# Extending the Eclipse Runtime

## - An AspectJ-Enabled Eclipse Runtime Engine-

- Eclipse BoF at AOSD´04 -

Martin Lippert
lippert@acm.org
www.martinlippert.com

# Motivation

- Use Eclipse 3.0 RCP to develop enterprise applications
- Use AspectJ to improve modularity

- **What happens if we want to use both techniques to develop applications?**
  - Especially to modularize cross-plugin pointcuts

# Solution: A load-time weaving runtime

- The basic idea:
  - Let the Eclipse runtime weave aspects into plugins at load-time
- The OSGi-based runtime architecture of Eclipse 3.0 allows:
  - to hook into the class-loading of the runtime via specialized framework adapters for the OSGi layer
- AspectJ 1.1 provides the necessary weaver API to implement load-time weaving
  - AspectJ 1.2 will officially support load-time weaving

# Demo

- **Using the Eclipse IDE 3.0 itself as the application enhanced via aspects**

(c) Copyright IBM Corp. and others. 2000, 2003. All rights reserved. Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S., other countries, or both.

With special thanks to Chris Laffra for the Monitor plugin

# Eclipse as Research Platform

- Eclipse 3.0 runtime is more flexible
  - Specialized adaptors to handle class loading
  - DynamicImport features to handle additional dependencies
  - Dynamic plugins offer interesting new possibilities
- Great support from the runtime people:

  *Special thanks to Jeff McAffer, Pascal Rapicault and Thomas Watson*

  - They opened up the runtime (for me ;-)
  - Very quick support and assistance
- Next steps:
  - Correct handling of the dynamic features of the runtime in combination with aspects (“run-time-like” weaving)