

Title	レガシーシステムから機能追加・修正が容易なシステムへの再構成
Author(s)	高橋, 悠
Citation	
Issue Date	2009-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/8134
Rights	
Description	Supervisor:鈴木正人, 情報科学研究科, 修士

レガシーシステムから 機能追加・修正が容易なシステムへの再構成

高橋 悠 (710042)

北陸先端科学技術大学院大学 情報科学研究科

2009 年 2 月 5 日

キーワード: ソフトウェア保守, 再構成支援.

1 背景と目的

現在のシステム開発の多くは新規に構築を行うことが少なく, 既存システムに追加・拡張を行うことで実現されることが多い. システムを再利用することで, 開発コストを低減させるためである. しかし, 再利用を繰り返すことによりソースコード量は肥大化し, システムの構造も複雑になる.

そのようなシステムの保守作業は, 保守コストの増大を招く. コストが肥大化する要因は, ソースコードが理解しづらいことと, 変更の適用手法が不明であることが挙げられる. これらの問題に対し, 有効な解決策はいまだ見つかっておらず, アドホックに対応している現状がある. 問題の解決策として, 意味を踏まえた理解支援や変更手法が必要である. ここで, 意味を踏まえた変更手法とは, システムの追加・修正などの要求に対して修正前よりも変更箇所が少なくなるようにシステムを変更する手法のことである.

本研究の目的は, 保守担当者が既存の複雑なソースコードを, 理解しやすく保守作業を効率的に行えるように再構成する手法を提案することである. また, 他のレガシーシステムもシステムの再利用をしやすいうに再構成することのできる手法を構築することが目的である.

2 保守の現状と問題点

システムの再構成技術の既存技術として, リファクタリングと大尾の提唱する研究がある. リファクタリングとは, プログラムの外部から見た振る舞いを変更せずにプログラム内部を変更する技術である. 複雑な構造をしたプログラムを, 振る舞いを変えずに整えることができる. しかし, リファクタリングで行える再構成は, 意味を変更するものではな

い。また、再構成を行う際に適用範囲を考える必要が生じる。保守担当者が未熟な場合、どこから着手すればよいのかすら分からないこともある。大尾の研究は、設計を対象に意味のある再構成手法を提案した。しかし、開発現場では、設計書が残されていることが少なく、手法を利用をする機会が少なくなっている。このため、意味を有したソースコードを対象にした再構成手法の確立が望まれている。

3 レガシーシステムの構造と再構成方針

レガシーシステムとは、構成が古いために新規サービスの導入が難しかったり、機能拡張が頻繁に行われたために構造自体が複雑になったりしているシステムのことを指す。

本研究では、レガシーシステムの中でも特に Java で作成された Web アプリケーションを対象とする。Java は言語仕様の追加や変更が短期間に頻出するため、以前のシステムの再利用や修正が困難になることが多い。また、初期の Web アプリケーション開発では、標準的な構成が定まっておらず、アドホックに開発が進められていた。

レガシーシステムでは、ひとつのオブジェクトに複数の役割が混在していたり、実行時に利用されないオブジェクトが存在することが多い。そのようなオブジェクトでは、記述の修正箇所が複数に散在していたり、不要なオブジェクトまで検討を行わなければならない。保守のコストが増大する要因である。

再構成の方針として、Web システムの標準的構成である 3 階層アーキテクチャを用いることにした。3 階層アーキテクチャは、Web ページの作成を行うプレゼンテーション層、データの加工を行うビジネスロジック層、データを保存するデータベース層で構成される。また、システムの構成を把握するために、フロー解析を行う。

4 再構成手法の提案

レガシーシステムを解析することで再構成手法を考案する。提案手法は 2 つの手順により構成される。

- 層の分離
- 有用なオブジェクトの抽出

レガシーシステムでは、ひとつのオブジェクトに複数の役割が混在しているという問題があった。3 階層アーキテクチャをもちいて層の分割を行うことで、役割ごとにオブジェクトを分類することができる。複雑化しているオブジェクトからプレゼンテーション層やデータベース層に移行すべき記述を見つけ、それらを別メソッドにする。そのメソッドを適したクラスに移行することにより層の分割が行われる。

また、レガシーシステムに追加・修正が繰り返されることにより、実行時に利用されないオブジェクトが多く含まれているという問題がある。フロー解析を行い有用なオブジェ

クトのみ抽出することで、利用されないオブジェクトを削除することができる。ユーザが起動するページから解析を行い、そのページが作成するオブジェクトを抽出することで実行時に有用なオブジェクトを検出することができる。

5 適用実験と評価

電子商取引システムを用いて適用実験を行った。再構成の効果を確認するため、層の分離では、あるオブジェクトにデータを増やす実験を行った。再構成を行う前のシステムでは、変更箇所が分散しており一貫性を保つのが困難だったのに対し、再構成後のシステムでは、データ構造が一括化されており、一貫した修正を行うことができた。

有用なオブジェクトの抽出手法では、電子商取引システムに対してフロー解析を行った。その結果、3階層アーキテクチャの各層ごとにフローを整理し、層をまたがる交信を低減した。

6 まとめ

本研究では、ソースコードを対象とするシステムの再構成手法を提案した。層の分割手法を適用することで、オブジェクト内の機能がひとつにまとまり、機能の追加・修正を容易に行うことができる。有用なオブジェクトの抽出手法を適用することで、実行時にシステムで使われているオブジェクトのみ抽出することができる。