平成19年度トップエスイー修了制作

Freediaフレームワークを用いた パターン指向リファクタリングのケーススタディ

早稲田大学　河村美嗣　yositugu@fuka.info.waseda.ac.jp

開発における問題点

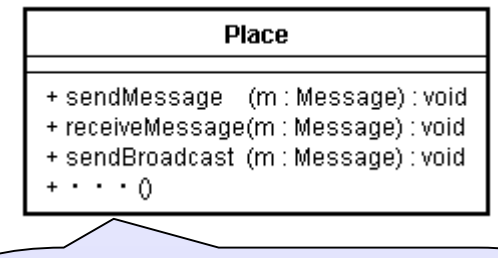
Freediaフレームワークとは、Smartiveプロジェクトにおいて国立情報学研究所本位田研究室を中心として開発されたエージェントフレームワークである。研究機関における開発では、最先端の機能をいち早く実装することを重要視し、要求が時代とアイデア次第で増加する。このため、機能を逐次追加することになり最初の設計が不適切になった。

手法・ツールの適用による解決

既存の設計を改善するため、ソフトウェアパターンの適用を行うリファクタリングであるパターン指向リファクタリングを行う。今回のケースでは、Strategyパターンを用いたリファクタリングを行った。
（Strategyパターン・・・アルゴリズムをオブジェクト化し、実行時にアルゴリズムを選択することができるようにするパターン）

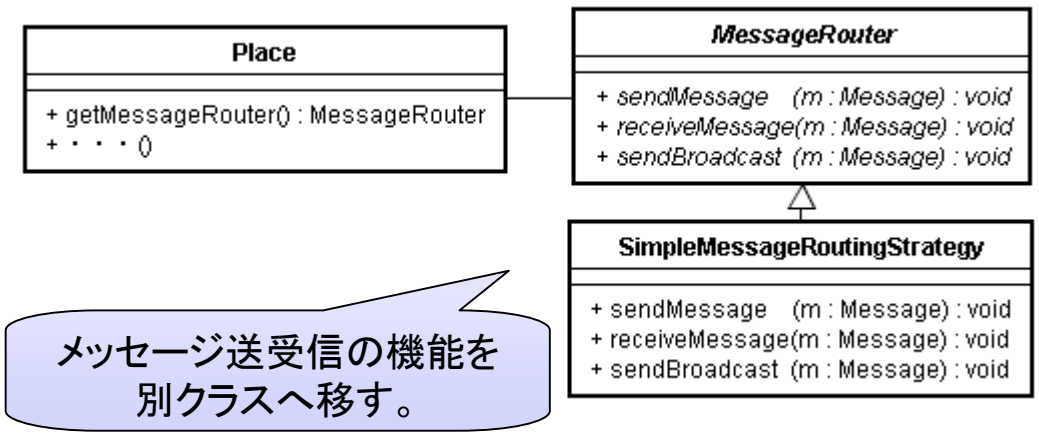
クラス図

リファクタリング前



自由な発想による研究に応じ機能追加、当初想定していない肥大化

リファクタリング後



メッセージ送受信の機能を別クラスへ移す。

評価

評価指標	Place (リファクタリング前)	Place (リファクタリング後)
メソッド数	122	106
属性数	19	15
メソッドの凝集性欠如度	0.923	0.912
メソッド複雑度の合計	307	249

Placeクラスは、大きく分けて4つの機能を持っていたため、そのうちの1機能をStrategyパターンを用いて別クラスに分離した。

その結果、メソッド数を16、属性数を4、クラス内メソッド複雑度の合計を58減らすことが出来た。

しかし、メソッドの凝集性欠如度は依然として高いため、さらにリファクタリングを行う余地があることが分かる。

まとめ

要求が明確でない、後からの仕様変更が多いプロジェクトの場合、初期段階に行った設計が不適切になる可能性が高い。このようなプロジェクトでは、後からリファクタリングによる再設計が必要になる。

今回のケースでは、リファクタリング対象にそれほど詳しくなく、リファクタリングのプロでもない者が、後からパターンという経験知を再利用することにより短期間に効果的なリファクタリングを行った。

参考文献:
ジョシュア・ケリーエブスキー (2006)
「パターン指向リファクタリング入門~ソフトウエア設計を改善する27の作法」日経BP pp381

国立情報学研究所 National Institute of Informatics
トップエスイー　～サイエンスによる知的ものづくり教育プログラム～
文部科学省 振興調整費産学融合先端ソフトウェア技術者養成拠点の形成