Title	Java
Author(s)	,
Citation	
Issue Date	2006-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1990
Rights	
Description	Supervisor: , ,

Japan Advanced Institute of Science and Technology

Extraction of Collaborating Classes Using Meta Patterns

Xudong Jin (410038)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 9, 2006

Keywords: meta patterns, template methods, hook methods, extracts collaborating classes, Java..

1 Background and Purpose

In the cooperative software development, a developer creates an artifact referring other artifacts. There are various kinds of dependencies among them, such as refine and trace. When "B depends on A", here, dependency relation is relation to "have to impact B when you changed A". Therefore work to change documents and a source codes having complicated dependency relation adequately without exception is not easy.

In Ochimizu 's lab for an UML figure and a Java source code, development to help an effective work change by security is pushed forward. In details Extracting automatically such relationship among UML model element and Java language is utilized to the work flow change support research. And, we have made some progress in extracting relationship of UML automatically.

In this paper we propose the method that extracts collaborating classes by analyzing the references between classes using meta patterns. The references related to   some collaboration are decided by the structural relationships between template methods and hook methods which are important components of a meta pattern.

2　Problems

It is necessary to consider a difference of a structure between the design and the simulation to extract a group of classes form a Java source code corresponding to the UML modeling. Usually, the function is implemented by the combination of several classes, instead of single class. In this paper, these classes are called as collaborating classes.

There are three methods to　extract　collaboration classes such as inheriting, implementing and reference. The first, two methods have been achieved by Watanabe. For reference method, it is necessary to classify it into collaboration relation reference and collaboration unrelative reference.

3　Extraction of collaborating classes using Meta patterns

Because there are some difference between simulators writing software structures by UML modeling elements . So it is difficult to adapt for all kinds of design patterns in one way. In the paper, we extract collaborating classes by considering all kinds of design pattern's characteristic,　since design pattern is included collaborating classes.

In my research, I defined three types typical construction which formed based on Pree's seven kinds of meta patterns. I extract collaborating classes using this, three types of construct which used to analyze the characteristics of Java in order to extract the collaborating classes.

4　Deciding collaborating relationship

In this Chapter, the template method and hook method are used to confirm the existence of collabor　ating relationship, which are the basic testing method adopted in this research.

Searching for template Method

(1) checking whether there is a inheriting relationship between arbitrary classes in Java source code. When there is no inheriting relationship, each method of this class need to be checked whether

it has reference relationship with other methods in this class or in the other classes.

if there has this kind of reference relationship, then go to step .

otherwise: there is no collaborating relation.

(2) if there is inheriting relationship, the method of this classes also need be checked by the step -(1) .

Searching for hook Method

(1) When two methods in are the same class, it is necessary to check whether the method is overwritten by the descendant class or not.

If it is overwritten: It is judged that there is a collaboration relationship.

If it is not (Only for (2)): the procedure from return to(2) to search next method.

(2) When template methods refers to the method of another class: Checking whether the reference method is overwritten: It is judged that there is a collaboration relationship.

Otherwise (only for -(2)): the procedure from return to(2) to search next method.

5 Collaboration relationship between classes among three class-collaboration-structures

seven meta patterns into defined by Pree, which are simplified into three mate patterns based on the template class in our method, template class is just accounted as 1 time no matter how many times the references occurred.

(1) The integrated collaboration structure: In the integrated cooperation structure, the method of the template and the method of the hook exist in the same class.

(2) United collaboration structure: In united cooperation structure, the method of the template and the hook exist in different classes.

(3) The recurrent uniting cooperation structure: In the recurrent uniting collaboration structure the method of the template and the hook exist in different classes and there is a recurrent relation between the template class and the hook class.

6 example

In order to verify the correction of this algorithm, we conduct three experiments stated below.

(1) GoF design pattern were used in this experiment, in which 20 design patterns were successfully extracted out of 23 design patterns.

(2) In order to test the general nature of this algorithm, the Java source code of elevator control system used in JAIST course was used in the experiment. The result shows that this method has general nature.

(3) 3. In UML, the real structure of the software is determined by programmer's intention. Different intention can carry out different collaboration classes. This algorithm was verified in this situation corresponding to different collaboration classes.

7 Summary and future work

In this paper, we proposed the algorithm that extracts collaboration-related reference relationship of collaboration by using the meta pattern. which was confirmed to be able to effectively extract an collaborative structure by using the structural relations between the template method and the hook method.

In the future we should improve the new algorithm which consider the other design patterns of collaboration e.t. Facade, Singleton, and Memento. It is necessary to investigate if there is some other different collaborative structures which are benefit to extract all the collaboration classes.