

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

4U-9 金田 忠士[†] 松浦 江里^{††} 藤原 譲^{†††} 宇都宮 公訓^{†††}

[†]筑波大学 工学研究科 ^{††}筑波大学第三学群情報学類 ^{†††}筑波大学 電子・情報工学系

1 はじめに

オブジェクト指向ソフトウェア開発技法 OMT(Object Modeling Technique)[1]では、大きくは、分析、設計、実装、テストの順に作業が進行する。分析フェーズでは、問題記述、問題領域の知識を用いて、個別にオブジェクトモデル、動的モデル、機能モデルを作成した後、それらを統合し、分析結果の多くをオブジェクトモデルにまとめる。ユーザインターフェースの決定も分析フェーズで行なわれる。設計フェーズでは、システム設計を行なった後、オブジェクト設計を行なう。設計は実装に影響されないことが望ましいが、オペレーティングシステム、データベース、クラスライブラリなど、現実には、実装環境を考慮して設計を行なわざるを得ない[3]。

著者等は、OMTによるソフトウェア開発の機械支援の一環として、ドキュメントの一貫性チェックを考慮しており、分析フェーズでの一貫性チェックについては昨年報告した[2]。その後、[3]に沿って設計フェーズでの一貫性チェックについて検討したので報告する。

2 設計ドキュメント

OMTの設計フェーズでは、分析フェーズで得られたドキュメントを基に以下の詳細化を行なう。

- 分析フェーズに引続き、動的モデル、機能モデルの情報をオブジェクトモデルに統合
- 操作を実装するアルゴリズムの決定と、関連の設計、データのアクセスパスの最適化
- 汎化階層の調整を含め、オブジェクト表現の決定

以下では、これらの過程を経て得られるドキュメント間での一貫性条件を列挙する。対象となるドキュメントは次のものである。

- オブジェクトモデル
- 操作仕様書
- メッセージ階層図
- イベントトレース図

- オブジェクトインタラクション図[4]
- 分析モデルのドキュメント

3 モデルの統合に関して

動的モデルの動作、活動、機能モデルのプロセスはオブジェクトモデルに操作として反映される。これに対するチェック条件：

1. 事象に対応している操作は送信側オブジェクトが子孫クラスであれば、“protected”, さもなければ“public” でなければならない
2. 機能モデル中の最下層のプロセスはオブジェクトモデル中の操作に対応する
3. 分析フェーズのオブジェクトモデルが追加されている操作、削除された操作について確認を求める

4 操作の実装に関して

分析フェーズで作成したユースケース（シナリオ）をもとに、メッセージ階層図、イベントトレース図を作成し、そのアルゴリズムを操作仕様書に疑似コードとして反映する。これに対するチェック条件：

1. 疑似コード中で呼び出しているメソッドはその操作が属するクラスか、または public 指定されている他のクラスのメソッドである。さもなければ、子孫クラスの protected 指定されたメソッドである
2. 上記の public 指定されているクラスのメソッドが属するクラスとは関連が定義されていること
3. メッセージ階層図中のクラス名、メソッド名とオブジェクトモデルの間に矛盾がないこと
4. クライアントクラスからサブライアクラスへの向きと関連の向きが一致していること
5. 役割名とメソッドの使い方、使われ方が一致している
6. メッセージ階層図中の入れ子の内側のメソッドはその外側のメソッドの操作仕様書の疑似コード中に記述されていること
7. メッセージ階層図中で FOR ループによる処理がなされている関連には“多”の多重度が定義されていること
8. メッセージ階層図中のクラスとイベントトレース図中のクラスに矛盾がないこと

Consistency Check for Object Design in OMT

Tadashi KANEDA[†], Eri MATSUURA^{††}, Yuzuru FUJIWARA^{†††},
Kiminori UTSUNOMIYA^{†††}

[†]Doctoral Degree Program in Science and Engineering, Univ. of Tsukuba

^{††}College of Information Science, Univ. of Tsukuba

^{†††}Institute of Information Sciences and Electronics, Univ. of Tsukuba

9. メッセージ階層図中のメソッドの呼び出しがイベントトレース図中にイベントとして記述されていること
10. メッセージ階層図中のメソッドのクライアント-サブライア関係とイベントトレース図中の矢印の方向が一致していること、self呼び出しも同様
11. オブジェクトインタラクション図中のインタラクションとオブジェクトモデル中の関連は矛盾しないこと
12. オブジェクトインタラクション図中のインタラクションに記述されている呼び出し先のメソッドは呼び出し先のクラスに定義されている
13. オブジェクトインタラクション図中のインタラクションに付随する引数はそのメソッドの入力として操作仕様書中に定義されていること
14. オブジェクトインタラクション図のインタラクションの実行順序がメッセージ階層図、イベントトレース図と矛盾しないこと
15. オブジェクトモデル中のメソッドの引数と操作仕様書中の入力に矛盾がないこと
16. オブジェクトモデル中のメソッドの返り値指定と、その操作仕様書の出力が一致していること

5 オブジェクト表現などに関して

オブジェクトの実装で用いるデータ構造を決定する。同時に、抽象クラス、委譲を導入し、クラスの汎化階層の調整を行なう。汎化は、制限による汎化と拡張による汎化を明確に区別する。制限による汎化で継承される属性はR(Restrict)属性として指定する。これらのことなどに関するチェック項目：

1. あるクラスの属性が別のクラスのインスタンスである場合、その属性に対するアクセスを司るメソッドはインスタンスが属するクラスのメソッドからなる
2. 分析モデルにない集約はユーザの確認を求める
3. スーパークラスの属性のうち、サブクラスで使用していないものがあれば、ユーザの確認を求める
4. 多重継承は独立した汎化階層からの継承、もしくは同じ汎化階層からの互いに素でない汎化クラスからの継承である
5. 制限クラスにおける、R属性を更新するような操作の継承は禁止する
6. 抽象操作は必ず、サブクラスで再定義されていること
7. 汎化階層の最下層のクラスでは、抽象操作の指定がないこと
8. サブクラスの操作の再定義はスーパークラスの操作のシグニチャと矛盾がないこと
9. 集約関係と委譲の記述に矛盾がないこと
10. ある抽象クラスのサブクラスすべてに対する関連をもつクラスはその抽象クラスへの関連に統一する
11. 分析フェーズと設計フェーズのオブジェクトモデルでモジュール分割が矛盾しないこと
12. オブジェクトモデル中の多重度の白丸、黒丸の指定は、設計フェーズ中では省略してもかまわない

6 おわりに

分析フェーズの一貫性チェック [2] は試験的に実装し、一部稼働している。ここでは、WSN[5] に沿った形式言語を用いてデータ構造と機能を記述している。本報告の設計フェーズの一貫性チェックについても同様の形式言語で記述した後、それを人手でコードに変換することにより、実装することになっている。次の段階では、できるだけ多くの部分をZ++[6] で記述し、一貫性チェックを行うことを検討している。

参考文献

- [1] Rumbaugh, J. et al.: *Object-Oriented Modeling and Design*, Prentice Hall, 1991.
- [2] 波瀲由布子, 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.
- [3] Kurt, K.D.: *Applying OMT*, SIGS Books, 1995.
- [4] Rumbaugh, J.: *Going with the Flow*, Journal of Object-Oriented Programming, 7(3), 1994
- [5] 宇都宮公訓, 廣瀬健, 深沢良彰他.: WSNによるスケジューラの詳細記述とその経験, Bulliten of Centre for Informatics Waseda University, Vol.5, Spring, pp.32-46, 1987.
- [6] Lano, K.: *Formal Object-Oriented Development*, Springer, 1995.