

MAJOR: UN ASPECT WEAVER CON SOPORTE PARA COBERTURA TOTAL

Alex Villazón, Walter Binder, Philippe Moret and Danilo Ansaloni

RESUMEN

Los *frameworks* actuales de Programación Orientada a Aspectos (*Aspect-Oriented Programming* - AOP) basados en Java, como ser AspectJ, utilizan técnicas de instrumentación de código de bajo nivel (*bytecode*) para insertar o aplicar aspectos dentro del código de las aplicaciones (proceso llamado *aspect weaving*). Lamentablemente, dichos sistemas no cuentan con soporte para aplicar aspectos dentro de la biblioteca estándar de clases de Java (*Java class library*). Esta restricción resulta en una limitación importante para la utilización exitosa de AOP en la implementación de herramientas basadas en aspectos, tales como *profilers*, *debuggers*, o herramientas para análisis de programas dinámicos en general. En este artículo se presenta MAJOR, un *aspect weaver* con soporte para cobertura total. Es decir, MAJOR se asegura que los aspectos son aplicados a todas las clases que son cargadas en una máquina virtual Java (*Java Virtual Machine*), incluyendo aquellas clases de la biblioteca estándar de Java. Se describe la complejidad de la instrumentación de dichas clases y se presenta un nuevo enfoque de instrumentación que permite la selección entre una solución en dos fases que está basada únicamente en Java, o una solución en una sola fase, pero que requiere una pequeña capa de código nativo. La solución monofásica permite aislar de mejor manera el proceso de aplicación de aspecto (*aspect weaving*), de la ejecución del código ya instrumentado con el aspecto (*woven code*).

Palabras Clave: Programación Orientada a Aspectos, *Aspect weaving*, Instrumentación de *Bytecode*, *Profiling*, *Debugging*, Máquina Virtual Java.