

クラス図変形規則集の編纂*

4 Z C - 8

権藤佐俊 吉田敦 磯田定宏†
豊橋技術科学大学 知識情報工学系†

1 はじめに

オブジェクト指向分析では、対象とする問題をモデル化し、クラス図を作成する。しかし、クラス図を完成させるには、何度もモデルの変形を繰り返す必要がある。そこで、初期モデルを作成したあとに適用される変形規則を抽出し、規則集を編纂する。これにより、初心者モデルの洗練支援ができると考えられる。

2 規則の一覧

参考文献 [1-11] より、19 個の変形規則を収集し、次の基準に従い分類した (表 1)。

必須規則 要求仕様を満たすために適用される規則。モデルの洗練はインクリメンタルに行うとする。従って、要求仕様を部分的に満たすモデルが存在するときに新たな要求仕様を加えた場合に必須規則が適用される。

最適化規則 モデルの最適化のために使われるが必ずしも適用しなくてよい規則。

本論文では表 1 に示す変形規則の中から*印をつけた三つの規則について説明する。

表 1: 規則一覧

	必須規則	最適化規則
継承		<ul style="list-style-type: none"> ・クラスの継承 ・サブクラスを列挙型属性に変更 ・委譲による継承の回避 ・集約による継承の回避
集合	<ul style="list-style-type: none"> ・群クラスの識別 (*) 	<ul style="list-style-type: none"> ・属性のインデックス化
関連	<ul style="list-style-type: none"> ・関連クラスの識別 	<ul style="list-style-type: none"> ・列挙型属性から関連に変換 ・関連クラスの変形 ・クラスター化
具現化	<ul style="list-style-type: none"> ・制御のクラス化 	<ul style="list-style-type: none"> ・アルゴリズムのクラス化 ・振る舞いのクラス化 (*)
属性	<ul style="list-style-type: none"> ・属性のクラス化 	<ul style="list-style-type: none"> ・クラスの属性化
その他	<ul style="list-style-type: none"> ・履歴の保存 (*) ・ユースケース制御パターン 	<ul style="list-style-type: none"> ・クラスの分割 ・スルーパス制御オブジェクトの削除

3 変形規則

各変形規則は 5 つの項目について記述されている。

分類 変形規則の分類を記述する。

*Compiling a rulebook of class diagram transformations
†Satoshi GONDO, Atsushi YOSHIDA, Sadahiro ISODA

†Department of Knowledge-based Information Engineering, Toyohashi University of Technology

適用条件 変形規則が適用できる条件を記述する。

変形方法 クラスの変形方法を記述する。

効果 変形規則の適用による利点・効果を記述する。

適用例 変形規則を適用した具体例を記述する。

3.1 群クラスの識別

3.1.1 分類

必須規則

3.1.2 適用条件

クラス A がオブジェクトを複数持ち、要求仕様からクラス A のオブジェクト全体に対する操作 M が見つかった場合 (図 1(a))。

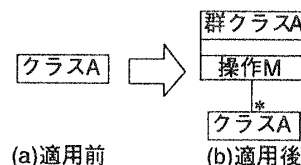


図 1: 群クラスの識別

3.1.3 変形方法

操作を割り当てるためにクラス A のオブジェクト全体を代表するクラスが新たに必要となる。図 1(a) から図 1(b) のように群クラス A を新たに作成し、群クラス A にオブジェクトの集合に対する操作 M を割り当てる。

3.1.4 効果

オブジェクトの集合に対する操作を適切にモデル化できる。

3.1.5 適用例

図 2(a) の「売上」クラスは「販売個数」と「日付」属性によってある商品が特定の日に何個売れたかを記録している。新たな要求仕様として、当日の売上が最大の商品と販売個数が最大の商品を調べる操作が追加されたとする。

この仕様を実現するには「最大売上商品を返す」と「最大販売個数商品を返す」という操作を追加する必要がある。しかし、これらの操作はオブジェクト群に対するものであるため、「売上」クラスの操作にすることはできない。よって、「売上」オブジェクトの全体を代表するクラスが必要になるので、図 2(b) のように「売上群」クラスを作成し、これらの操作を割り当てる。

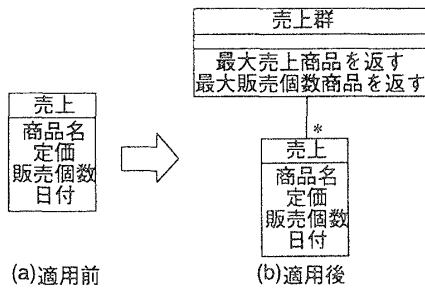


図 2: 群クラスの適用例

3.2 振る舞いのクラス化

3.2.1 分類

最適化規則

3.2.2 適用条件

クラス A の操作 M が次の条件のどれかに当てはまる場合。

- 操作自身が独自のデータ D を持つ。
- 複数のクラスが同一あるいは類似した操作を持つ。
- 操作が複雑である。

3.2.3 変形方法

操作 M を具現化しクラス B を識別する。操作 M をクラス B の操作にし、データ D をクラス B の属性にする。クラス A とクラス B の間に関連を作成し委譲により操作 M を利用する。

3.2.4 効果

- 処理の重複を除去できる。
- 振る舞いの再利用が可能になる。

3.2.5 適用例

図 3(a) の「文書」クラスと「概要」クラスはそれぞれ文書と概要の文章内容を持つ。また、文章内容から指定された単語を探す「単語検索」の操作を持つ。この操作は検索する対象のみが異なり検索処理の役割は同じだとする。このとき図 3(b) のように検索操作をクラス化し、委譲により利用する。

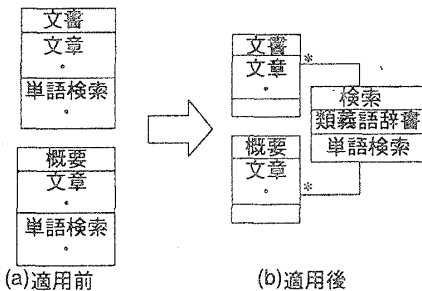


図 3: 振る舞いのクラス化の適用例

3.3 履歴の保存

3.3.1 分類

必須規則

3.3.2 適用条件

オブジェクト間の相互作用により(関連の)リンクが作成され、この相互作用を履歴として残したい場合。

3.3.3 変形方法

関連を関連クラスに変形し、関連が持っていた情報を関連クラスの属性とする。

3.3.4 効果

処理の履歴の保存や undo 処理が可能になる。

3.3.5 適用例

図 4(a) は図書館システムのクラス図の一部で図書と利用者には「借りている」という関連がある。このクラス図では現在ある利用者がある図書を借りているということしか表現できない。過去の貸し出しの履歴を残したい場合、図 4(b) のように関連から関連クラスに変形し「貸し出し」クラスに貸し出しの情報を保持させる。

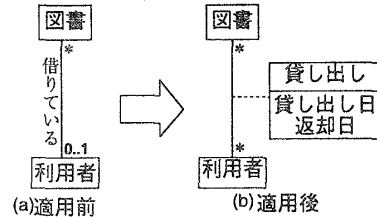


図 4: 履歴の保存の適用例

4 まとめ

文献から変形規則集を編纂した。これを用いることにより

- 初心者の学習の助けとなる。
- 分析者間で知識や経験を共有でき、意志の疎通が容易になる。

などの利点があると考えられる。

今後は既存の分析事例での各変形規則の使用頻度および分析時間の短縮効果を測定し、モデル化支援に対する効果を評価する。

参考文献

- [1] J.Rumbaugh, "Trouble with twins: Warning signs of mixed-up classes", JOOP, 7(4), pp.16-21, July/Aug. 1994.
- [2] J.Rumbaugh, "An object or not object?", JOOP, 5(3), pp.20-25, June 1992.
- [3] J.Rumbaugh, "The life of an object model", JOOP, 7(1), pp.24-32, Mar./Apr. 1994.
- [4] J.Rumbaugh, "Driving to a solution: Reification and the art of system design", JOOP, 8(4), pp.8-13, July/Aug. 1995.
- [5] J.Rumbaugh, "Let there be objects: a short guide to reification", JOOP, 5(7), pp.9-14, Nov./Dec. 1992.
- [6] J.Rumbaugh, "Disinherited! Examples of misuse of inheritance", JOOP, 5(9), pp.22-24, Feb. 1993.
- [7] J.Rumbaugh, "On the horns of the modeling dilemma: Choosing among alternate modeling constructs", JOOP, 6(7), pp.8-17, Nov./Dec. 1993.
- [8] 磯田定宏, "オブジェクト指向モデリング", コロナ社, 1998.
- [9] I.Jacobson, "オブジェクト指向ソフトウェア工学 OOSE", 監訳:西岡利博, 渡辺克宏, 梶原清彦, トッパン, 1995.
- [10] Craig Larman, "実践 UML パターンによるオブジェクト指向開発ガイド", 監訳:今野睦, 依田智夫, プレンティスホール出版, 1998.
- [11] 西澤和正, "オブジェクトクラスタに基づく最適な制御構造の作成規則", 情報処理学会第 59 回全国大会, 4ZC-6, 1999.