Title	ソースコードの依存解析と視覚化手法に関する研究
Author(s)	横山，浩一
Citation	
Issue Date	2001-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1437
Rights	
Description	Supervisor:落水 浩一郎，情報科学研究科，修士

Dependency Analysys and Visualization of Source Codes

Kouichi Yokoyama

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

February 15, 2001

Keywords: visualization, dependency analysys, software maintenance.

1 Background and Purpose

As the object technology has advanced, softwares designed and implemented by it has increased. Such softwares are highly modularized into small peaces (classes), so it isn't easy for us to understand their whole structure and runtime behavior by reading raw source codes. Unfortunately we have many cases to do so. Old softwares often lose their desing documents. Design documents may not describe the resulting softwares precisely. When we modify such softwares for accommodating them to changing requirements or port them on new platforms, we have to read their source codes. We needs a computer support for understanding softwares from their source codes.

There are many works for the same purpose. RAD environments generally have a class viewer and/or a syntax-directed editor. These tools provide visualization of source codes with figures or/and colors to makey it easy to understand them. These tools, however, provide only one or two levels of abstraction in the visualization. User can't freely come and go between a global and local view. For example, a class viewer provides only a global structure of classes, and syntax-directed editor provides only a local syntactical information. They need to switch two views related loosely. It is hard to buld a consistent model of a software through such tasks. Such tools should provide many levels of abstraction seemlessly.

We propose a source code browser 'SZ_Browser'. This seemlessly provide many abstraction levels in visualization of source codes on demand by users. This supports source codes written by the Java Programming Language. Users easily grasp the relations between each view point of one abstraction and another level.

2 Dependency Analysis

The dependency relationship defined in this paper means that there exist inheritance or reference relationships between/among constituents. Dependency analysis means to get those relationships from a source code. We can utilize these relationships as supplementary information to read and understand a source code without specifications.

3 Presentation Method

There are many relationship between cases described in the previous section. There are several levels of the relatioship-granuality in an object-oriented program: between classes; between operations; between class and operation. It is very clumsy and complex to show those relationships at a time on the screen. We need some presentation method which extract and show the part of relationships based on some criterion. One of the hint to define the criterion may be distance of two classes. Using the distance we can only show the relationships being interested from some view. Other relationship are hidden with some cue.

We call this concept the Semantic Zooming. We can adjust our focus only to the matter of concern by using semantic zooming. If we want to know the details of some relationship, we can add the relationship displayed by zoom-in. If we want to get higher level meaning of the relationships, we can decrease the number of relationships by zoom-out. In these cases, the size of font should be varied together.

Another idea for semantic zooming is as follows; Suppose that there are two clusters of relationships and the level of granuarity of each cluster is different. In the case, we sometime want to enlarge or shrink to make the granuarity of zooming equal. For an example, suppose there is a cluster of classes and realtionships among them. And suppose there is another cluster which generates object instances based on some rules. We want to see both clusters to understand according to a hierachy relationship like class, method, codes in a method.

4 Information Visualization

Visualization using Computer Graphics(CG) have been mainly used in visualizing the computation results in scientific computation area. We, however, are going to extend these techniques to Information Filtering, Software Development and so on because of spread and popularization of low cost and high performance computer with high resolution display. Card proposed many new applications area of the information visualization with proposing the idea; interaction with abstract data.

His idea enable us to apply information visualization technique not only for direct indication of visual object like picture but also to abstract entities with revealing the internal structures of the object.

Next step is a technology development of a new type of user interface which allows us to play zoom-in and zoom-out operation based on the level of the granuality of internal structure.

We applied the concept and the techniques mentioned above to an object-oriented program. The model and techniques which we developped in this paper allows us to perform semantic zooming to the source text of a program with respect to the internal structure of the source code. I believe this technique is better than the traditional Class-Viewer techniques because our tool support us directly to understand the relationship between a whole view of program and the detail internal structure enable of the program.

5 Prototype Implementation

In this paper, we adopted Java programs as data. We propose the dependency relationships which should be useful for understanding Java source code. We also developed the algorithm which can analyze the dependency relationships among constituents of a Java program. We also discussed the performance of understanding programs comparing to manual analysis of source codes.

Based on the dependency relationships and the information extracted by our tool, we propose the framework of information visualization. We show the effectiveness and usefulness of our model and a tool by introducing four states and three interactions necessary for analyzing and visualizing informations.

Prototype is partially developed to show the effectiveness of our fundamental research results.