なチェック項目を列挙する。

2 モデル内での一貫性チェック

オブジェクトモデルに関しては、以下のチェック等を行なう。

- (1) 同一クラス内の関連において、両端で役割を指定しているか
- (2) 2項関連の両端で指定する役割は、互いに反対の立場を意味するものであることの確認を求める
- (3) {ordered} 指定の側の多重度は複数か
- (4) {subset} 指定で矢の頭がある側は複数か
- (5) 多重継承において、(黒塗り三角形で表記された)互いに素でない汎化クラスからの継承か
- (6) 多重継承において、独立した汎化階層に属するクラスからの継承か
- (7) 制限クラスにおいて、制限された属性を更新する可能性のある操作が継承されていないか (R属性の指定を導入)
- (8) 操作の継承で、シグニチャに矛盾がないか
- (9) 操作でアクセスしている属性へのアクセス経路が存在するか
- (10) 多重度が複数指定のとき、オブジェクトを一意的に識別できるか
- (11) 導出属性を変更していないか

動的モデルに関しては、以下のチェック等を行なう。

- (1) 独立したサブグラフがないか (並行状態で独立したサブグラフを除く)
- (2) ある状態から複数の出遷移があり、それらの遷移を引き起こす事象が、異なるオブジェクトによって作られていると判断できる場合、その状態と遷移を表示して確認を求める
- (3) ある状態からの複数の自動遷移があるとき、それらすべての遷移にガード条件が付いているか

Consistency Checks on OMT Models

by Yuko Namikata[†], Masao Aihara[†], Yasuhiko Ikebe^{††}, Dong-Sheng Cai[†], and Kiminori Utsunomiya[†][†]University of Tsukuba ^{††}University of Aizu

- (4) ある状態から複数の遷移が出ているとき、ガード条件の付かない自動遷移が含まれていないか
- (5) 状態図のスーパー、サブの関係は一貫しているか
- (6) ある状態図が単発型であるべきかどうか
- (7) 単発型の状態図において、一つの初期状態と、一つ以上の最終状態があるか
- (8) ある層の状態から、それより下位の層の状態図中の状態への直接の遷移、あるいは上位の層の状態図への直接の遷移を表わすとき、Harelの等高線記法によっているか
- (9) Harelの等高線記法で、等高線への入遷移があるとき、その等高線で表わす状態の展開中に初期状態が一つ指定されているか
- (10) Harelの等高線で表わされる状態から、上位の層に向う出遷移が一つ以上あるか
- (11) 入れ子の状態図中の(サブ)状態への入遷移や出遷移において、遷移がいくつかの階層にまたがる場合には、その遷移によって起こる動作の順序を表示して確認を求める
- (12) 遷移の詳細化の階層に対応して、事象の詳細化階層を作って表示し確認を求める
- (13) 事象説明書の内容と状態図の間で、事象の受渡しに矛盾がないか
- (14) ガード条件付き遷移をもとにして、オブジェクトの依存関係を表示し確認を求める
- (15) 状態図で並行サブ状態をもつ状態が、状態説明書でも並行サブ状態をもつと定義されているか

機能モデルに関しては、以下のチェック等を行なう。

- (1) 1枚のデータフロー図中に独立なサブグラフが存在しないか
 - (2) システムの最上位のデータフロー図で、データフロー図とその外部との境界がアクターか
 - (3) 詳細化された(下位の層として展開された)プロセスに対するデータフロー図で、図全体として入ってくるデータフロー、出て行くデータフローが、(その上位の)プロセスに入ってくるデータフローと、出て行くデータフローに一致しているか
 - (4) プロセスの再帰的な展開は、注意を促すため、その旨を表示して確認を求める
 - (5) オブジェクトの生成を表わす白抜き三角形の先はデータストアか
- ### 3 モデル間での一貫性チェック
- (1) 動的モデルが、オブジェクトモデルで指定された集約構造を反映するように作られているか
 - (2) 分割されたサブ状態で実行される活動が、下位層で展開された状態図においても、複合状態として、矛盾なく記述されているか
 - (3) 制限による汎化において、オブジェクトモデルの汎化構造と動的モデルの汎化構造が等価か
 - (4) 拡張による汎化において、属性の拡張に関して、スーパークラスの状態図とサブクラスの状態図が互いに素か
 - (5) スーパークラスの状態図とサブクラスの状態図で、事象の受渡しに関して矛盾がないか
 - (6) 集約値を分解、併合するデータフローがデータ辞書と矛盾しないか
 - (7) 操作仕様がデータフロー図から得られるか
 - (8) データストアに対する操作が、データフロー図とデータ辞書で矛盾しないか
 - (9) オブジェクトモデルで属性として扱われている問合せ操作が、動的モデルの状態定義に使われていないか
- ### 4 おわりに
- 上記の一貫性チェックを行なうチェッカを開発中であり、早期に完成させ、効果を確認することになっている。さらに、コメントによる制約を増やして一貫性チェックの精度を上げること、設計フェーズ後のモデルに対する一貫性チェッカの開発も考えている。第2世代OMT、OMTとBooch法の統合にも対処して行く予定である。