A Research of Software Development Environment with Source Code Compreheision Support.

Satoshi Niikura (110123)

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

February 7, 2008

Keywords: Comprehension Support, Source Code, Filtering, Development Environment.

1 Introduction

Inprovement in software functions makes the amount of codes larger and the structure more complex. Sometimes we use a open source for development, or employ norvices who are not familier with the code. In those cases, we have to need read codes written by others. It is difficult that understanding the structures of source codes. In the worst case, developers have to read the whole of source codes.

Code comprehension is one of the method for helping those developers. We define code comprehension supports as functions that extract and show some parts of codes which should be modified as Previous researchers provide some tools . Most of them cannot control the amount of information

We propose a comprehension tools for C language code, which consist of several fine-grained operations called *filters*. Our tool can provide information only developer requires. It is expected that developer decreasing the cost of sotfware maintenance.

2 Requirment for Source Code Comprehension Support

Information generally required by developers contains that of structures in cases such as (1) declaration of variable type, (2) dependency relations We employ an approach for composition of fine-grained filters.

3 Realization of Filters

A user can extract information by iterative application of (1) filter selection, (2) filter composition. Nagai proposed a basic set of filters we extend as following filters:

1. Extract executive part in several control structures
2. Extract a declaration part or executionpart in some semantic block
3. Extract ranges in which a specified variables occurs
4. Extract ranges which are result of tracing effectness of dependency in assignment statements
5. Extract neibours of specified statement
6. Extract a definition / reference parts for a specified global variable
7. Extract a definition of variable and its type

A user can control the target of extraction by changing parameters which represents line number or depth tracing dependency. Assume that an assignment `z=x+y`, `z` is a global valiable. In some case, a user wants know range of effectness of its valiables, the system gives a filter In other case, a user wants tracing dependency, the system gives a filter of extraction

The user can extract various kind of infomation by application of sequential composition, which mean make a filter output as another input, and composition defined by logical AND/OR of the results given by each filters.

4 Implements and Evaluation

We implement filters defined by section 3 as function on AST, which are the result of syntax/semantic analysis of codes. Our tool consists of the following three parts: anarisis, extraction, showing part. In anarisis part, the tool gets a AST by using outside tools. In extraction part, the tool gets subset of AST by using designed filter. And in the other, the tool displays sub code corresponding AST.

Users can select operation in (1) setting a filter and its parameters, or (2) running filter composition. In all operations, the tool keeps these results as histories brecause of abailables after.

Because of checking effectivity in comprehension support, an experiment have done by open source software. This is the reasons that it can find part modified from old version into new by using release note. So, it is confirmed whether the tool get required informations, by comparing the range required modification where is extracted from older by the tool with the range from newer. As the result, the tool has succeed in terms of being able to extract modified part for another tool being not able, in which find table of column which register pointer of special function.

5 Conclusion

Source code comprehension support tool has been developed which extract information required by user with composition by primitive filters. In the imprements, users can get required informations in less time, by controling activity of filter with parameter and by extracting direct/AND/OR composition repeatedry.

It finds the tool effective by experiment with open source software. However, users must try and error with various filters to select filter selection and composition. There are problems, for example, providing pairs of filter and its parameters in special cases.