

粗粒度なパターン指向リファクタリングの考案*

1Q-04

青柳 哲 伊藤友昭 中井戸健至 木村 耕
電気通信大学 情報工学科

1 はじめに

近年、ソフトウェアの外部的振る舞いを維持しつつ、理解や修正が容易になるように内部構造を変化させるリファクタリングが注目を集めている [1].

一方、オブジェクト指向に基づくソフトウェア開発においてソフトウェアの再利用を促進させるアプローチとしてデザインパターン(以下 DP)が普及しつつある[2]. DP の適用には、ソフトウェアの設計段階で適用する手法の他に既存のソフトウェアに必要に応じて適用する手法が考案されている[3]. このような場合、開発者にリファクタリングと DP の双方の知識と経験が必要となる。そこで、本研究では既存のソフトウェアに対する DP 適用のコード変更の手順を粗粒度のパターンリファクタリング(以下 PR)として定め、さらにそれを用いた支援ツールを提案する。

2 パターンリファクタリング

PR は次の方針に基づく。

- DP の目的などから典型的な DP 適用前の初期状態を定める。
- 初期状態から DP 適用後の目標状態までのコード変更手順を定める。
- 変更手順をクラスの追加や削除といった細粒度のリファクタリング(基本リファクタリングと呼ぶ)で表す。

PR は表 1 の項目により構成される。

開発者は次の手順で DP の適用を行う。

- 1, 対応 DP, 目的, 初期状態から適用する PR を選ぶ。
- 2, ソフトウェアの構造と初期状態のクラス図とを比較して引数を決定する。
- 3, 実行手順に引数を適用し、コードの変更手順を作成する。
- 4, サンプルコードと目標状態を参考にして、変更手順に沿ってコードを変更する。

PR の例として、「Add Proxy」を以下に示す。

表 1 パターンリファクタリングの構成

<u>パターンリファクタリング名</u>	PR を識別するための名前
<u>対応デザインパターン</u>	適用されるデザインパターン名
<u>目的</u>	PR により改善されるソフトウェアの問題点
<u>初期状態</u>	PR 適用前のソフトウェア構造
<u>目標状態</u>	PR 適用後のソフトウェア構造
<u>引数</u>	PR の実行に必要な適用箇所
<u>実行手順</u>	コード変更手順
<u>サンプルコード</u>	Java 言語で記述されたサンプルコード
<u>備考</u>	適用に必要な条件や、注意事項など

パターンリファクタリング名 : Add Proxy

対応デザインパターン : Proxy

目的 : 生存にコストのかかるクラスや、初期化に時間のかかるクラスに代理クラスを立ててコストの低減や初期化時間の分散を図る。

初期状態 : (図 1 参照)

Subject…実クラス

Request1,2,……Subject の操作

Attribute1,2,……Subject の属性

目標状態 : (図 1 参照)

Subject…初期状態と同一

Proxy…代理クラス

realSubject…Proxy が持つ Subject 型の属性

AbstractSubject…Subject と Proxy のスーパークラス

引数 :

Subject, Proxy, AbstractSubject, realSubject

実行手順 : 表 2

サンプルコード : 省略

備考 : スコープ内に Proxy, AbstractSubject と同じクラスの名前がないとする。

*An idea of coarse grain pattern oriented refactoring by AOYAGI Satoshi, ITOH Tomoaki, NAKAIDO Takeshi and KIMURA Koh, Department of Computer, The University of Electro-Communications.

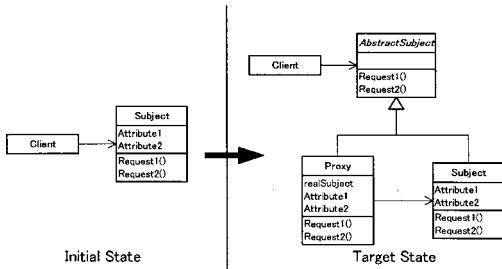


図 1 「Add Proxy」初期状態, 目標状態

表 2 「Add Proxy」実行手順¹

1. クラスの作成で AbstractSubject と Proxy を作成.
2. 継承関係の確立で Subject・Proxy 間および AbstractSubject・Subject 間に継承関係を確立.
3. 変数の追加で Proxy に変数 realSubject, Attribute1, Attribute2 を作成.
4. メソッドアクセサの作成で Subject の各メソッドを実行するメソッド Request1, Request2 を Proxy に追加する.
5. 抽象メソッドの宣言で AbstractSubject に抽象メソッド Request1, Request2 を宣言.
6. 関連先のスーパークラスへの置換で Client クラスの参照先を Subject から AbstractSubject へ変更.

3 支援ツール

本ツールは PR を用いて、対象コードのクラス図と DP から、必要な変更手順を出力しリファクタリングの支援を行う。

図 2 は本ツールのフローを表し、①～③において次の動作を行う。

- ① ソフトウェア構造、適用する DP の種類および適用箇所の情報を取得する。
- ② PR データベースを参照し、該当する PR を決定する。
- ③ 基本リファクタリングで構成される変更手順を出力する。

本ツールをモデリングツールや DP 選択ツール、リファクタリング適用支援ツールと組み合わせることで、DP の選択からコードの変更までを自動化できると考えられる。

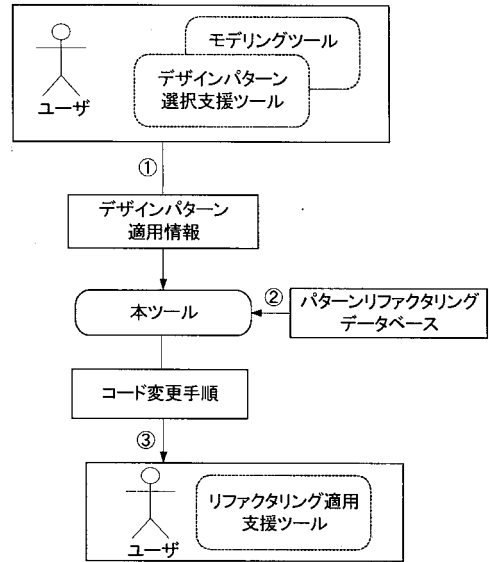


図 2 支援ツールのフロー

4 おわりに

PR と対応する支援ツールを用いることで、既存のソフトウェアに対する DP の適用における能率が向上し、またツールによる自動化が可能になるものと考えられる。

現在、本ツールの構想に基づくプロトタイプが稼動中である。今後は、より実用的なソフトウェアに対する評価を行うとともに、発表者らがかつて発表した DP 選択ツール[4]や、リファクタリング適用支援ツール[5]との統合を考えている。

参考文献

- [1] マーチン ファウラー 著, 児玉公信, 友野晶夫, 平澤章, 梅澤真史 訳: “リファクタリング: プログラミングの体質改善テクニック,” ピアソン・エデュケーション, 2000.
- [2] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides 著, 本位田真一, 吉田和樹 監訳: “オブジェクト指向における再利用のためのデザインパターン 改訂版,” ソフトバンクパブリッシング, 1999.
- [3] Lance Tokuda and Don Batory: “Automating Three Modes of Evolution for Object-Oriented Software Architectures,” In Proceedings of COOTS'99 (1999).
- [4] 山腰哲, 青柳哲, 木村耕: “エージェントを用いたパターン指向リファクタリングツールの考案,” 情報処理学会第 60 回全国大会講演論文集 (分冊 1), pp.339-340 (2000).
- [5] 青柳哲, 中塚房男, 木村耕: “クラスエージェントを用いたリファクタリングシステムの考案,” 情報処理学会第 62 回全国大会講演論文集 (分冊 1), pp.277-278 (2001).

¹ 斜体は基本リファクタリング名を表す。