Abstract

The main applications of program slicing include various software engineering activities such as program understanding, debugging, testing, program maintenance, complexity measurement and so on. Program slicing is a feasible method to restrict the focus of a task to specific sub-components of a program. It can also be used to extract the statements of a program that are relevant to a given computation. Applying slicing technique to software architectures can benefit software development in two main ways. The first one concerns maintenance of a component-based software. By using slicing tools on an architectural description, we can determine which components might be affected when a given component is modified. Second, architectural reuse can be facilitated. While reuse of code is important, reuse of software design and patterns are expected to offer greater productivity benefits and reliability enhancements.

References

Program Slicing using Test Cases

- David Binkley, "The Application of Program Slicing to Regression Testing\textquotedbl;
- S. S. Anju, P. Harmya, Noopa Jagadeesh, R. darsana, "Malware detection using assembly code and control flow graph optimization\textquotedbl", ACM Digital library, No. 52, 2010

Index Terms

Computer Science
Software Engineering
Keywords

Program slicing  test cases  static slicing  dynamic slicing  control flow graph
program dependence graph