SHINOBI: A Tool for Automatic Code Clone Detection in the IDE

Shinji Kawaguchi*, Takanobu Yamashina†, Hidetake Uwano‡, Kyhohei Fushida*, Yasutaka Kamei*, Masataka Nagura*§, and Hajimu Iida*

*Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, Nara, Japan
Email: {kawaguti, kyohei-f, yasuta-k, nag}@is.naist.jp, iida@itc.naist.jp
†Nihon Unisys, Tokyo, Japan
Email: takanobu.yamashina@unisys.co.jp
‡Department of Information Engineering, Nara National College of Technology, Nara, Japan
Email: hideta-u@is.naist.jp
§Presently with Systems Development Laboratory, Hitachi, Ltd.

Abstract—Recent research has acknowledged that code clones decrease the maintainability and reliability of software programs, thus it is being regarded as one of the major factors to increase development/maintenance cost. We introduce SHINOBI, a novel code clone detection/modification tool that is designed to aid in recognizing and highlighting code clones during software maintenance tasks. SHINOBI is implemented as an add-in of Microsoft Visual Studio that automatically reports clones of modified snippets in real time.

I. INTRODUCTION

Code clones are one of the major obstacles to software maintenance. If a defect is contained in a code portion that has been copied many times, we have to investigate not only that code portion, but also all copied codes. This is very time consuming especially for large scale software.

Presently, many code clone detection tools have been proposed, with some of these tools highly integrated with the development environment [1], [2], [3]. This is very important as programmers change code clones using some sort of development environment, not using clone detection tools.

These clone detection environments require the running code clone detection to be manually switched on. However, such manual detection cannot be run so frequently because code clone detection from the whole source code takes more than five or ten minutes for large scale software. CloneTracker [1] reduces detection tool invocation by using a Clone Regional Description (CRD). CRD is robust for source code changes, and it enables tracking of code clones if source codes are modified from when the code clone detection was applied. However, CloneTracker needs to re-run the code clone detection tool to detect newly created code clones.

We believe that it is important to detect newly created code clones so as to react to code clones in their early stages. This is because it is practically recommended to refactor code clones that appear too frequently as it is most likely not part of the original design. In such a situation, the code clone detection results should be shown in real time, reducing the spread of this code clone and reducing refactoring.

This paper introduces SHINOBI, a tool for automatic code clone detection. The main features of SHINOBI are as follows.

- SHINOBI is highly integrated with Microsoft Visual Studio. For instance, it is implemented as an add-in of Visual Studio. A programmer can easily check and edit detected code clones.
- SHINOBI automatically detects code clones with source code being edited. The detection process is automatic, implicit, and quick. A programmer can get a list of code clones without noticeable time penalty whenever he develops with the IDE.
- SHINOBI highlights code clones to help recognize code clones during software maintenance tasks.

II. SHINOBI

Figure 1 shows a screenshot of SHINOBI. SHINOBI automatically detects code clones being edited in Visual

![Figure 1. SHINOBI Client Screenshot](image-url)
SHINOBI tells you how many code clones exist in the whole source code as soon as a copy of a code portion is made. This feature helps us notice code clones and which code needs refactoring at an early stage. Such automatic detection is especially essential if you are unaware of other developers creating clones. Using SHINOBI, you can see how much code has been copied by other developers as well as yourself.

SHINOBI is also used for software maintenance. When fixing source code, SHINOBI informs of the code clones in the source code which could be candidates of code to be fixed. SHINOBI automatically detects code clones without explicit code clone detection invocation. Without SHINOBI, more attention is needed to identify code clones when source code is fixed.

IV. Conclusion
In this paper, we introduced SHINOBI, a clone-aware software development environment, highly integrated with Microsoft Visual Studio. SHINOBI automatically and implicitly detects code clones and shows them to help recognize them. Developers easily understand how many code clones exist in the whole source code. As stated in Section III, SHINOBI will be useful when developers have to edit code clone in software maintenance tasks. SHINOBI is available at http://sdlab.naist.jp/prj_shinobi.html

ACKNOWLEDGEMENT
We would like to thank employees at Nihon Unisys Ltd. for their cooperation and valuable advice. This work was supported by StagE Project, the Development of Next Generation IT Infrastructure, supported by MEXT.

REFERENCES