Reconfiguration in the system which can easily perform function addition and modification from the Legacy system.

Yuh Takahashi (710042)

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

February 5, 2009

Keywords: Software Maintenance, Reconfigration Support.

1 Introduction and Aim

It is rare that the system development is built newly now. And it is often that the developers add it to an existing system. Because the developers believe that the development expense is getting cheaper, they reuse a system. However, the quantity of source code is enlarged by repeating reuse, and the structure of the system becomes complicated.

The maintenance work of such a system causes the increase of the cost. Because a source code is hard to understand it and does not understand the revision method of the source code, costs increase. The effective solution for the problem is not yet found. The developers cope on an ad hoc basis. Complehension support and change technique with the meaning are necessary for us to solve this problem. We have to develop a method in order to reduce the number which should be modified comparing to the system before the change is applied. I want to reconstitute to be easy to understand the existing source code that the maintenance developer complicated. In addition, I want to reconstitute so that the maintenance person in charge works on maintenance effectively. A purpose of my study is to suggest those technique. Our technique is applicable to reconstruct other Legacy systems.

2 Technology and problems of the maintenance

I show two technology as reconfiguration technology of the systems, Refactoring and a study to propose of Daio. When I watched a program from the outside, Refactoring does not change behavior. With that in mind, Refactoring changes the inside of the program. Refactoring fixes the complicated program without changing behavior. However, Refactoring cannot perform reconstitution to change a meaning. In addition, the developer must think about a range with the need to perform Refactoring when he does a reconfiguration. When a developer is unripe, he may not understand the start that should perform Refactoring. A study of Daio is suggestion of the reconfiguration technique with the meaning for the design. However, it is rare for development specifications to be left on the site. Therefore it is rare that technique is used by a developer. On this account I want to establish reconfiguration technique with the meaning for the source code.

3 The structure of the Legacy system and the policy of the reconfiguration

The constitution of the Legacy system is old. As for it, introduction of the new service is difficult. As for it, Extensions is performed frequently. Therefore structure becomes complicated. In my study, I study it for Legacy system. I in particular study it for Web application written by Java. What specifications are added to a language by Java, and is changed is performed a lot in a short term. Therefore it is hard to recycle an old system made with Java, and a revision is difficult. By the early Web application development, standard constitution was not decided. Developers pushed forward development in ad hoc. With many of Legacy systems, plural roles coexist to one object. In addition, there is an object if used with many of Legacy systems for run time. For such objects, it is necessary for the developer to examine plural objects. It leads to maintenance cost increase.

As a policy of the reconfigurations, I use three tears architecture that are the standard constitution of the Web system. Three tears arcitecture consists of it at three hierarchies. Three hierarchies are presentation layer making the Web page, business logic layer processing data, and database

layer saving data. And I perform flow analysis to grasp the constitution of the system.

4 Suggestion of the reconfiguration technique

I suggest reconfiguration technique by analyzing Legacy system. The suggestion technique consists of it by two procedures.

- Separation of the layer
- Extraction of effective objects

By the Legacy system, there was the problem that plural roles coexisted to one object. I divide the layer with Three tear arcitecture. Therefore I can classify objects every role. I find the description that should switch over from an object becoming complicated to presentation layer and the database layer. And I make them the other method. The division of the layer is performed by shifting to the suitable class by the method.

There are many objects which are not used for run time by addition and a revision being performed in Legacy system, and there is a problem included in. I perform flow analysis and extract a useful object. I can delete a disused object by doing so it. I analyze it from the page that a user starts. I can detect an object becoming useful for run time by extracting the object which the page makes.

5 An experiment and evaluation

I tested it with E-commerce system. By the separation of the layer, I performed an experiment to increase data to a certain object to confirm the effect of the reconfiguration. By a system before performing reconstitution, a change point dispersed. And it was difficult to keep consistency. For it, data structure was made a lump by a system after a reconfiguration. I was able to perform a coherent revision.

In Extraction of effective objects tecnique, I performed flow analysis for an E-commerce system. As a result, I was able to arrange a flow every each layer of Three tear arcitecture. I reduced update to sit astride in a layer.

6 Conclusion

In my study, I suggested the reconfiguration technique of the system which intended for a source code. A function in the object was finished in one by applying the division technique of the layer. I was able to perform addition and the revision of the function easily. I can extract only an object made with a system at the time of practice by applying the extraction technique of a useful object.