JAIST Repository
https://dspace.jaist.ac.jp/

| | |
|---|---|
| Title | アスペクト指向的なモジュール記述を可能とする仕様記述言語 |
| Author(s) | 山田，聖 |
| Citation | |
| Issue Date | 2005-03 |
| Type | Thesis or Dissertation |
| Text version | author |
| URL | http://hdl.handle.net/10119/964 |
| Rights | |
| Description | Supervisor:鈴木 正人，情報科学研究科，博士 |

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

# Aspect-Oriented Behavioral Interface Specification Language

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

January 11, 2005

## Abstract

When we develop the software, it is general to split a software into some modules in order to avoid difficulties caused by the scale or the complexity of the software. The software composed of such modules can be modeled on some independent modules implementing its own function and on dependency relation between such modules defined by the use of function implemented by other modules. Design by Construct (DbC) is a software organization methodology based on the idea that regards such a relation between modules as a contract. In DbC methodology, the function supplier shows some conditions that must be satisfied just before or after the function execution. And also the function supplier and the user make contract for the satisfaction of such conditions. Then this contract arranges responsibilities between the function supplier and the user.

In the object-oriented programing language based on DbC, conditions that must be satisfied just before or after the execution of the method are expressed as logical expressions called pre-condition and post-condition respectively and they are described as assertions. When the condition does not satisfied, we can understand that caller of the method violates the contract when the satisfaction of pre-condition failed, or callee violates it when the post-condition does not satisfied. In this specification description style, logical formula of the conditions specified for each methods become complicated, and this makes difficult to develop incrementally keeping consistency of assertions and coherence between assertions and program. In the ordinary pre-/post-condition declaration method, we must embed the descriptions of such conditions in the program code, therefore we cannot modularize them independently of the program structure.

In this paper, I propose an aspect-oriented modularization method and its description language that makes possible to modularize pre-/post-conditions independently from the structure of the methods or the class. This modularization method use the notions such as joinpoint (the position on the program execution flow where we can check pre-/post condition), point cut (the set of joinpoints), and advice (the pair of a pointcut and a logical expression that are expected satisfying at the pointcut). And also it defines the assertion aspect as a set of advice. This modularization method makes it possible to split condition declarations that are expected to satisfy on a join point into some advices. Using this capability, when it is possible to describe the behavior of an object as a composite of some independent sides of the behavior, assertion descriptions for each side of behavior can be modularized as assertion aspects independently. Also, in this modularization methods, some assertion declarations that have same condition expressions can be described as an assertion declaration using the advice. Using this assertion modularization mechanism, we can avoid to glow the size and the complexity of assertion declarations caused by the growing size of the class.

**Key Words: Design by Contract, Aspect-Oriented Programming, Assertions**