Design and Implementation of a Class Browser based on the Object-Oriented Patterns

HAGIWARA Toyotaka

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

February 14, 1997

Keywords: Design Pattern, Object Oriented, Reverse Engineering.

1 Background and purpose

Recently, application examples of the object oriented technique has been increased. The expert software developer use object oriented technique aggressively in the various steps of software developments. Especially, for implementing large-scale software, they use object oriented programming languages.

In the implementation using object oriented programming languages, class libraries achieve important part. It is the important element that the set of classes contained in the software developed in the past, too. These sets of classes are reused when developing software newly. However, to reuse the sets of classes, the programmer must understand them. To understand them means to understand what design it has.

Generally, the designer designs the set classes to cooperate for realizing complex functions. In reusing the set of classes, by understanding the design of the set of classes, it is possible to use the complex functions realized by the cooperating classes, in addition to using the functions of each class. Hence, this paper proposes a support method for the programmer to understand the design of the set of cooperating classes.

2 Related work

The design pattern is used for the design of the set of cooperating classes. It is very useful to understand what design pattern is used, when the programmer try to understands the design of the set of classes contained in existing software.

Also, object oriented pattern Party is an useful general idea to extract the cooperating classes from a set of classes, especially by using a computer.

3 Our Approach

We adopt the Java language for the subject of our work. This is due to the following reason. Java is the strong typed language. In such languages, it is easy to collect information from the source code. Therefore, it is favorable for the information collection from the set of classes which is contained in the existence software. Also, it handles all elements as classes. Therefore, it is easy to concentrate on building the support mechanism about the set of classes, more than on the conglomerate program language like C++.

There are two design principles in the design pattern, which are shown below. As the 1st point, distinguish between the type of a object and the class of object. And as 2nd point, it is to use object composition to some types of object. The mechanism separating the class of object and the type of object shown as the 1st point is equipped in Java language. It is called the Interface.

As for the object oriented pattern Party, it isn't considering this separation mechanism. To handle the interface mechanism of Java, we extended object oriented pattern Party. By doing so, it became possible to handle a class and type distinction in Party. Also, by trace chains of Party, the programmer can acquire information about the object composition. This is effective to the use of the knowledge about which design patterns are used by the set of class in its design.

4 Design of the class browser

As the support mechanism to the programmer, we designed and implemented a class browser. This class browser can be browsing based on Party. By tracing chains of object oriented pattern Party, it is possible to express the class group to entertain, cooperating.

Browsing is made of class browser about the chain of this object oriented pattern Party. It corresponded to the interface mechanism of Java and it extended object oriented pattern Party. It applied class browser which was experimentally made to the class set to examine the effectivity. As a result, it was possible to easily detect the part which is using an interface mechanism. Moreover, it was possible to recognize the structure which is frequently used in the design pattern in browsing the chain of object oriented pattern Party around the interface mechanism. However, it was necessary to refer to the code making a class name a reference to specify an actually used design pattern and actually.

5 Application and evaluation

It evaluated experimental production browser than existing class browser. As a result, experimental production browser got the conclusion that it is excellent about the ability to support a programmer about the understanding of the class set which it entertains, cooperating compared with existing class browser.

There was one which displays a graph to the chain of the function calling, too, in existing class browser. However, those graphs display only the part of the class set to entertain with they cooperating only. Therefore, to understand the structure of the class set to entertain, cooperating, it is insufficient.

6 conclusion and future work

It got the following conclusion by the class browser implementation.

- It introduced the chain of object oriented pattern Party to class browser.As a result, the programmer could win the information which is necessary to understand in the class set to entertain, cooperating which it isn't possible to distinguish between about the acceding graph. Therefore, as for the class set to entertain, cooperating, the understanding support of the programmer is possible.
- It introduced the interface mechanism of the Java language into object oriented pattern Party. With the thing, the assistance to suppose the purpose, the motive of the designer became possible.
- The display of object oriented pattern Party with class browser is more effective from the point to support the understanding of a programmer about the class set to entertain, cooperating than the display of the calling graph with function.

As the problem in the future, it is possible to give following problem.

- We must change the way of Party of the language except Java of extending it for which it is possible to express the class of object and type of object distinction when supporting it.
- We are one while being in the design principle of the design pattern. It is necessary to contrive the means of handling "the transfer of the request by object composition" appropriately.