

## **MAJOR: AN ASPECT WEAVER WITH FULL COVERAGE SUPPORT**

**Alex Villazón, Walter Binder, Philippe Moret and Danilo Ansaloni**

### **ABSTRACT**

Prevailing Aspect-Oriented Programming (AOP) frameworks for Java, such as AspectJ, use bytecode instrumentation techniques to weave aspects into application code. Unfortunately, those frameworks do not support weaving in the Java class library. When implementing aspect-based tools, such as profilers, debugger, or dynamic program analysis tools in general, the aforementioned restriction becomes an important limitation for successfully applying AOP.

In this article we present MAJOR, an aspect weaver with full coverage support. That is, MAJOR ensures that aspects are woven into all classes loaded in a Java Virtual Machine, including those in the standard Java class library. We describe the intricacies of instrumentation of the Java class library and present an extended instrumentation approach allowing the user to choose between a pure Java weaving solution based on a two-phases instrumentation scheme, or a single-phase one requiring a tiny native code layer. The single-phase approach allows to better isolate the weaving process from the execution of the woven code.

**Keywords:** Aspect-Oriented Programming, Aspect Weaving, Bytecode Instrumentation, Java Virtual Machine.