

Title	メタパターンを用いたJavaソースコードにおける協調クラス群の抽出に関する研究
Author(s)	金, 旭東
Citation	
Issue Date	2006-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1990
Rights	
Description	Supervisor:落水 浩一郎, 情報科学研究科, 修士

メタパターンを用いた Java ソースコードにおける 協調クラス群の抽出に関する研究

金 旭東 (410038)

北陸先端科学技術大学院大学 情報科学研究科

2006 年 2 月 10 日

キーワード: メタパターン, テンプレートメソッド, フックメソッド, 協調クラス群, Java
ソースコード.

1 本研究の背景と目的

ソフトウェア共同開発においては,すでに作成した成果物を参照しつつ,新しい成果物が生成されるため,成果物間にさまざまな種類の依存関係が発生し,依存関係の数も多くなる.これによって開発時の修正作業やバグの発生に伴って生じる変更作業が困難になる.さらに,通常,成果物間の依存関係は設計者やプログラマ自身により定義されるが,試行錯誤を伴う設計活動ではその管理は容易ではない.

落水研究室では,すでに作成した成果物に対して,このような依存関係を自動生成することで,変更に必要なコストの軽減を図るための研究を進めている.具体的には,UML および Java 言語を対象に,依存関係を自動抽出し,それを利用して変更作業支援のワークフローを自動生成する研究を行っている.すでに UML 要素間に依存関係を自動生成する研究については一定の成果を得ている.

本研究の目的は,UML のモデル要素に対応する,Java ソースコードにおけるクラス群を抽出することである.本研究では,UML 図面中の要素の機能を実現するため実装された Java ソースコード中の関連クラスの固まりを協調クラス群と定義する.具体的には,Java クラス間の参照関係をメタパターンを用いて解析することにより協調クラス群を抽出する方法を提案する.参照には協調に関与するものとそうでないものがある.協調関係の判定には,メタパターンの構成要素であるテンプレートメソッドとフックメソッドの関係を用いる.テンプレートメソッドは,具象メソッド,抽象メソッド,フックメソッドの任意の組み合わせを呼び出すメソッドで,その詳細を実装することなくアルゴリズム輪郭だけを決定するメソッドである.フックメソッドは,サブクラスでオーバーライトされることを想定したメソッドである.テンプレートメソッドとフックメソッドとの協調の型を定義し,それを利用して協調クラス群を抽出するアルゴリズムを開発する.

2 問題点

UML のモデル要素に対応する,Java ソースコードにおける協調クラス群を抽出するためには,ソフトウェアの設計時構造と実装時構造の差異を考慮する必要がある. 一般に UML 図面中のあるモデル要素を実装した場合,必ずしも単一のクラスとはならない.多くの場合は一個以上のクラス群として実現される.

協調クラス群を抽出する手掛かりとしては継承,実装,参照の3種類がある.継承,実装を用いる場合は渡部による先行研究によって実現された.参照に関しては,エレベータ制御システム仕様に基づく開発されたソースコードから,参照には,協調に必要な参照と不必要な参照の切り分けの問題があることがわかった.

3 メタパターンによる協調クラス群の抽出

UML 図からどのような構造のソフトウェアを実装するかは,実装者の意図によって異なるものであり,協調クラス群となり得るすべての協調を定義することは不可能である.協調クラス群の例としてデザインパターンが考えられる.デザインパターンは様々な状況に適用できるテンプレートのようなものである.本研究では,すべてのデザインパターンで使われている協調クラス群の特徴を利用して協調クラス群を抽出するアルゴリズムを開発する.

Prece によれば,デザインパターンは7つのメタパターンに分類される.本研究では,さらに7つのメタパターンを,テンプレートメソッドとフックメソッドが協調する構造的な特徴により三つの代表的な型に分類した.この三つの代表的な型に対応する Java の言語的特徴を整理し,Java プログラム内の協調するクラス群を抽出する方針を取る.

4 協調関係の判定法

本節では,テンプレートメソッドとそれに対応するフックメソッドが存在するか否かにより協調関係を確認する方法について述べる.

1. テンプレートメソッドの探索

(1)Java ソースコード中の任意のあるクラスについて,まず,継承関係があるか否かをチェックする.継承関係がない場合はそのクラスのメソッドが同じクラスのメソッドを利用しているか,または,ほかのクラスのメソッドを利用しているかをチェックする.ある場合は: ステップ2でフックメソッドの探索を行う.ない場合は: 協調関係がないと判定する.

(2)継承によりグループになったクラス群に対して,グループの全てのクラスのメソッドに対して(1)のチェックを行う.

2. フックメソッドの探索

(1) テンプレートメソッドと同じクラス内のメソッドを利用した場合、利用されたメソッドがそのクラスのサブクラスにオーバーライトされていないかをチェックする。オーバーライトされていたら： 協調関係があると判定する。オーバーライトされていなかったら（1 - (2) の場合のみ）： 1 - (2) に戻り探索を続ける。

(2) ほかのクラスのメソッドを参照する場合：参照されたメソッドがオーバーライトされていないかをチェックする。そうであれば： 協調関係があると判定する。そうでなければ（1 - (2) の場合のみ）： 1 - (2) に戻り探索を続ける。

5 三つの協調構造から見たクラス間の協調関係

Pre によれば、メタパターンは7つあるが、本研究では、テンプレートクラスのオブジェクトがフッククラスのオブジェクトの多重度については、協調に關与する参照を判断する立場からは無関係なので注目しない。このような視点からすると、7つのメタパターンの構造的特徴を3つにまとめることができる。

1. 統合的協調構造：統合的協調構造においては、テンプレートメソッドとフックメソッドは、一つのクラスに存在する。
2. 結合的協調構造：結合的協調構造においては、テンプレートメソッドとフックメソッドが異なるクラスに存在する。
3. 再帰的結合協調構造：再帰的結合協調構造においては、テンプレートメソッドとフックメソッドが異なるクラスに存在し、テンプレートクラスとフッククラスの間にも再帰的關係がある。

6 適用例

本研究で開発したアルゴリズムの妥当性を確認するため、以下の3つの実験を行った。

1. GoF のデザインパターンに本アルゴリズムを適用した結果 23 個のデザインパターン中 20 個抽出できた。抽出できなかった Factory Method, Flyweight, Abstract Factory については、生成されるオブジェクト群を併せて解析するなどの必要があり今後の課題である。
2. デザインパターンを含む一般的な協調関係の抽出能力を確認するため北陸先端大の講義で使われているエレベータ制御システムの Java ソースコードに本アルゴリズムを解析した結果、一般的な協調クラス群の抽出にも適用できることがわかった。
3. UML モデル要素を実装するには、実装者の意図によって異なりますので、それに対応する協調クラス群の抽出も可能であることを確認できた。

7 まとめと今後の課題

本研究では,メタパターンを用いて協調に関連する参照のみを取り出すアルゴリズムを提案した.テンプレートメソッドとフックメソッド間の構造的関係を利用して有効な協調構造を抽出できることを確認した.

本研究の検討対象外となっている,3つのデザインパターン Facade,Singleton,Memento についての協調は今後の課題としてさらに,アルゴリズムの開発を行う必要がある.

すべての協調クラス群に対応できるアルゴリズムにするため,異なる協調構造も検討する必要がある.