Aspect-Oriented Programming*

Gregor Kiczales
University of British Columbia
201-2366 Main Mall
Vancouver, BC V6T 1Z4, Canada
+1.604.822.4806
gregor@cs.ubc.ca

Erik Hilsdale
Xerox PARC
3333 Coyote Hill Rd
Palo Alto, CA 94304, USA
+1.650.812.4735
hilsdale@parc.xerox.com

ABSTRACT
Aspect-oriented programming (AOP) is a technique for improving separation of concerns in software design and implementation. AOP works by providing explicit mechanisms for capturing the structure of crosscutting concerns. This tutorial shows how to use AOP to implement crosscutting concerns in a concise modular way. It works with AspectJ, a seamless aspect-oriented extension to the Java(tm) programming language, and with AspectC, an aspect-oriented extension to C in the style of AspectJ. It also includes a description of their underlying model, in terms of which a wide range of AOP languages can be understood.

General Terms
Design, Languages, Theory.

Keywords
Aspects, aspect-oriented software design, AspectJ, AspectC, concerns, crosscutting, modularity.

1. INTRODUCTION
Aspect-oriented programming (AOP) is a technique for improving separation of concerns in software design and implementation. AOP works by providing explicit mechanisms for capturing the structure of crosscutting concerns.

Using traditional techniques the implementation of concerns like exception handling, multi-object protocols, synchronization, and resource sharing tends to be spread out across the source code. The lack of modularity for these concerns makes them more difficult to develop and maintain.

This tutorial will show how to use AOP to implement concerns like these in a concise modular way. The effect of using AOP on modularity, extensibility, separate development and overall program comprehensibility will be discussed, as well as issues in the adoption of AOP into existing projects.

The tutorial will work with AspectJ, a seamless aspect-oriented extension to the Java(tm) programming language, and with AspectC, an aspect-oriented extension to C in the style of AspectJ. It will also include a description of their underlying model, in terms of which a wide range of AOP languages can be understood.

Applications of AOP techniques and environments will include general-purpose programming, operating systems, and embedded systems.

2. EXPERIENCE REQUIRED
Attendees should have experience doing object-oriented design and implementation, and should be able to read Java and/or C code. No prior experience with aspect-oriented programming is required.

3. REFERENCES
[1] AspectJ website. http://aspectj.org.

[2] Aspect-Oriented Programming website. http://www.parc.xerox.com/aop.

[3] Kiczales, G., et al. An Overview of AspectJ. In Proceedings of the European Conference on Object-Oriented Programming (ECOOP). Springer-Verlag, Budapest (2001).

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

ESEC/FSE 2001, Vienna, Austria