Title	多言語に対応した解析・理解支援のフレームワークに関する研究
Author(s)	伊藤，徹弥
Citation	
Issue Date	2009-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/8105
Rights	
Description	Supervisor:鈴木正人准教授，情報科学研究科，修士

A Research on Framework for Code Analysys and Code Comprehension Support supporting Multi Language

Tetsuya Ito (0710008)

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

February 5, 2009

Keywords: program complehension, supporting environment , navigation tree , language dependency, multi language.

1 Introduction

In current software development, it is necessary to find the part where a developer realizes a specific function in sake corresponding to requirements. However, there is the case that it is difficult that a developer understands where the specific function is realized because the structure of the source code is complicated definitely.

In this research, we propose a methodology how to extract required information from source codes effectively. We propose some filters and specify their function in order to apply source codes written in different language. Our goal is maked a framework from the result.

2 Control of information

In this research, we employ to control information is information extraction using primitive filters. Niikura proposed a method. We introduce set of concern and introduce that a user changes a range paying attention to the cause for concern with expansion of a concern. We realize expansion of concern with a combination of the filter. Furthermore, we suggest some more filters in order to adopt them to sources written in different program language. We use ten kind of filters.

- to decide concern
    - Extract a definition of variable and its type(DECL)
    - Extract a class or a function or a value including the character string that we input(EXT_NAME)
- to expand concern
    - Extract basic blocks corresponds to specific control structures(CTRL)
    - Extract several lines which are located at number of(NEIGHBOUR)
    - Extract occurences which are result of tracing effectness of dependency in assignment statements(TRACE)
    - Extract the declaration part of specified function(F_DECL)
    - Extract the reference part of the function(REF)
    - Extract classes in which specified interface is implemented, or abstruct class is extended(IMPLEMENTED)
- to reduce concern
    - Extract a declaration part or executionpart in specified semantic block(SEP)
    - Extract set of lines in which a specified variables occurs(RANGE)

All of filters take to parameters.AST showing the whole source codes. As a parameter common with these 2, the filter extracts information from a source code. There is AST showing the whole source code. And there is a set of the information to express concern. concern line, a function to pay attention to, a variable to pay attention to are nominated for a parameter peculiar to each filter. For example, it is assumed that a developer pays attention to a certain function when he decodes a source code. It is assumed that it is necessary for a developer to know a part calling the function that he pays attention to then. As a parameter of the case, a developer gives a filter the name of the function and the line number, number of arguments and a model and the type of the return value. As a result, the part where the function is called is extracted with line number.

Filters supposed by Niikura are considered to apply only sources in C language. In order to apply them to other language such as Java language, we classified characteristics peculiar to a language. Overload is nominated for the example as the concrete example.

3 Navigation

We examined information extraction method with the filter so that a user extracted information. We show the procedure next.

1. Search the name that it is very likely that there is connection in the realization part of the function

2. Extract a control flow from a function summons and an argument included in the result that we searched

In addition, we introduced a navigation tree to nominate efficiency for. A navigation tree is a tree expressing the application history of filters. The node keeps the information of the attention part. The edge keeps a information of the filter which we applied and parameter. You can easily get information of a part that we paid attention to and the filter which we applied. if we watch a generated tree. When the result that we extracted has as nothing related to do with a change, we can do rollback by showing a generated tree in a user.

4 Evaluation of experiment

We experimented on two so that we identified the effectiveness of the methodology. We assumed an experiment to extract the part that needed a function change the first experiment. We assumed an experiment whether could extract all the parts that we should have changed the second experiment.

We used two versions of the open-source software for an experiment. At first, from the release note of the new version, we get information about the difference with the old version. Next, we assume this a change requires in the old version. We use a proposal method for an old version. We use suggestion technique for an old version and extract the candidate of the part that I should change. And we acquire for difference of an old version and the new version and compare the result that I extracted with for the difference. If the result that we compared accords, it may be said that we was able to extract the part that I should change.

As a result of first experiment, we were able to extract an objective part.

As a result of second experiment, structure of the software was changed, but a revision was given to an extraction result entirely.

From this result, the proposal technique can be said to be effective.

5 Conclusion

We proposed by an effective information extraction method by the usage of the primitive filter. And, we introduced a navigation tree, and we enabled rollback when an information extraction process stopped. In addition, we improved the filter. We showed it about language dependence, and we showed a multi-language about a part to take into account when we let we cope. And we showed it about a part to take into account when we let language dependence and multi languages support to make a framework.