

Title	ソフトウェア開発環境における構成要素の分類と競合回避手段に関する調査研究 [課題研究報告書]
Author(s)	権, 亨漢
Citation	
Issue Date	2013-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/11302
Rights	
Description	Supervisor:鈴木正人, 情報科学研究科, 修士

An Investigative Study on Classification of Elements and Methods for Avoidance of Competition on Software Development Environment

KWEON, Hyung Hahn (0910701)

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

February 2, 2013

Keywords: Integrated Development Environment, Conflict, Behavior Dominant Information, Software configuration, Software preference

In software development process recent years, Integrated Development Environment tools such as Eclipse, Microsoft Visual Studio and others are highly used in developing and running phases as well as traditional command-line based environments. These IDEs integrate several functionalities of software generating activities including configuring with many preference information, compiling with specific options, linking with libraries and generating of executable forms from source codes written by user. Inspecting behaviors of these tools, we can describe that these functionalities are kind of converting from input to specific output and called by predefined orders. The results of converted executables include stand-alone single input-to-output type and cooperated type with outer software functions such as Web framework.

In order to make sure correct running of the program and output result from it with specific target platform, it is important to describe additional information such as options and preference files correctly, not only the code algorithm and syntactic structure. Operations of converting from source codes to executable form and correct running result need specifying of various informational parameters that determine the characteristics of result output.

In addition, we also face various change of environment around tools not only the tool itself in software development phases. It includes change of related libraries and frameworks such as version up and downs, change of target platform, restrict of

specified versions due to requirement specification. Without considering integrity of tools and its environment, we get incorrect results from the source codes. It is very hard to find and fix these kinds of problems without deep understanding of nature of configurable preference information to the software tools.

In this report, we define ‘conflict’ as wrong result of generating and running of software from source codes due to incorrect determination or specification of preference information. We define some basic informational concepts including the “Behavior dominant information” which is the information set which can be affected to the program except source code structure, based on language and environmental elements of software and its configurations. To detect and find how to resolve the reason where the conflicts occur from software. With these basic concepts of information units, we investigate, find the possible behavior dominant information and describe relationship of conflict based on that information.

We apply these results to some actual platforms including C/C++, Java and some Web frameworks such as Tomcat/Struts2 and ASP.NET and others. With the result, we describe that these behavior dominant information are coupled with specific programming language and development environments. Some information such as the options and mapping description for converters and libraries are highly tightened with languages such as C/C++ and Java, Location information of preference information with Web framework tools. These information units helps us to determine target software configuration, and to develop some tools such as analyzer of detecting inconsistency in preferences requirements.