Aspect-Oriented Programming

Gregor Kiczales
University of British Columbia
201-2366 Main Mall
Vancouver, BC, Canada
gregork@acm.org

Categories and Subject Descriptors
D.3.3 [Programming Languages]: Language Constructs and Features – classes and objects, modules, packages.

General Terms
Languages.

Keywords
Aspect-oriented programming, aspect-oriented design, software development.

1. INTRODUCTION
Aspect-oriented programming (AOP) is changing the way software is developed. AOP builds on previously modularity techniques like procedures and object-oriented programming. AOP goes beyond these techniques by supporting the modularization of crosscutting concerns. AOP can make it possible to develop modular designs and implementations of concerns like synchronization policies, business rules, resource management, architecture enforcement, security rules, persistence layer interaction, performance optimizations and many others. Adoption of AOP is proceeding quickly, especially in the enterprise Java community.

This tutorial will present a deep introduction to AOP. The course is intended to benefit developers, technical managers, and researchers interested in an introduction to the value of AOP. The emphasis will be on the practical impact of AOP on software development. We will cover the problems AOP solves, how it works, how to use it, and how to judge when it might be useful.

We will also discuss key technical issues of the underlying technology, and address the most important concerns about AOP.

The main topics we will address are:
Motivation, Concepts and Mechanisms
What is AOP? What problems does it solve? How does it work? How does it relate to OOP and other technologies? What effect does AOP have on designs and code? What is the role of tool support in working with AOP?

Going Beyond Programming
How does AOP affect the whole development process? What does it mean to "think aspects" in both design and implementation? How can understanding AOP improve designs and code even if you do not use an AOP language?

Adoption Strategies
What are the best approaches for adopting AOP? We will discuss an incremental adoption strategy that enables teams to begin working with AOP and getting value from it, while still minimizing risk to the shipping product.

Long Term Implications
Where is AOP going? What are the potential strategic leverage points in terms of AOP and platform architectures? What do different industry players appear to be doing?

The tutorial will work primarily with AspectJ 5, the most widely used AOP language today. But the basic concepts discussed apply to all AOP systems in use today.