

Polymorphic Bytecode Instrumentation

Bytecode instrumentation is a widely used technique to implement aspect weaving and dynamic analyses in virtual machines such as the Java Virtual Machine. Aspect weavers and other instrumentations are usually developed independently and combining them often requires significant engineering effort, if at all possible. In this article we present polymorphic bytecode instrumentation (PBI), a simple but effective technique that allows dynamic dispatch amongst several, possibly independent instrumentations. PBI enables complete bytecode coverage, that is, any method with a bytecode representation can be instrumented. We illustrate further benefits of PBI with three case studies. First, we describe how PBI can be used to implement a comprehensive profiler of inter- and intra-procedural control flow. Second, we provide an implementation of execution levels for AspectJ, which avoid infinite regression and unwanted interference between aspects. Third, we present a framework for adaptive dynamic analysis, where the analysis to be performed can be changed at runtime by the user. We assess the overhead introduced by PBI and provide thorough performance evaluations of PBI in all three case studies. We show that pure Java profilers like JP2 can, thanks to PBI, produce accurate execution profiles by covering all code, including the core Java libraries. We then demonstrate that PBI-based execution levels are much faster than control flow pointcuts to avoid interference between aspects, and that their efficient integration in a practical aspect language is possible. Finally, we report that PBI enables adaptive dynamic analysis tools that are more reactive to user inputs than existing tools that rely on dynamic aspect-oriented programming with runtime weaving. These experiments position PBI as a widely-applicable and practical approach for combining bytecode instrumentations.

Author: Walter Binder (walter.binder@usi.ch)