

DESARROLLO AVANZADO Y RAPIDO DE HERRAMIENTAS DE ANALISIS DINAMICO PARA JAVA

Alex Villazón, Walter Binder, Danilo Ansaloni and Philippe Moret

RESUMEN

Las técnicas de instrumentación de bajo nivel de código intermediario (bytecode) son utilizadas ampliamente en herramientas de ingeniería de software para la Máquina Virtual Java (Java Virtual Machine - JVM) para realizar formas de análisis dinámico de programas, como por ejemplo perfiladores (profilers) o depuradores (debuggers). Si bien la manipulación de código al nivel de bytecode es muy flexible, ya que las transformaciones no tienen restricciones, las herramientas basadas en esta técnica son tediosas de implementar y son propensas a la inserción de errores. Como una alternativa prometedora, la especificación de la instrumentación de bytecode a un nivel más elevado, utilizando Programación Orientada a Aspectos (Aspect-Oriented Programming – AOP), puede reducir el tiempo y el costo del desarrollo de herramientas. En este artículo, nos concentramos en tres deficiencias comunes en los sistemas AOP existentes con respecto al desarrollo de herramientas basadas en aspectos – (1) la falta de mecanismos para pasar datos entre consejos tejidos (woven advices) en variables locales, (2) el soporte para análisis estáticos definidos por el usuario durante la integración de los aspectos (weaving time), y (3) la ausencia de puntos de corte para inserción de aspectos (pointcuts) a nivel de bloques individuales básicos de código. Proponemos @J, un lenguaje y un tejedor (weaver) AOP basado en anotaciones que integra soporte para las tres características mencionadas anteriormente, cuyos beneficios son ilustrados con ejemplos concretos.

Palabras Clave: Programación Orientada a Aspecto, Análisis durante el proceso de Tejido de Aspectos, Instrumentación de Bytecode, Análisis Dinámico de Programas.