Dependence flow graph for analysis of aspect-oriented programs

ABSTRACT

Program analysis is useful for debugging, testing and maintenance of software systems due to information about the structure and relationship of the program's modules. In general, program analysis is performed either based on control flow graph or dependence graph. However, in the case of aspect-oriented programming (AOP), control flow graph (CFG) or dependence graph (DG) are not enough to model the properties of Aspect-oriented (AO) programs. With respect to AO programs, although AOP is good for modular representation and crosscutting concern, suitable model for program analysis is required to gather information on its structure for the purpose of minimizing maintenance effort. In this paper Aspect Oriented Dependence Flow Graph (AODFG) as an intermediate representation model is proposed to represent the structure of aspect-oriented programs. AODFG is formed by merging the CFG and DG, thus more information about dependencies between the join points, advice, aspects and their associated construct with the flow of control from one statement to another are gathered. We discuss the performance of AODFG by analysing some examples of AspectJ program taken from AspectJ Development Tools (AJDT).

Keyword: Dependence flow graph; Control flow graph; Dependence graph; Aspect-oriented; Program analysis; Intermediate representation; Maintenance