AspectJ™: the Language and Support Tools*

Erik Hilsdale, Jim Hugunin, Mik Kersten, Gregor Kiczales, Cristina Lopes, Jeffrey Palm
aspectj.org
Computer Science Laboratory
Xerox Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304, USA
{hilsdale, hugunin, mkersten, gregor, lopes, palm}@parc.xerox.com

ABSTRACT
Complex systems usually contain design units that are logically related to several objects in the system. Some examples include: tracing, propagation of interrupts, multi-object protocols, security enforcement etc. This crosscutting between those design units and the objects is a natural phenomenon. But, using traditional implementation techniques, the source code – i.e. the classes – becomes tangled with the implementation of the crosscutting concerns.

AspectJ is an aspect-oriented extension to the Java™ programming language that enables the clean modularization of crosscutting concerns. Using AspectJ we can encapsulate in program modules (aspects) the implementation of those design units that would otherwise be spread across the classes.

This demo illustrates what the AspectJ language can do and it shows the tools that support developing programs with this language. We present an example program, and demonstrate the edit-compile-debug cycle in an IDE that supports AspectJ.

* This work was supported in part by the Defense Advanced Research Projects Agency under contract number F30602-97-C-0246.

OOPSLA 2000 Companion Minneapolis, Minnesota