Title	UML Java
Author(s)	,
Citation	
Issue Date	2005-03
Type	Thesis or Dissertation
Text version	none
URL	http://hdl.handle.net/10119/1926
Rights	
Description	Supervisor: , ,

Japan Advanced Institute of Science and Technology

Automatic Generation of Corresponding Relations between UML Diagrams and Java Source Code

Natsuki Watanabe (310126)

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

February 10, 2005

1 Background and Purpose

In a change revision work in software joint development, work to change documents and a source cords having complicated dependence relation adequately without exception is not easy. When "B depends on A", here, dependence relation is relation to "have to change B when you changed A".

Therefore, for an UML drawing and a Java source cord, development of an information model to help an effective work change by security is pushed forward. I define the meta model that connected an UML drawing and a constitution element type of a Java source cord by dependence relation and add dependence relation to a figure of UML and an example of a Java source cord automatically to be concrete by using it. By this, I generate an information model for change support and can plan automation of dependence-related search work.

A definition of correspondency to show an UML drawing and an agreement part between Java source cords is indispensable to realize this. However, it is not easy to define neither—neither of correspondency because structure does not agree with an UML drawing made at the time of a design with the Java source cord which is the implementation to the public.

Therefore, by this research, I break off problems of disagreement of such a structure and suggest technique to generate correspondency of both. In addition, with an existing UML drawing and Java source cord, I build a system generating correspondency of both automatically. An UML drawing intends for a figure of class and an illustration of collaboration.

2   Definition of problems

It is necessary to consider a difference of design time structure and implementation time structure of software to define correspondency between a certain UML drawing - Java source cords for the purpose of this research. I generally realize a function of a model element by a set of a class of higher than nothing without it being always it with a single class when I implemented one model element which there is in an UML drawing as a source cord.

The following thing is nominated for the reason why such a difference occurs.

- Application of a rule of a designer (an individual, an organization)

- Application of a design pattern

- Security of a merit in object oriented programming

Therefore it is impropriety for generation technique of general correspondency only by extracting a class agreeing in a name and an attribute of a model element. By this research, I define disagreement of structure of both as the division of a class to be able to put in a source cord and extract all classes to fall under as a correspondence part in a source cord automatically and suggest technique to exhibit.

3   Generation process of correspondency

By this research, I divide an UML drawing and a generation process of correspondency of a source cord into two phases of next phases.

1. Extraction of a group of correspondence classes in a source cord Extraction of a group of correspondence classes in a source cord

2. Inquiry of groups of correspondence classes by appointment of a model element I can refer for a thing having correspondency from a set of a group of correspondence classes which were accumulated by the correspondency generation method that I defined by the above by appointing a model element of an UML drawing. I design it and implement prototype of an automatic generation system and realize this.

4 Extraction of groups of correspondence classes

I give grounds and a specific standard and have to search to extract a group of correspondence classes from a source cord. At first I chase existing specific relation between classes in a source cord. Furthermore, for relation between each classes, I add a chase range as a condition to limit a range to chase. In this research, I search for relation between the next classes.

- Extension and Implementation

  In a source cord, it is very likely that a class or interface connected by the succession, relation of implementation constitutes the same class when I put it on an UML drawing. Include such a class except a class except a user definition class or a set of interface in the case of number of the elements 1 and name it by this research with a succession group. For a succession group, a function and connection of a model element remove a few class or interface by defining a chase range and extract only a group of correspondence classes. I was able to extract only a group of correspondence classes from a succession group with high precision when I added a chase range and applied to an existing source cord.

- Reference relation

  Two succession groups connected by reference relation in a source cord can satisfy a function of a model element by these cooperation. The cases that the other succession group expresses the state that one succession group has are nominated for an example with a State pattern. I added a chase condition to reference relation and applied a more

constant condition to a, in addition, existing source cord. As a result, when I searched for only the succession / implementation, I compared it and there was more it and was able to extract the succession group which I should have included in a group of correspondence classes.

5   Summary and Future problems

I recognized that an UML drawing and a definition of correspondency between Java source cords and automatic generation were indispensable for development of the information model who helped a work change in software joint development and at first, by this research, defined the difficulty that should have been solved to achieve this. And it solved this difficulty and suggested technique to generate correspondency of both. In a specific condition bottom, I was able to extract a group of correspondence classes with high precision when I applied to an UML drawing and an example of a Java source cord for suggestion technique. I give below a future problem in this research.

- Detailed definition of a chase range of relation between classes When a succession group has reference to other succession groups, I am unconditional and include a succession group referred to by a chase range of the reference relation that I defined by this research in a group of correspondence classes. Therefore it stayed as an important problem that there was a case to drop precision not to be able to remove the class which you must not include as a group of correspondence classes. It is necessary to go into details about a reference-related chase range more in future to get high extraction precision under a general condition.

- Extraction of a more detailed correspondence part

  By this research, I extracted a point in a source cord corresponding to a model element in a grain of a class unit. Furthermore, I narrow a range and, by a model element, have to extract it with a field level and a method level.

- Developing GUI of an automatic generation system

I get possible to help an UML drawing and visual understanding of correspondency between source cords by making a correspondency automatic generation system GUI.