

Title	JAVAソースコードにおける協調クラス群の抽出
Author(s)	グエン ヴァン, トゥアン
Citation	
Issue Date	2010-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/8953
Rights	
Description	Supervisor:落水 浩一郎, 情報科学研究科, 修士

JAVA ソースコードにおける協調クラス群の抽出

NGUYEN VAN TUAN(0810021)

北陸先端科学技術大学院大学 情報科学研究科

2010 年 2 月 9 日

キーワード: Java, 協調クラス群, デザインパターン, メタパターン, UML, クラス図.

1 背景、目的

情報システムを開発する際に、膨大な量のソフトウェア図面とプログラムが存在している。また、このソフトウェア図面やプログラムの中に複雑な依存関係があるために、変更作業が困難である。変更に必要な労力の軽減、信頼性の向上を保証するために、変更作業を自動化して支援するツールが必要である。また、変更作業を自動化して支援するツールにおいては、設計モデルの要素に対応するソースコードにある協調クラス群を自動的に発見することが大切である。

本研究の目的はユースケースを実現する（クラス図中のクラス群に対応する）Java クラス群を協調クラス群と定義し、それを抽出することである。

2 関連研究

2006年に金旭東はメタパターンを用いたクラス群を協調クラス群と定義し、Preceの6種のメタパターンを3つの協調構造に分類して、各協調構造の特徴による協調クラス群の抽出アルゴリズムを開発した。GoFの23種のデザインパターンを対象として、抽出アルゴリズムを適用し、17種のデザインパターンを利用したクラス群を抽出する可能があった。抽出できない6種のデザインパターンの中に、3種のデザインパターンはメタパターンで説明できなくて、3種のデザインパターンはメタパターンで説明できる。

3 研究方法

協調クラス群を抽出するために、我々は先行研究を改良して、ユースケースに含まれるJavaクラス群を抽出する。このことを達するために、二つの課題がある。

1. メタパターンで説明できないデザインパターンを解析可能にする。具体的には、メタパターンの構造的特徴を利用すること以外に、新しく「構造の特徴」と「振る舞いの特徴」を導入して、メタパターンを適用するクラス群を抽出する。
2. ユースケースを実装したクラス図に対応する Java クラス群を抽出する。

課題1については、クラス図を実装する際に、デザインパターンを使って実装することが多いため、協調クラス群を抽出するためにデザインパターンを利用したクラス群を抽出することが大切である。また、デザインパターンの大部分が Pree のメタパターンで説明できるため、金旭東の開発したメタパターンを用いたクラス群を抽出するアルゴリズムの失敗した原因を調査し、すべての Pree のメタパターンに対応できるように再開発した。さらに、メタパターンで説明できないデザインパターンを利用したクラス群を抽出するために、デザインパターンの構造と振る舞いの特徴をまとめて、抽出アルゴリズムを開発した。

課題2については、デザインパターンを適用した箇所を抽出するアルゴリズムを利用して、Java ソースコードにあるデザインパターンを利用したクラス群を抽出する。抽出したクラス群を利用して、デザインパターンの構造的特徴に基づいてユースケースを実装したクラス図の一つひとつのクラスに対して対応をする Java クラス群を抽出する。その後、サブグラフ同型判定アルゴリズムを適用して対応をつける手法を提案した。また、デザインパターンを使わないで実装する場合があるため、抽出するクラス図のクラスに対応する Java クラス群が違う場合がある。そのため、我々は違いを修正するためにデザインパターンを利用していないソースコードの追跡規則を提案した。

4 実験

この抽出方法を利用して、各実験を行い、Patterns in Java という本に載っている 41 種のデザインパターンの中 39 種のデザインパターンに対応する可能性がある。小規模なエレベータ制御システムと ATM システムで実験した結果により、抽出アルゴリズムの精度は 92.5% 以上であった。

5 まとめと今後の課題

本研究は、金旭東の提案したアルゴリズムを改善して、メタパターンの特徴の以外に構造と振る舞いの特徴を導入して、Java ソースコードにおけるデザインパターンを利用したクラス群を抽出する方法を提案した。デザインパターンを利用したクラス群を抽出するアルゴリズムを利用して、クラス図の要素に対応する Java クラス群を発見する方法を提案した。ユースケースのクラス図の要素に代わって対応するクラス群を入れることにより、ユースケースに含まれるクラス群を抽出した。

今後の課題としては、実験で失敗した事例への対応、より多いデザインパターンを利用したソースコードへの対応と Java 以外のオブジェクト指向プログラミング言語への対応が挙げられる。