Lecture 7
Aspect-orientation (AO*)

A new paradigm in Software Engineering

Last lecture and tutorial …
Software Quality Measurements

- We have shown the use of quality measurements to monitor the progress of software development
- The development/restructuring (maintenance) activities (refactoring, tuning, adding features) can be guided by the metrics of softgoals

Today …
On Aspect Orientation

- Today we explain the paradigm of aspect-orientation
  1. Concepts: What are aspects?
  2. Practices: Aspect-orientation at large
     - AOP: Aspect-oriented programming
     - AOSD: Aspect-oriented software development
     - *AORE: Aspect-oriented requirements engineering*
     - AOSR: Aspect-oriented software reuse (probably next lecture)
  3. A case study of AORE
  4. Summary

1. What are aspects?

1. Some design principles
   - Divide and conquer: problem solving/design principle
   - Modularization: high cohesion/low coupling
     Separation of concerns
   - DRY: Don't Repeat Yourself
     Increase the fan-in
2. Previous paradigms
   - 70s – 80s:
     Structured programming (Goto's considered harmful) =>
     Structured Analysis, Structured Design
   - 80s – 90s:
     Object-oriented programming (OOP) =>
     OOA/OOD => UML
3. Why another paradigm ?
   - Since late 90s …
     Separation of the *crosscutting* concerns
4. What are aspects?
   - Modularizing the crosscutting concerns

1.1 Some design principles

Structured programming

- What is structured program?
  - A program has no more GOTO's
  - Only three kinds of structure prevails
    - Sequential
    - If-then-else
    - Loops

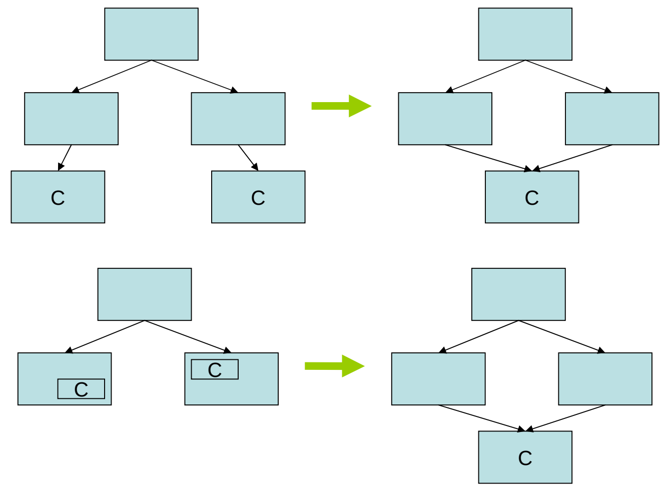
  [Dijkstra: Goto considered harmful]
  - In other words, every statement block has single-entry, single-exit as Hammock Graph

  [Weiser: Program slicing]
- "*Whenever possible, we wish to maximize* ***fan-in*** *during the design process. Fan-in is the raison d'être of modularity. Each instance of multiple fan-in means that some duplicate code has been avoided*." **raison d'être:** grounds for existence (http://www.french-linguistics.co.uk/dictionary/)

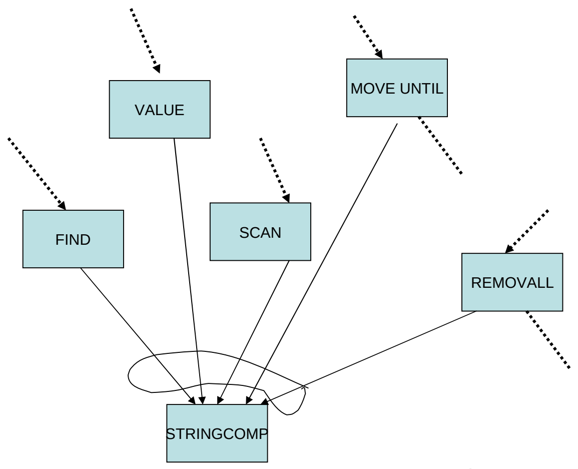
  [Yourdon & Constantine79] Structured Design (pg. 172, see also http://wwwpa.win.tue.nl/wstomv/quotes/structured-design.html)

  [parnas: Modularization, information hiding]

(1) A decomposition hierarchy from abstract to concrete: Divide and Conquer, Structured Design; (2) Don't Repeat Yourself, Factoring / Refactoring …



Example



Yourdon & Constantine, SD, pg.168

1.1 Some design principles

Object-oriented programming

- Everything is an object (Smalltalk)
- Information hiding / Encapsulation: object groups related data and the operations on the data into a module
- Object has structural relationships:
  - inheritance: generalization / specialization: isA/instanceOf
  - aggregation : hasA / isPartOf
  - associations: 1-to-many, 1-to-1, many-to-many
- In the end, the structurally-related objects are *packaged* into components

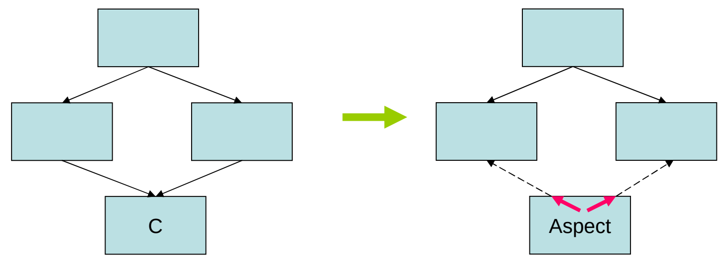
1.2 Aspect-orientation

- Component language (any structured or OO language, even corresponding design and requirements specification)
- What are crosscutting concerns?
- An aspect language
  - What are joinpoints?
  - What are pointcuts?
  - What are advices?
- A weaving mechanism

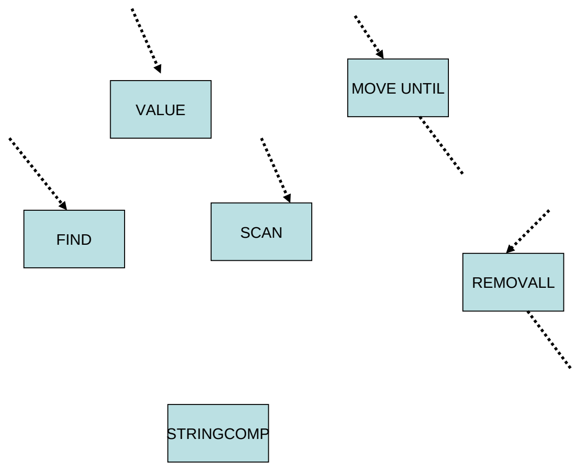
Aspect concepts

- Concepts: cross-cutting, component, aspect, join points, weaving

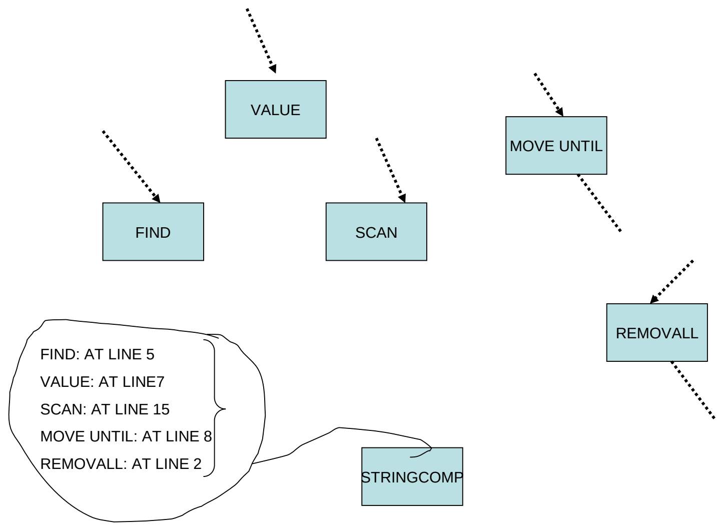
AOP hides the join points



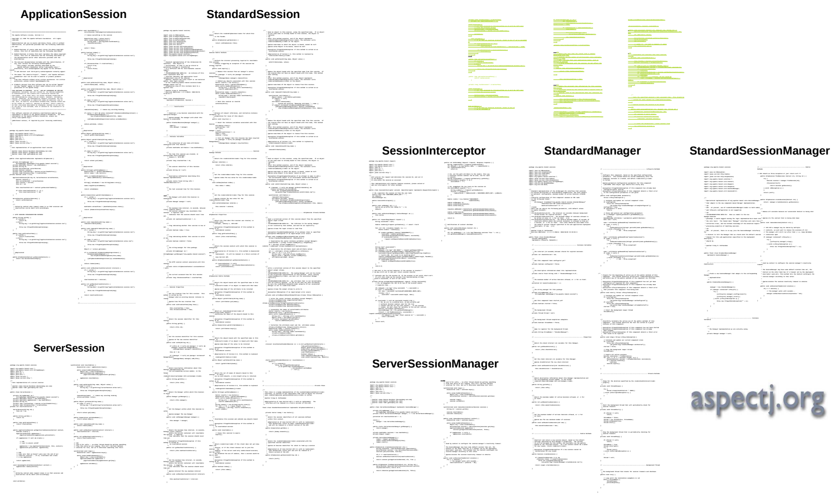
AOP (THE MAGIC)



AOP (NOT REALLY MAGIC)



AOP example



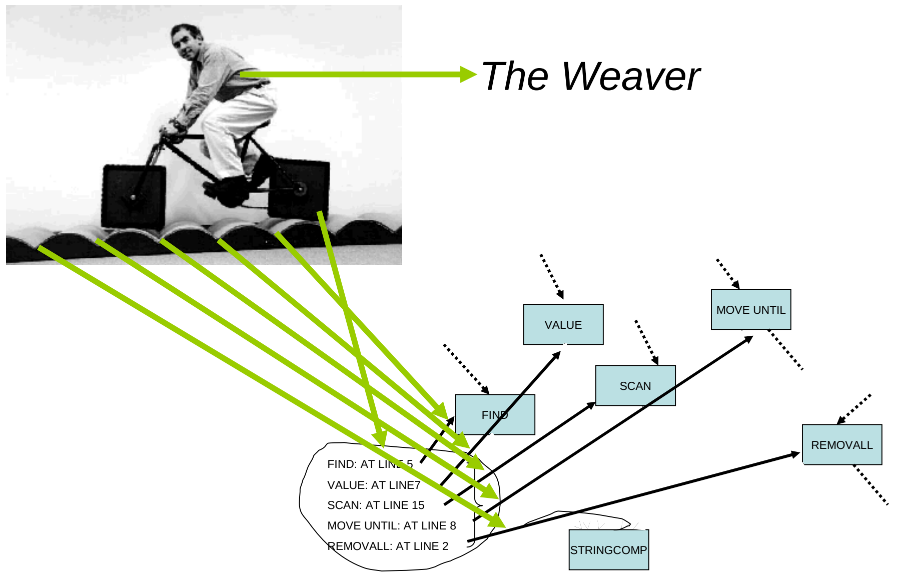
Stan Wagon's bike

My square-wheel bike, on permanent display at Macalester College. This construction, believe it or not, earned me an entry in "Ripley's Believe It or Not"; beats standing in a block of ice for three days or growing three-foot long fingernails.

--

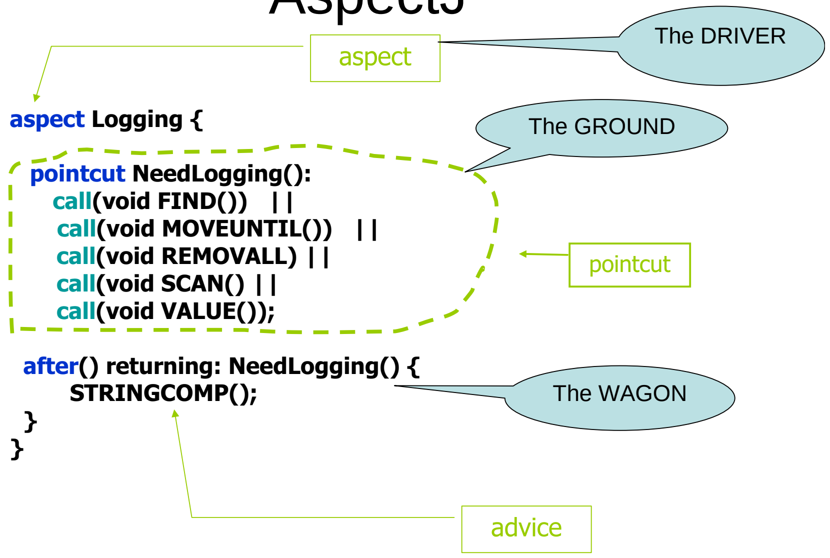
http://www.stanwagon.com

Stan Wagon (wagon@macalester.edu), Prof. of Mathematics and Computer Science, Macalester College, St. Paul, Minnesota



Yourdon & Constantine, SD, pg.168

AspectJ



```
aspect Logging {
  pointcut NeedLogging():
     call(void FIND())   ||
      call(void MOVEUNTIL())   ||
      call(void REMOVALL) ||
      call(void SCAN() ||
      call(void VALUE());

  after() returning: NeedLogging() {
      STRINGCOMP();
  }
}
```

2. Aspect-orientation at large

2.1 Aspect-oriented Programming

- It permeates into almost every popular high-level programming languages
- Java
  Hyper/J, *AspectJ*, *AJDT*, *JBoss*
- C/C++/C#
  AspectC/C++, *C#*
- PHP
  AOPHP, *AspectPHP*

  … and many many more: see AOSD.NET

Every AOP mechanism has to support

- Definition and representation of aspects
  - Definition of Advices in the component language
  - Definition of Joinpoints in regular expressions
    - *Optionally, they can introduce new data members, changing the structures of components*
  - *Representation: New keywords, New directives, XML, but never change the code of components directly*
- Implementing a weaver
  - As preprocessor => generates woven components in the component language (AspectC, AOPHP)
  - As instrumenting compiler => generates woven components in the bytecode for the languages supporting reflection (AspectJ)
  - As interpretator => interpreting the woven code on-the-fly (AspectPHP)

2. Aspect-orientation at large

2.2 Aspect-Oriented SD

- AO includes the whole lifecycle of SE
  - http://www.aosd.net
- There is a conference AOSD
- There are workshops on Early Aspects at AOSD, OOPSLA, ICSE
- Hot topics related to all other SD technologies
  - Aspect-oriented Refactoring
  - Aspect Mining
  - Aspect-oriented Debugging
  - Aspect-oriented Testing
  - Aspect-oriented Slicing
  - Aspect-oriented Model Checking

…

2. Aspect-orientation at large

2.3 Aspect-Oriented RE

- Lessons learnt from success stories
  - SP => SA
  - OOP => OOA
  - Why not AOP => AOA?
    - Separation of crosscutting concerns earlier
    - Avoid duplication as early as possible
    - Identify aspects before mining them from code
- Discover aspects in the early requrements
  - From structured requirement documents
  - From unstructured (textual) documents
- Verify discovered (candidate) aspects in AOP

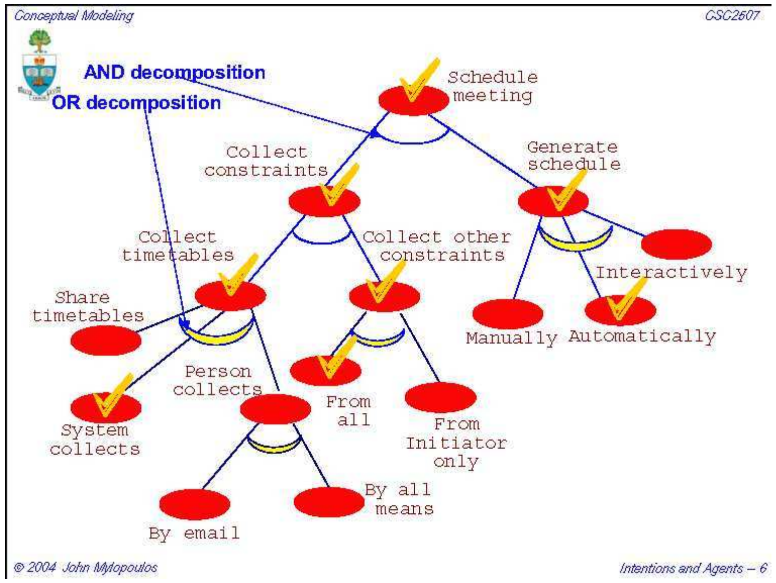
3. A Case Study on AORE

1. Quickly go through goal-oriented requirements engineering basics
2. A requirements engineering process to elicit early aspects (goal aspects)
3. A reverse engineering exercise to identify candidate aspects (code aspects)
4. Linking goal aspects with code aspects

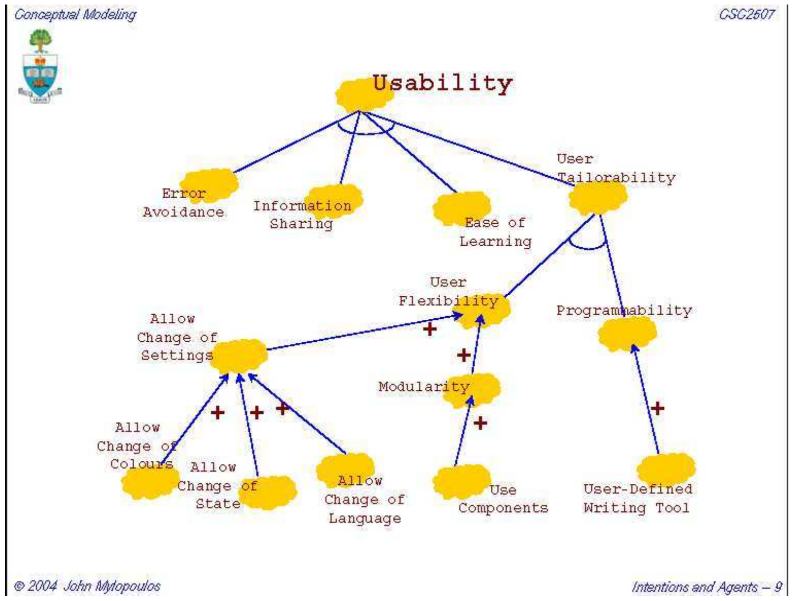
3.1 Requirements Goal Models

- A goal model is an intentional model
- A goal can be decomposed into AND or OR subgoals
- A goal model has both hard and soft goals
  - A hard goal can be either satisfied or denied
  - A soft goal is partially satisfied => *satisficed*
- Soft goal uses HELP (+), HURT (-), MAKE (++) or BREAK (--) correlations to show partial satisfaction (satisfice) from a set of subgoals

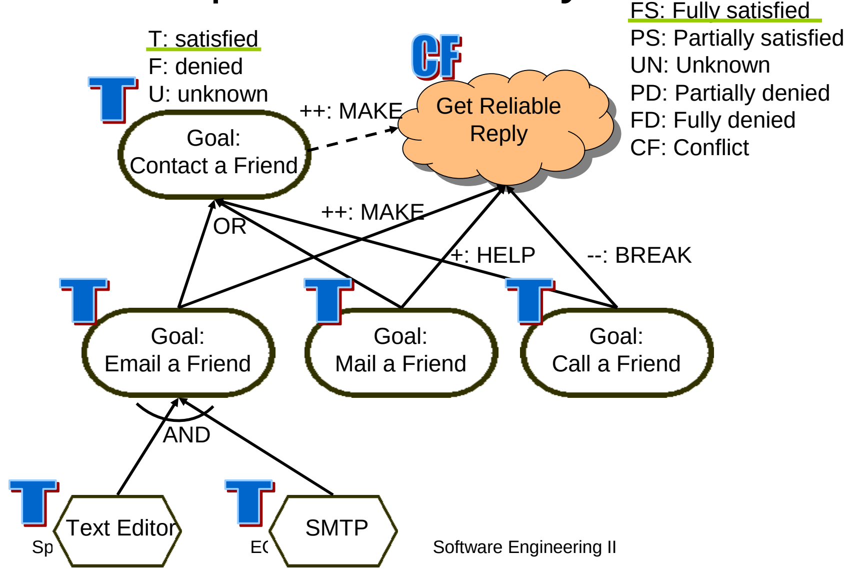
3.1.1 Hard goal model

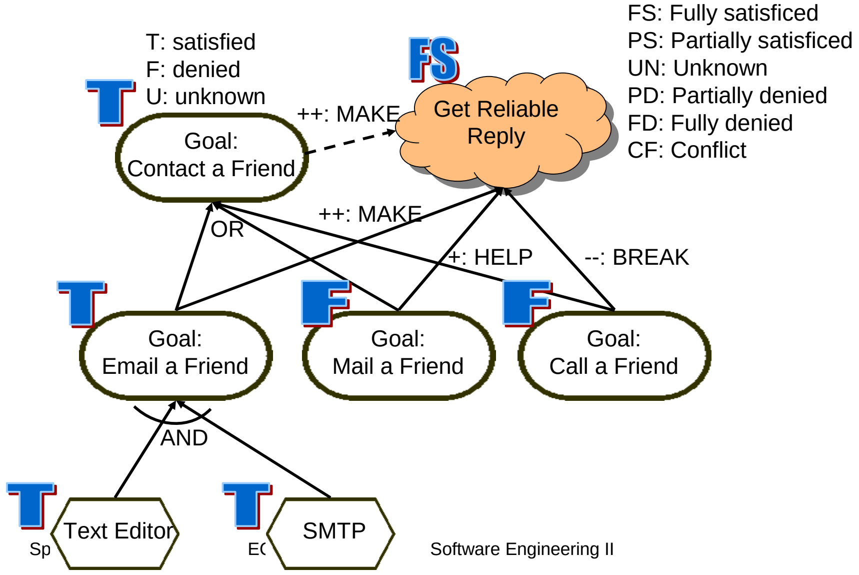


3.1.2 Soft goal model



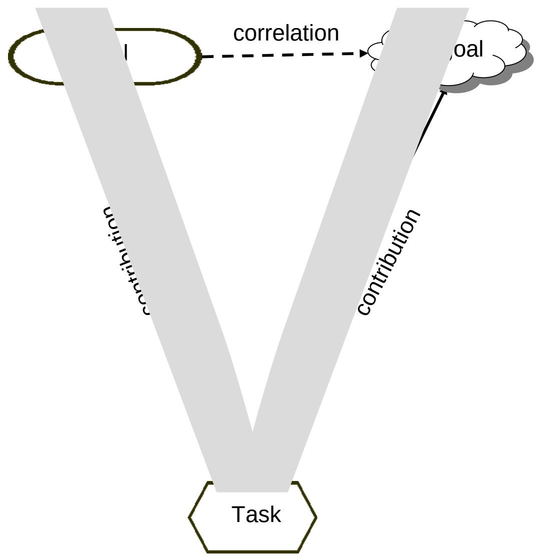
3.1.3 Goal-Oriented Requirements Analysis





3.1.4 V-graph

In order to reason about interplay of functional and non-functional requirements, we create a particular type of goal model, called *V-graph*



3.2 The Process

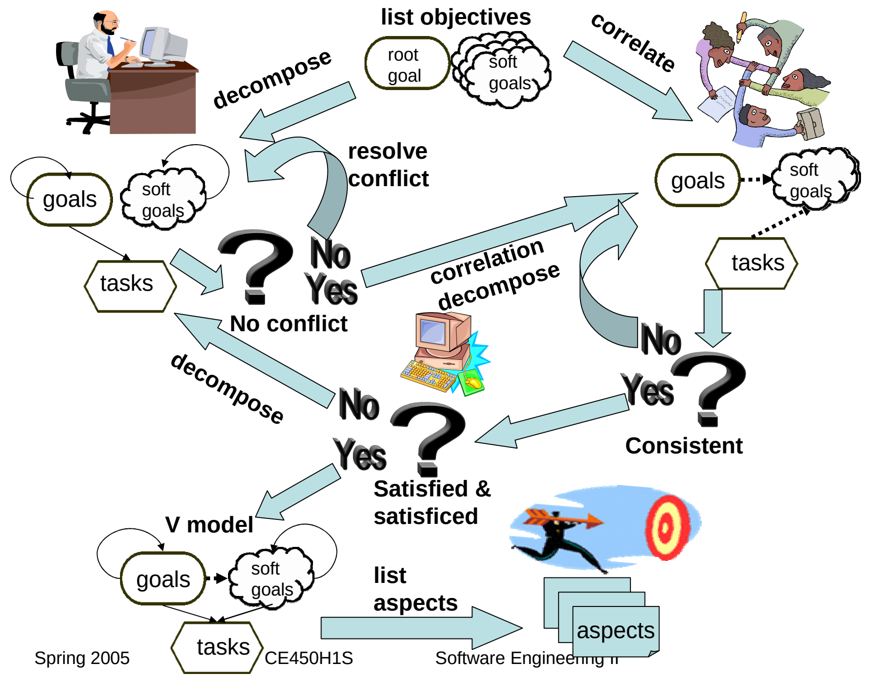
– *Start from root-level goals and soft goals, correlate and decompose them into a V-graph*

– *A goal analysis based on the label propagation algorithm is used to check for:*

  - *Conflicts*
  - *Inconsistencies*
  - *Denial of any goal or soft goals*

– *After resolving the problems, a proper V-graph is obtained*

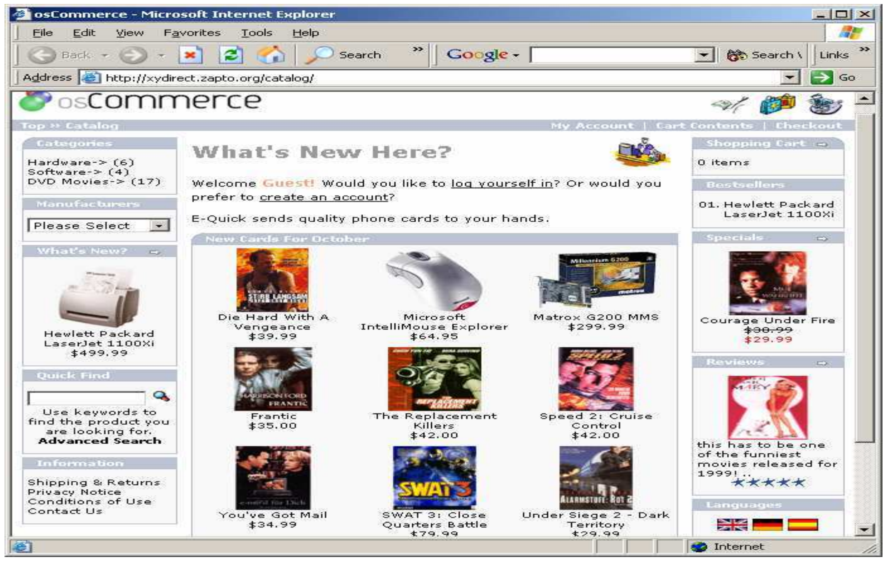
– *Then we list the candidate aspects from the V-graph*



3.3 A Case Study

- Medi@Shop adapted from literature:
  Castro, Kolp, Mylopoulos, Towards requirements-driven information systems engineering: the Tropos project, Journal of Information Systems, 2002.
  Can we find aspects from early requirements?
- osCommerce studied from an LAMP (Linux, Apache, MySQL, PHP) Open-Source project: (http://www.oscommerce.com)
  Do they manifest in the developed software?

osCommerce (version 2.2m2)


osCommerce - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Google
Search
Links
Address
Go
Internet

http://xydirect.zapto.org/catalog/

osCommerce

Top » Catalog | My Account | Cart Contents | Checkout

Categories
Hardware-> (6)
Software-> (4)
DVD Movies-> (17)

Manufacturers
Please Select

What's New?
Hewlett Packard LaserJet 1100Xi
$499.99

Quick Find
Use keywords to find the product you are looking for.
Advanced Search

Information
Shipping & Returns
Privacy Notice
Conditions of Use
Contact Us

What's New Here?

Welcome Guest! Would you like to log yourself in? Or would you prefer to create an account?

E-Quick sends quality phone cards to your hands.

New Cards For October

Die Hard With A Vengeance $39.99	Microsoft IntelliMouse Explorer $64.95	Matrox G200 MMS $299.99
Frantic $35.00	The Replacement Killers $42.00	Speed 2: Cruise Control $42.00
You've Got Mail $34.99	SWAT 3: Close Quarters Battle $79.99	Under Siege 2 - Dark Territory $29.99

Shopping Cart
0 items

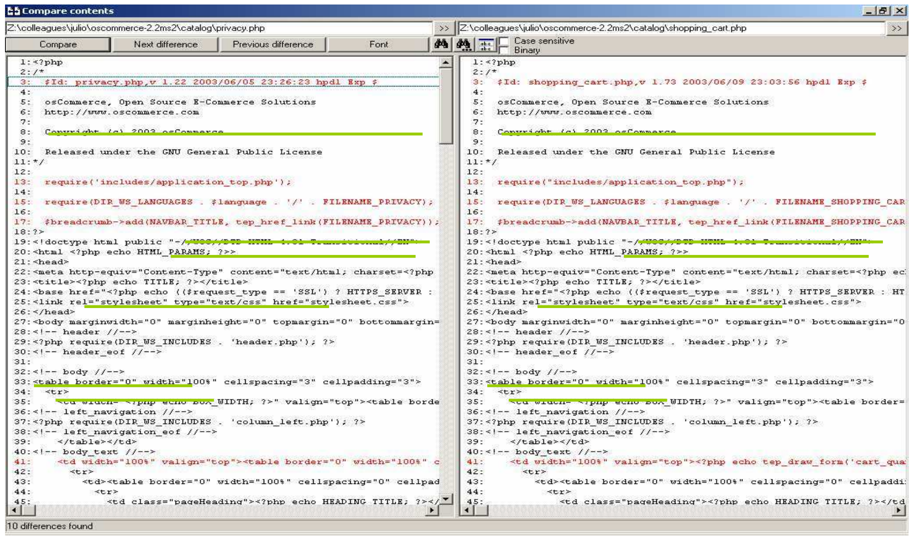
Bestsellers
01. Hewlett Packard LaserJet 1100Xi

Specials
Courage Under Fire
~~$38.99~~
$29.99

Reviews
this has to be one of the funniest movies released for 1999! ..
★★★★★

Languages

Duplications in code



Compare contents

Z:\colleagues\julio\oscommerce-2.2ms2\catalog\privacy.php

```
<?php
/*
  $Id: privacy.php,v 1.22 2003/06/05 23:26:23 hpdl Exp $

  osCommerce, Open Source E-Commerce Solutions
  http://www.oscommerce.com

  Copyright (c) 2003 osCommerce

  Released under the GNU General Public License
*/

  require('includes/application_top.php');

  require(DIR_WS_LANGUAGES . $language . '/' . FILENAME_PRIVACY);

  $breadcrumb->add(NAVBAR_TITLE, tep_href_link(FILENAME_PRIVACY));
?>
<!doctype html public "-//W3C//DTD HTML 4.01 Transitional//EN">
<html <?php echo HTML_PARAMS; ?>>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=<?php
<title><?php echo TITLE; ?></title>
<base href="<?php echo (($request_type == 'SSL') ? HTTPS_SERVER :
<link rel="stylesheet" type="text/css" href="stylesheet.css">
</head>
<body marginwidth="0" marginheight="0" topmargin="0" bottommargin=
<!-- header //-->
<?php require(DIR_WS_INCLUDES . 'header.php'); ?>
<!-- header_eof //-->

<!-- body //-->
<table border="0" width="100%" cellspacing="3" cellpadding="3">
  <tr>
    <td width="<?php echo BOX_WIDTH; ?>" valign="top"><table borde
<!-- left_navigation //-->
<?php require(DIR_WS_INCLUDES . 'column_left.php'); ?>
<!-- left_navigation_eof //-->
    </table></td>
<!-- body_text //-->
    <td width="100%" valign="top"><table border="0" width="100%" c
      <tr>
        <td><table border="0" width="100%" cellspacing="0" cellpad
          <tr>
            <td class="pageHeading"><?php echo HEADING_TITLE; ?></
```

Z:\colleagues\julio\oscommerce-2.2ms2\catalog\shopping_cart.php

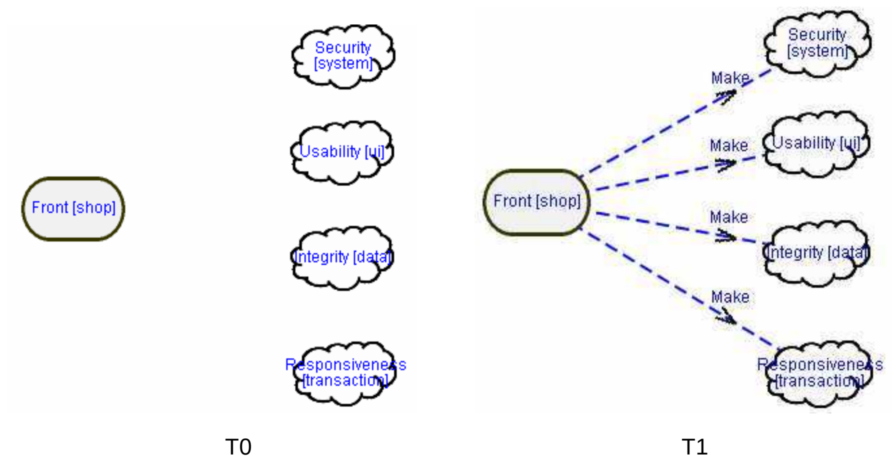
```
<?php
/*
  $Id: shopping_cart.php,v 1.73 2003/06/09 23:03:56 hpdl Exp $

  osCommerce, Open Source E-Commerce Solutions
  http://www.oscommerce.com

  Copyright (c) 2003 osCommerce

  Released under the GNU General Public License
*/

  require("includes/application_top.php");

  require(DIR_WS_LANGUAGES . $language . '/' . FILENAME_SHOPPING_CAR

  $breadcrumb->add(NAVBAR_TITLE, tep_href_link(FILENAME_SHOPPING_CAR
?>
<!doctype html public "-//W3C//DTD HTML 4.01 Transitional//EN">
<html <?php echo HTML_PARAMS; ?>>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=<?php ec
<title><?php echo TITLE; ?></title>
<base href="<?php echo (($request_type == 'SSL') ? HTTPS_SERVER : HT
<link rel="stylesheet" type="text/css" href="stylesheet.css">
</head>
<body marginwidth="0" marginheight="0" topmargin="0" bottommargin="0
<!-- header //-->
<?php require(DIR_WS_INCLUDES . 'header.php'); ?>
<!-- header_eof //-->

<!-- body //-->
<table border="0" width="100%" cellspacing="3" cellpadding="3">
  <tr>
    <td width="<?php echo BOX_WIDTH; ?>" valign="top"><table border=
<!-- left_navigation //-->
<?php require(DIR_WS_INCLUDES . 'column_left.php'); ?>
<!-- left_navigation_eof //-->
    </table></td>
<!-- body_text //-->
    <td width="100%" valign="top"><?php echo tep_draw_form('cart_qua
      <tr>
        <td><table border="0" width="100%" cellspacing="0" cellpaddi
          <tr>
            <td class="pageHeading"><?php echo HEADING_TITLE; ?></td
```

10 differences found

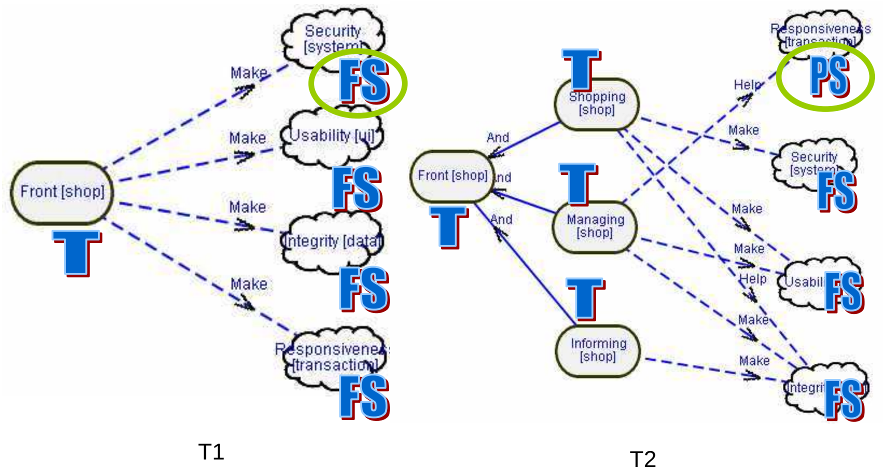
Candidate code aspects in the code Clone detection (by Semantic Design,Inc)

LOC	#clones	Code description	Need refactoring?
1	319	require($path . $file);	No
1	260	echo $expression;	No
559	2	class email;	No
2	292	define ($variable, $value);	No
76	2	class mime;	No
4	67	messageStack->add ($error);	Yes (NFR)
15	15	Postal code zone check	Yes (FR)
22	10	require(application_top.php); SSL check	Yes (FR/NFR)
3	64	Set HTML head CHARSET	Yes (NFR)

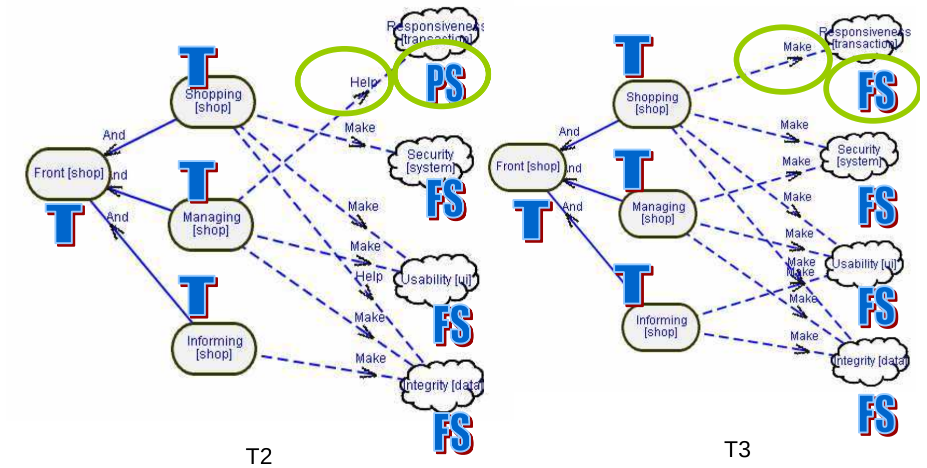
3.4 Identifying goal aspects Correlate initial goals and softgoals



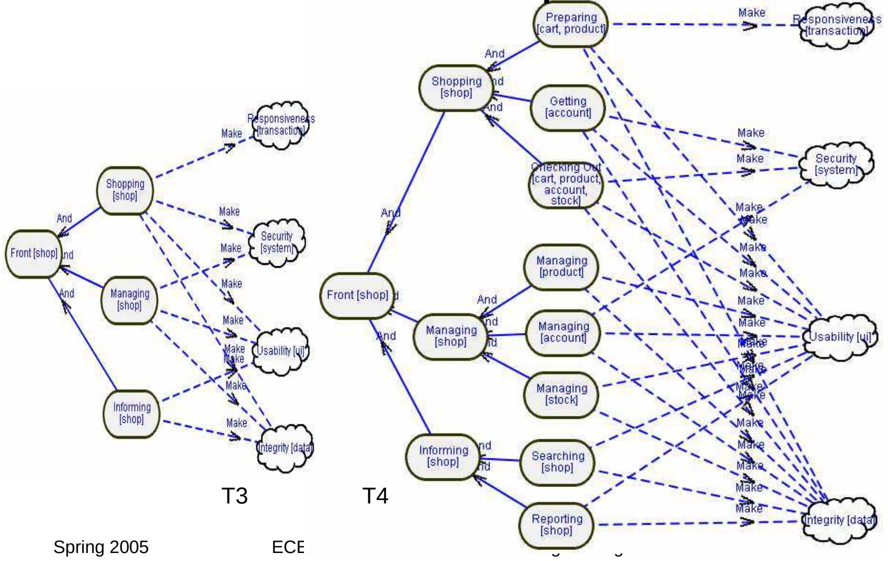
Inconsistent decomposition

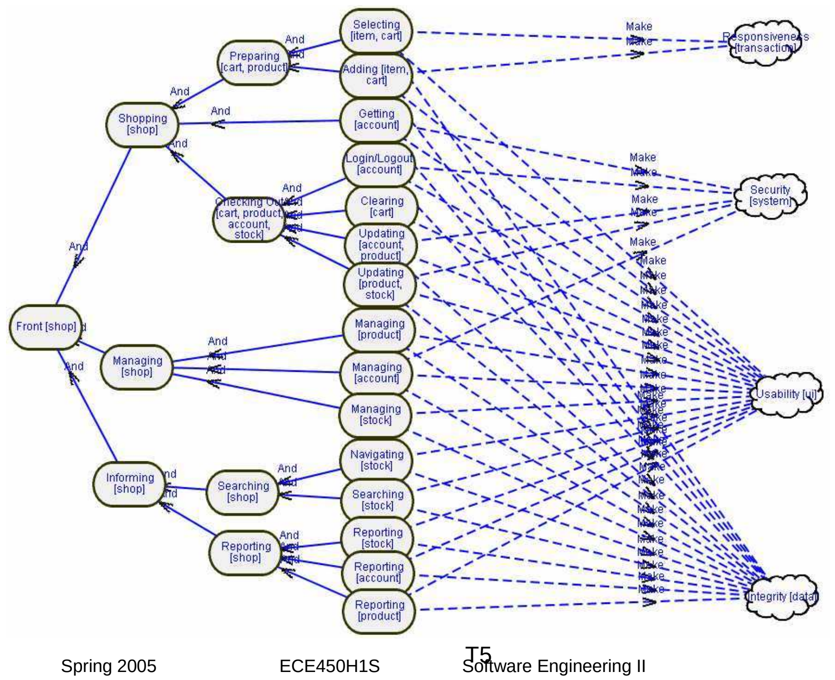


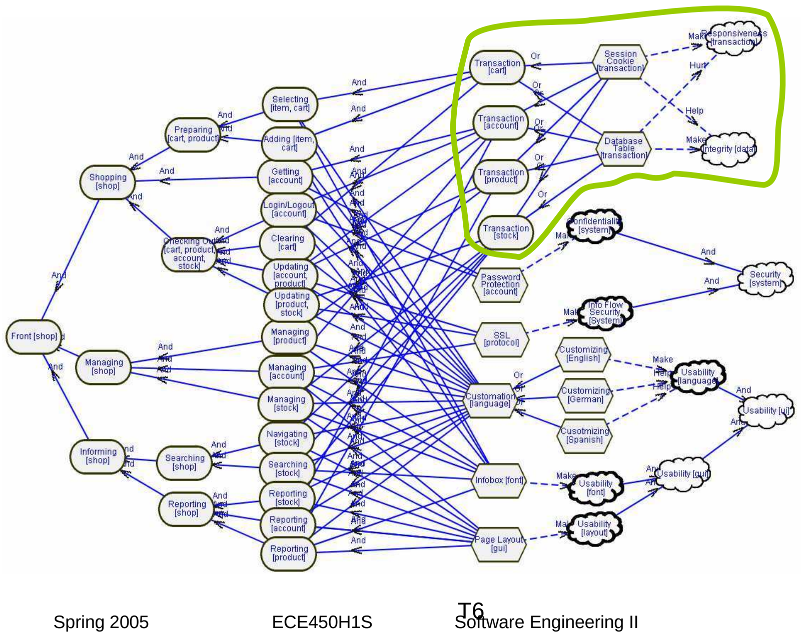
Resolving inconsistency



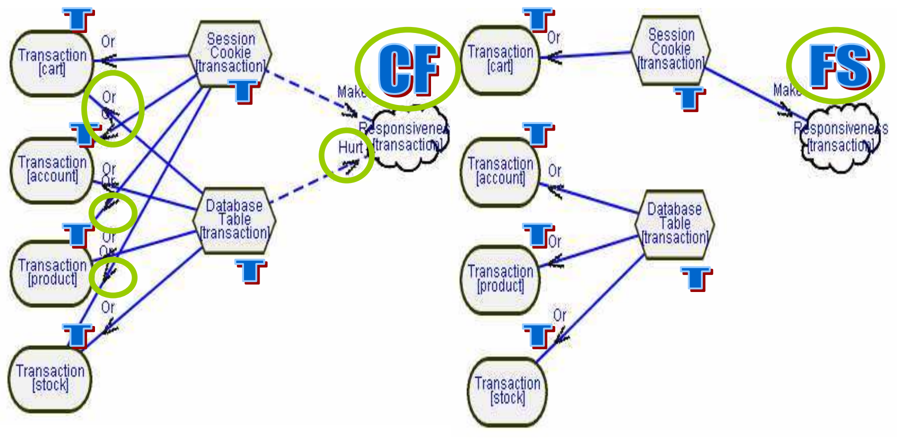
Further decomposition





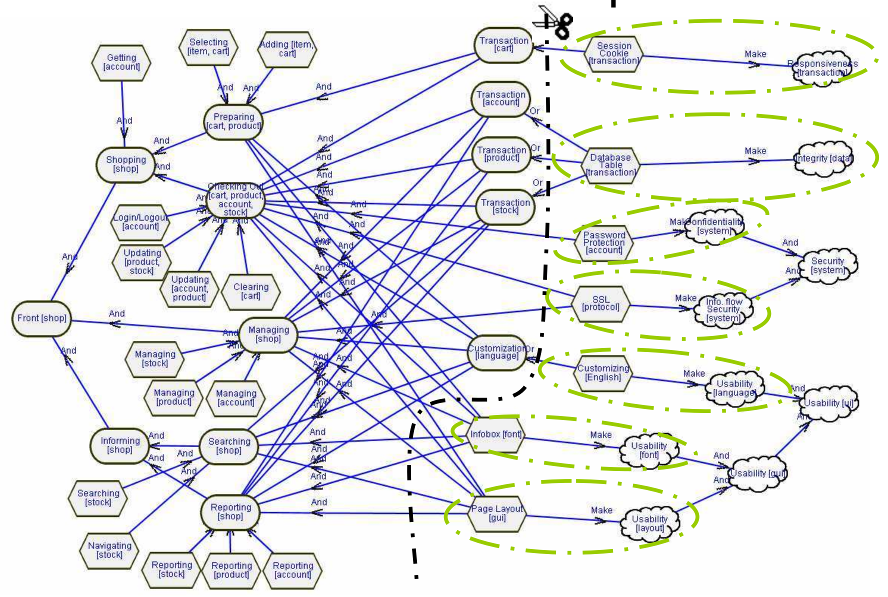


Resolving Conflicts



T6 T7

Result candidate aspects



Goal Aspects

goal aspect Responsiveness[transaction] {
pointcut transaction():
Preparing[cart,product]) ||
CheckingOut[cart, product, account, stock]);
required () **by**: transaction() {
SessionCookie[transaction]();
}
};

- AspectJ-like syntax
- Allow weaving the operationalized tasks with goals specified in the pointcut

Your exercise

- Reverse Engineering
  Identify some aspects in the OpenOME
  - Clone-detection or Callgraph extraction
  - Goal analysis
- Forward Engineering
  - Implement some new NFR through AspectJ

4. Summary

- The concepts of aspect-orientation
- The practise of AOP, AOSD, AORE, AOSR
- A Case study of AORE

Further readings

[AOP] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. “Aspect oriented programming”. *LNCS*, 1241:220--242, Oct. 1997.

[AORefactoring] C. Zhang, H.-A. Jacobsen. “Refactoring Middleware with Aspects”. TPDS 14(1):1058-1073. 2003

[AOMining] C. Zhang, H.-A. Jacobsen. “PRISM is research in Aspect Mining”. OOPSLA, 2004.

[AORE] Y. Yu, J.C. Leite, J. Mylopoulos. “From goals to aspects: discovering aspects from goal models”. *RE’04*, 2004.

What’s next …

- A tutorial on aspect-oriented programming tools
  - AspectJ
  - Eclipse/AJDT
  - Visualizing Aspects
  - Aspect mining tool
- A lecture on (aspect-oriented) Software Reuse
  - Q7 in the OpenOME