

上村匡稔\*

小川満†

宇都宮公訓‡

筑波大学 理工学研究科

筑波大学 情報学類

筑波大学 電子・情報工学系

### 1. はじめに

著者等は先に OMT におけるモデルの一貫性チェックについて報告[1, 2]した。その後、オブジェクト指向ソフトウェア開発技法は進歩し、モデリング言語として UML(Unified Modeling Language)[3, 5]が、モデルの制約記述言語として OCL(Object Constraint Language)[3, 4]が開発され、標準的に使われるようになってきた。これを機に、クラス構造図について、[1, 2]の内容を洗い直すとともに、新たな観点から一貫性条件を追加することを試みたので報告する。

### 2. モデルの記述と一貫性チェック

一般に、モデルはその精度と正当性の確度を高めるため、より完全に、より詳細に、同一の内容についても複数の視点から記述することが望ましい。このように記述されたモデルに対して、以下のような立場から無矛盾性、異常性、完備性を検査することをここでは一貫性チェックと呼んでいる。

- ・ UML および OCL で記述された内容の意味論的な一貫性
- ・ 方法論で決まる記述規約に関する一貫性
- ・ 制約による包括的テスト条件(generic test requirement)の記述に関する一貫性[6]

---

Consistency Check for Class Diagram in UML and OCL

\*Tadatoshi Kamimura, †Mitsuru Ogawa,

‡Kiminori Utsunomiya

\*Master's course in Sciences and Engineering, University of Tsukuba

†College of Information Sciences, University of Tsukuba

‡Institute of Information Sciences and Electronics,  
University of Tsukuba

・著者らが提案する基準に関する一貫性

記述されたモデルが UML や OCL の構文規則に違反しているかどうかのチェックは本論文の興味の外である。

次節で述べる一貫性条件をチェックするにはその内容がモデルの記述の中に含まれていなければならぬ。そのため、OCL を用いて積極的に制約を記述することを前提としている。次節の一貫性条件はクラス構造図中に制約として何を記述するかのガイドラインにもなっている。

### 3. 一貫性条件

以下に一貫性条件を列挙する。

- ・同一クラス内の関連では、役割を必ず指定する
- ・2 項関連の両端の役割は、互いに反対の立場を意味するものを指定する
- ・{ordered}が指定された関連端の多重度は、必ず複数にする
- ・{subset}の関係にある関連は、矢の頭があるほうの関連の多重度を超えない
- ・操作を再定義する際、継承される操作と振る舞いが異なるように再定義しない
- ・操作を再定義する際、シグニチャが矛盾しないようにする
- ・抽象操作は、汎化階層の最下層のクラスで指定しない
- ・操作でアクセスしている属性については、クラス図上でその属性へのアクセス経路が存在することを確認する
- ・多重度が複数のとき、特定のオブジェクトにアクセスする方法が与えられていることを確認

する

- ・関連クラスのインスタンスは、属する関連のインスタンスに対し一つしかないようにする
- ・弁別子を指定したサブタイプは、排他的であることを確認する
- ・実行時にタイプが変わる場合、変更できる範囲の汎化に<<dynamic>>を付加する
- ・{frozen}指定された関連端は、関連元が消滅するまで変化しない
- ・多重度が複数であるとき、インスタンスの構造を{hierarchy}や{dag}などと明記する。
- ・多重分類の有効な組み合わせには、{complete}で示したサブタイプを含む必要がある
- ・主キーを示す{key}がついた弁別子からの関連端は、0..1である
- ・派生属性、派生関連は派生元を明記する
- ・パラメータ拘束済みクラスは、操作や属性を追加しない
- ・クラス間の複数のパス（関連を遷移的にたどるもの）について、多重度の上限と下限は一致する
- ・操作でアクセスしている属性には、クラス図上でアクセス経路が存在する
- ・誘導可能性のない関連は、未決定か双方向関連のどちらであるか確認する
- ・{xor}関係にある複数の関連は、それらが同時に成り立たないか確認する
- ・互いに素でない汎化階層からの多重継承は、属性に曖昧さが残ることを確認する
- ・クラスの不变条件を記述する
- ・メソッドの事前条件、事後条件を記述する
- ・複数のインスタンス集合から部分集合を取り出して、サイズについて制約を記述する
- ・状態遷移のガード条件を記述する
- ・汎化においてスーパークラスとサブクラスの違いを記述する

- ・{unique}指定された属性は、すべてのインスタンス中でユニークな値を持つ
  - ・状態によって関連先の多重度が変化するような場合について記述する
  - ・関連について、反射律、対称律、推移律が成り立つかどうか明記する
- 特定の属性に関する制約を記述し、テストに役立てる。

#### 4. 今後の方向

モデルの精度、正当性の確度を高めるために制約として記述すべきことがらの開発、その記述言語としての OCL の拡張、動的モデルへの拡張、チェックの自動化等を予定している。

#### 5. 参考文献

- [1] 波瀬由布子、合原正男、宇都宮公訓他: OMTにおけるモデルの一貫性について、情報処理学会第52回全国大会講演論文集(5), pp. 103-104, 1996
- [2] 金田忠士、宇都宮公訓他: OMTにおけるオブジェクト設計の一貫性チェックについて、情報処理学会第54回全国大会講演論文集(1), pp. 423-424, 1997
- [3] OMG Unified Modeling Language Specification V1.3, <http://www.omg.org/uml>, June 1999
- [4] Jos Warmer, Anneke Kleppe: The Object Constraint Language, Addison-Wesley, 1999
- [5] Martin Fowler, 羽生田栄一監訳: UML モデリングのエッセンス 第2版, 翔泳社, 2000
- [6] Robert V. Binder, Testing Object-Oriented Systems: Addison-Wesley, 1999