

発表概要

識別子命名からのオブジェクトの機能分析に基づく プログラム構造変化の類型抽出

山下 大貴^{1,a)} 竹内 和広²

2015年11月6日発表

本研究では、バージョンアップにともない、ソフトウェアが修正・拡張され、頑健性が向上する様子を分析する。特に、ソフトウェアの理解や保守を向上させるといった様々な利点があるとされているデザインパターンの実装を主眼に分析を行う。従来のデザインパターン分析ツールでは、オブジェクトの構成を主眼に、プログラムが採用するデザインパターンを分析する。それに対して、本研究では、デザインパターンの内部オブジェクトの機能形成をデータ構造の観点から分析する。その際、オブジェクトの機能の特定は、オブジェクトとそれに関係するメソッドの識別子命名に用いられる自然言語の用法を分析用辞書に整理することにより実現する。また、分析用辞書は、リポジトリマイニングを活用して構築したものをを用いる。研究の成果については、まず、提案の分析手法が、従来のデザインパターン特定ツールに比べて、精度的に劣らず、時系列分析に適している点を示す。次に、複数のオープンソースプロジェクトのマイナーバージョンアップにおける構造変化の分析例として、当初は list データ構造のみのプログラム構造から iterator パターンに近いプログラム構造に変化することにより、ソフトウェアの頑健性を向上させているといった、いくつかの構造変化の類型の発見について説明する。このような構造変化の類型の発見は、デザインパターンの実装が段階的に頑健となっていく方法論を知るうえで有益な知見と考える。

Extracting Temporal Patterns in Structural Revisions of Program Based on Functional Analysis of Objects by Word Usage of Their Identifiers

HIROKI YAMASHITA^{1,a)} KAZUHIRO TAKEUCHI²

Presented: November 6, 2015

In this paper, we analyze structural changes of software through its history of upgrades. Our viewpoint of the analysis is temporal features of implementation of the design patterns, which make the maintenance of software easier and improve its readability. For this purpose, we propose a tool that analyzes software by the functions of objects that constitute design patterns under our assumption that these functions can be analyzed by the naming of identifiers for objects and methods in the program. As a basis of the analysis, we conduct the data mining to software repositories and collect usages and rules of natural language words for naming identifiers. Contributions of our work to program engineering are twofold: we firstly show the capabilities of our proposing tool that can identify function of the objects by comparing with one of the well-known tools that have been used for the design pattern analysis to existing programs. Secondly, we show our findings that there are some temporal patterns through software upgrades: for example, a simple implementation of data structure sometimes grows up as more complex one where the data structure is a part of the iterator pattern. These findings will contribute to make it clearer how the robustness of implementation of design patterns is improved through the revisions.

¹ 大阪電気通信大学大学院工学研究科
Graduate School of Engineering, Osaka Electro-Communication University, Neyagawa, Osaka 572-8530, Japan

² 大阪電気通信大学情報通信工学部
Faculty of Information and Communication, Osaka Electro-Communication University, Neyagawa, Osaka 572-8530, Japan

a) mi14a009@oecu.jp