Aspect Oriented Programming
Jaclyn Chang, Hai Huo, Kolton Lehmann, Jeremy Nauta, Henry Rosvick, Cuong Truong

Objectives
- Compare Aspect Oriented Programming (AOP) with Object Oriented Programming (OOP)
- Understand how AOP works
- Discuss the pros & cons of AOP
- Describe practical applications of AOP

Important Acronyms
- AOP - Aspect Oriented Programming
- OOP - Object Oriented Programming

Roadmap
- AOP vs. OOP
- Benefits of AOP
- Understanding AOP
- AspectJ Tutorial
- Real Applications of AOP
AOP vs. OOP

Aspect-oriented programming does not replace object-oriented programming - it complements it.
Key Terms

Aspect
Cross-cutting concern
Pointcut
Advice

Aspect

A real world object can be represented as a variable, function, data structure or a class

When some objects related to each other (such as a feature) are linked to many other parts of the program, it is called an aspect.

Problem with Aspects

A Possible Solution


Cross-Cutting & Concerns

A concern is a particular set of information that has an effect on the code of a computer program.

Cross-cutting is when action occurs at the same time in two different locations.

What is a Cross-cutting Concern?

Definition:
Parts of a program that rely on or affect many other parts of the system.
Create system interdependencies or coupling -> reduce modularity.

Examples:
Movie
Code: a functionality is needed across many different layers/modules/classes within the application.
Logging Problem

This is a programmer's nightmare

http://www.programcreek.com/2011/08/what-is-aspect-oriented-

Logging Solution

Pointcut

Definition:
Points that happen within the execution code, such as method calls, method executions, etc.

Pointcut vs. Event Trigger

Pointcuts occur when a certain piece of code is called

Similar to:
Event triggers occur when something happens, like a button click

Pointcut vs. Event Trigger

Pointcuts notify advices that a certain event has occurred

Similar to:
Event triggers notify event handlers that a certain event has occurred

Pointcut Logic

TRIGGER [pointcut name]
WHEN [function logic]
is executed

The pointcut [pointcut name] can then be used by advices to execute code
Pointcut Examples
TRIGGER myPointcut WHEN function() is executed
TRIGGER fooOrBar WHEN foo() OR bar() is executed
TRIGGER fooAndBar WHEN foo() AND bar() are executed

Advice
Definition: A section of code that is executed by a pointcut
An advice acts similar to an event handler

Advice Types
Linear
Before - executes before a pointcut
After - executes after a pointcut

Special
After Exception - executes after an exception is thrown
After Finally - executes after a finally occurs
Around - hybrid of "Before" & "After"

Print Example
Method Called: functionA()

@Before & @After in Assembly Code

```
function functionB
  [Assembly code]
  BEFORE functionB
  JUMP to functionB

[Assembly code]
functionB
  AFTER functionB
  RETURN
```

Method Called: functionA();
Example #1
Where does the advice 'before functionB' occur?
A. Point #1
B. Point #3
C. Point #2
Answer: C
Point #2

Example #2
What is the output if we print "After" at the advice 'after functionC'?
A. After Function
B. Function After
Answer: B
Function After

Example #3
Find the text printed for the following advices:
• Print "A" before functionA
• Print "B" before functionB
• Print "C" after functionC
Answer: A
A
B
Function C
C B

Example #4
Find the text printed for the following advices:
• Print "A" before functionA
• Print "B" after functionB
• Print "C" after functionC
Answer: B
A
Function
C
B

AOP Languages/Libraries
Java
 AspectJ, CaesarJ
.NET
 LOOM.NET, Afterthought
Python
 Aspyct, Pyutilities
C / C++
 AspectC++
Objective-C
 AspectCocoa

And many, many more...
AspectJ Tutorial

Real Applications of AOP

logging: keep track of program’s execution history.

Overloading checks: avoid system crashes caused by overload in network transactions

References

Clips From: Quantum of Solace, MGM 2008

*Aspect-Oriented Software Development with Use Cases*
By: Ivar Jacobson; Pan-Wei Ng
Publisher: Addison-Wesley Professional
Pub. Date: December 30, 2004
Found in: Safari Books Online (via UCalgary Online Library)

*Aspect-Oriented Analysis and Design: The Theme Approach*
By: Siobhán Clarke; Elisa Baniassad
Publisher: Addison-Wesley Professional
Pub. Date: March 23, 2005
Found in: Safari Books Online (via UCalgary Online Library)

By: Adrian Colyer; Andy Clement; George Harley; Matthew Webber
Publisher: Addison-Wesley Professional
Pub. Date: December 14, 2004
Found in: Safari Books Online (via UCalgary Online Library)

Software Evolution
Chapter 9: Evolution Issues in Aspect-Oriented Programming
By: Kim Mens, Tom Tourwe
Publisher: Springer Berlin Heidelberg
pages 203 – 232

*Google Guice: Agile Lightweight Dependency Injection Framework*
By: Robbie Venbrux
Publisher: Apress
Copyright 2008
Chapter 4 pages 59 – 69

Aspect-Oriented Programming
By: Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Videira Lopes, Jean-Marc Loingtier, John Irwin
Publisher: Springer-Verlag

The AspectJ™ Programming Guide
By: The AspectJ team
Publisher: Copyright (c) 1998-2001 Xerox Corporation, 2002-2003 Palo Alto Research Center, Incorporated
Abstract: This programming guide describes the AspectJ language. A companion guide describes the tools which are part of the AspectJ development environment.
Found in: Google

An Introduction to AOP
By: Sing Li
Publisher: Sing Li, Wrox Press
Date: Sept 30, 2005
Abstract: This author will guide you through the basic concepts of aspect-oriented programming with some simple, open source examples of applying AOP with AspectJ.
Found in: Google