

Lecture 2

Software Re-engineering

Some material is based on the CSER projects at U of T
Covers almost all concepts of the course
Detail explanations to come ...

Copyright © Yijun Yu, 2005

Last lecture ...

General Information

- Instructor: Yijun Yu yijun@cs.toronto.edu
- Office: BA7200 (Bahen Center, 7th floor),
946-8530
- Office hours: Wed 5pm – 6pm, Fri 2pm-3pm
- TA: Alexia Giannoula alexia@comm.utoronto.ca
Clark Merchant Clark.Merchant@utoronto.ca
Mazen Almaoui mazen@dsp.utoronto.ca
- Class homepage:
<http://www.cs.toronto.edu/~yijun/ece450h>

Marking Scheme adjusted

- No midterm
- Final Exam 50% (Exam week)
- Course Project 50%
 - Assignment 1 (15%): Feb 11
 - Assignment 2 (15%): Feb 25
 - Assignment 3 (20%): April 8

Our Course Project

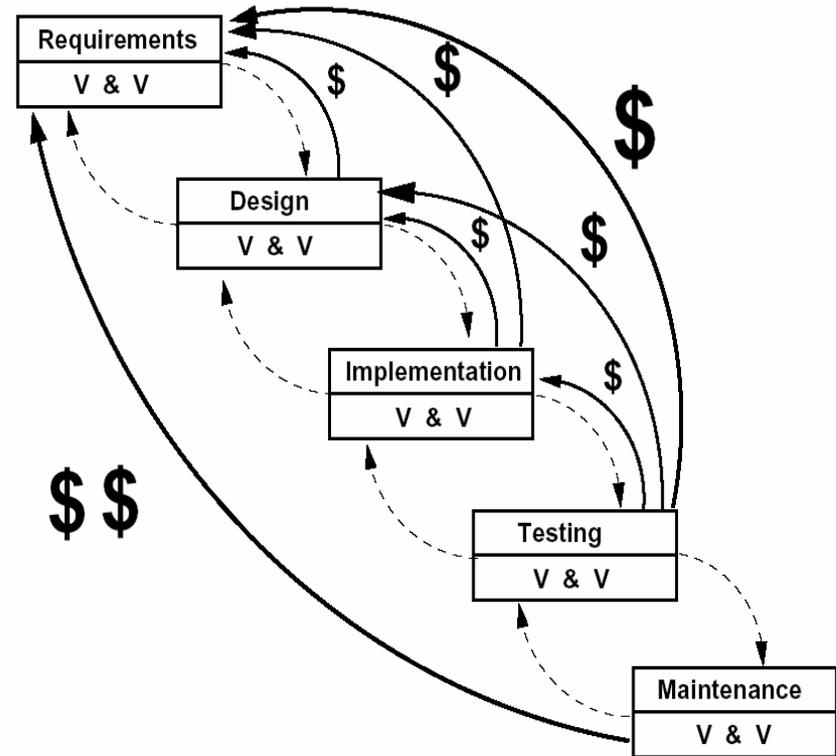
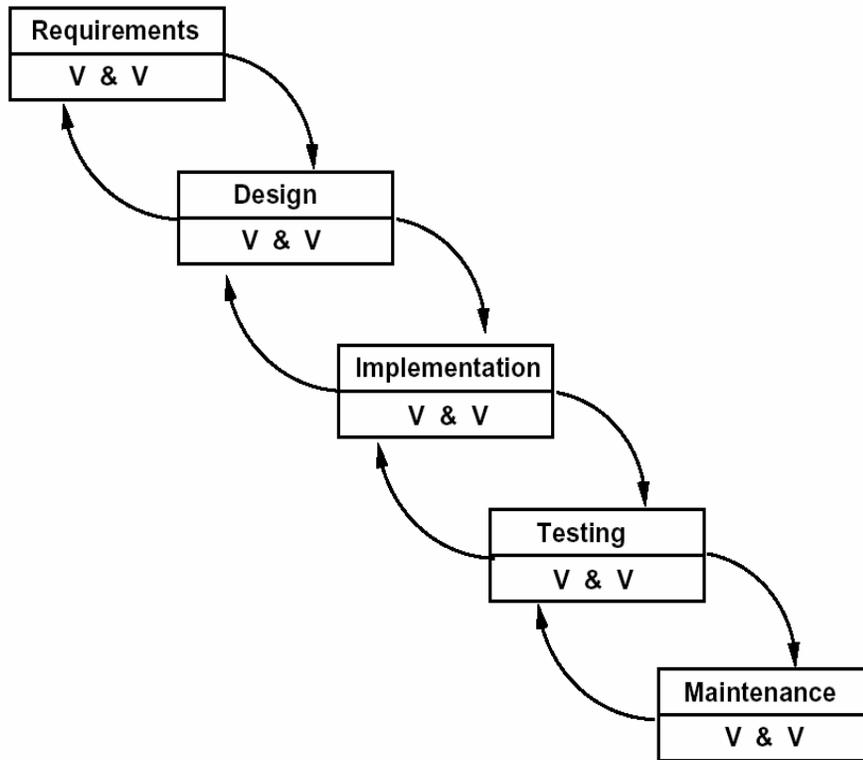
- This is a “brand-new” software reengineering project, emphasizing on reusing, restructuring, refactoring large-scale software systems, and team work !
 - A1: Understanding the architecture of a legacy system (OpenOME, OmniEditor) (15%)
 - A2: Design OmniGraphEditor web service (15%)
 - A3: Reengineering OpenOME to use OmniGraphEditor web service of other teams (20%)
- Tutorials will cover detailed approaches and tools to help you with the project

Today ...

1. Review SE process
2. Discuss Reengineering Concepts
3. Go over some case studies, a road map to our lectures and tutorials:
 - VIM*: componentization, reveal architectures
 - osCommerce*: aspect elicitation, reveal requirements
 - SquirrelMail*: goal elicitation from refactored code
4. Your exercise is to use the learnt knowledge to study two other legacy software systems:
 - OpenOME* and *OmniEditor*
5. Summary

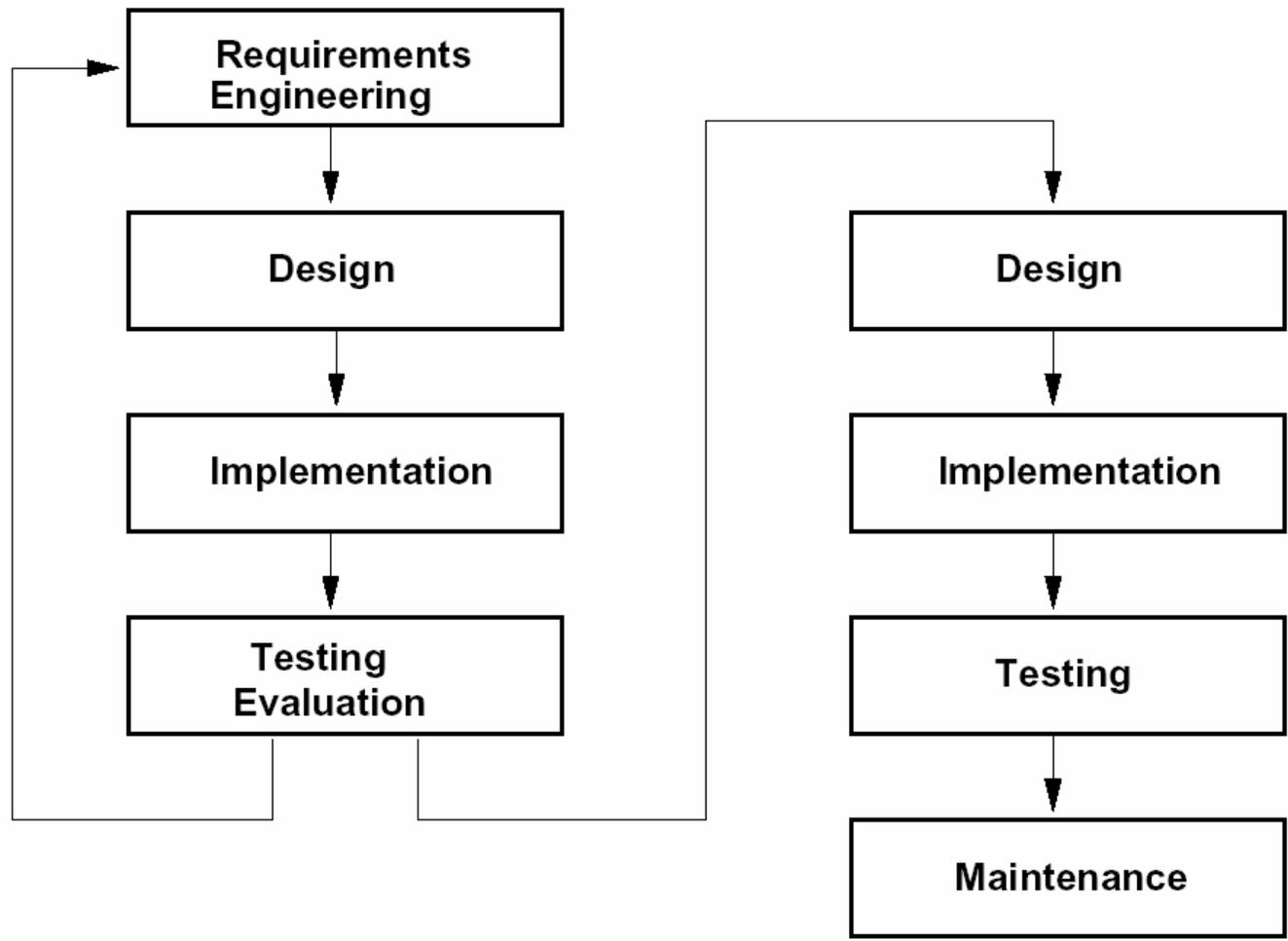
1. Software Engineering Process

The Waterfall process model



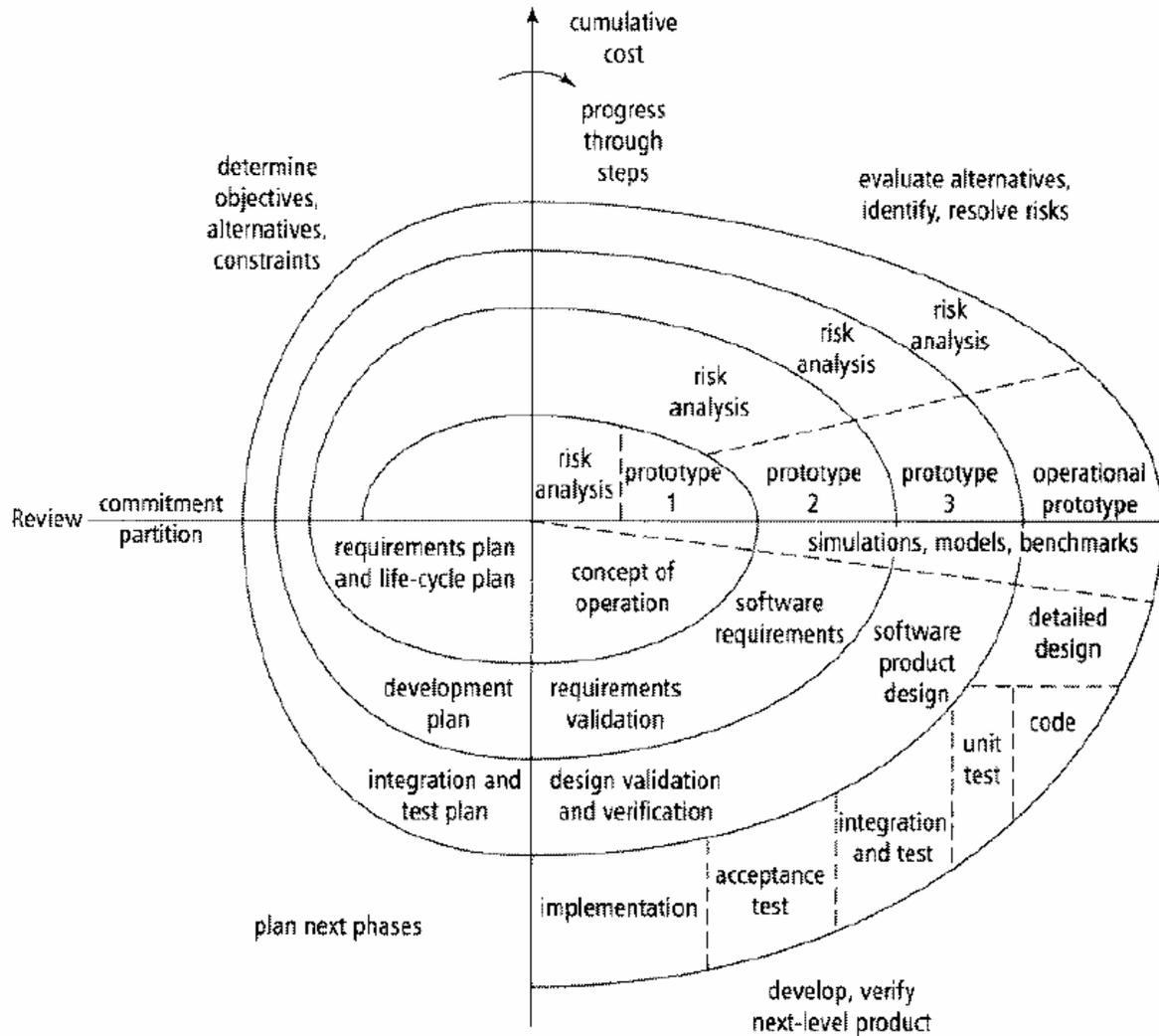
1. Software Engineering Process

Rapid Prototyping process



1. Software Engineering Process

Spiral (incremental) process



2. Reengineering concepts

- Why Software Reengineering?
 - Legacy software are increasing
(Software vs. Hardware)
 - New technology appearing
(Moore's law)
 - Successful ratio of projects increasing
(IBM internal history)
 - Companies are more competing
(now we have the "open-source" movement and free-software foundation)
 - Quality attributes are demanding
(That's the selling point)
 - People are changing
(developers joining and leaving, customers are changing)
 - Software maintenance are pressing
(Largest cost in software development lifecycle >60%)

2. Reengineering concepts

What is software reengineering?

To a large extent, it involves maintenance activities:

- Understanding (predictive)
- Repairing (corrective)
- Improving (perfective)
- Evolving (adaptive)
- Related topics
 - Quality-driven software engineering (-ilities, quality attributes)
 - Requirements engineering (goals, non-functional requirements)
 - Software architectures (architectural views: components, statecharts, features, ...)
 - Model-driven development (MOF, UML, EMF)
 - Design patterns (structural, behavioural)
 - Software refactoring (the code smells)
 - Performance tuning (trade-offs, multi-criteria optimizations)
 - Paradigms: Object-oriented, Goal-oriented, Agent-oriented, Aspect-oriented...

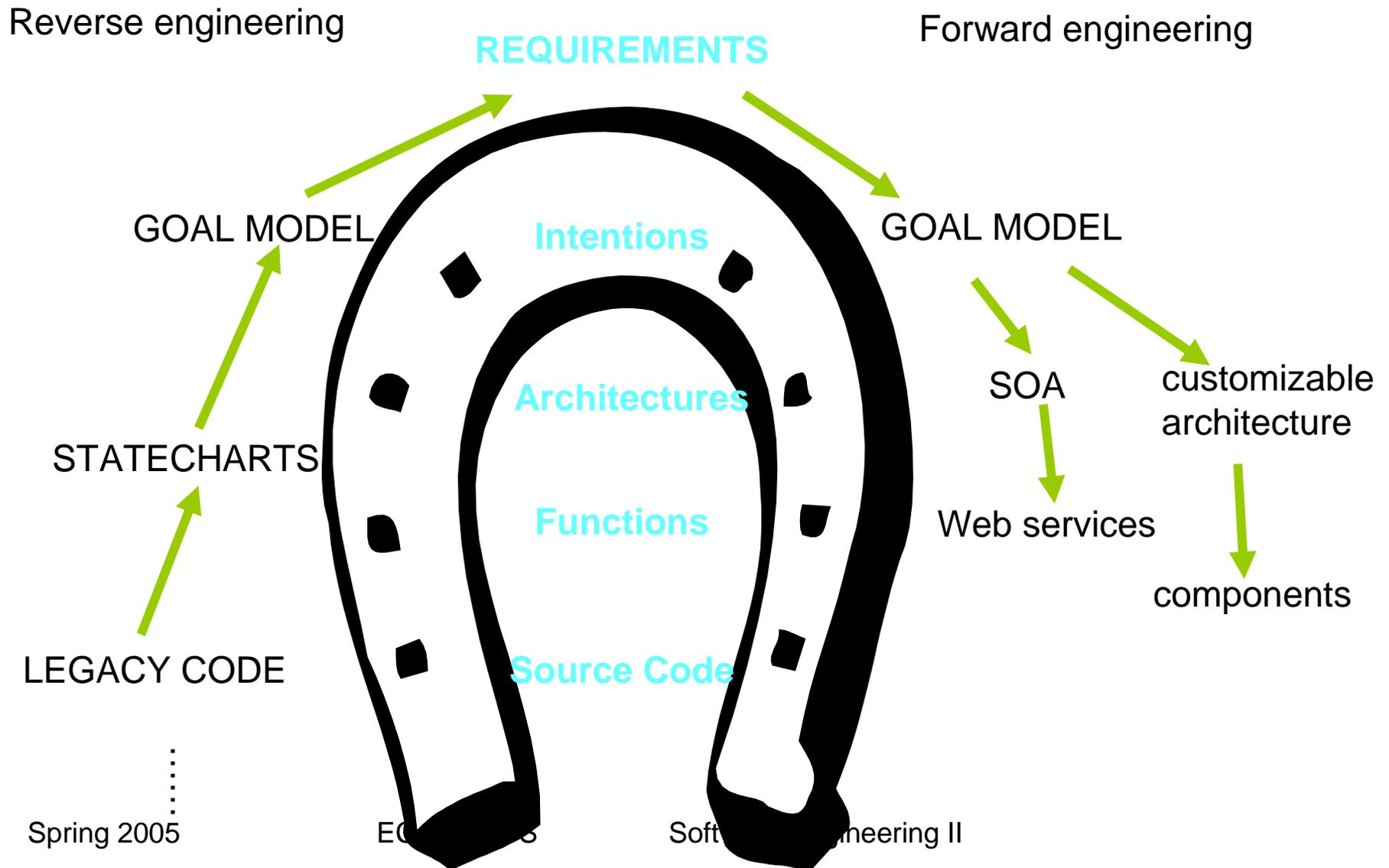
2. Reengineering concepts

The Horseshoe model



Niagara Falls

The Reengineering Horseshoe



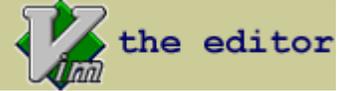
Reading assignments on software architectures

- Previous lecture note for ECE450H1S:
“What is software architecture?”
“How to represent it?”
 - D. Penny. “Introduction to software architecture”:
<http://www.cs.toronto.edu/~chechik/courses00/ece450/lectures/penny.2up.pdf>
 - M. Chechnik. “ADL and Darwin”.
<http://www.cs.toronto.edu/~chechik/courses00/ece450/lectures/Marsha-Darwin.pdf>

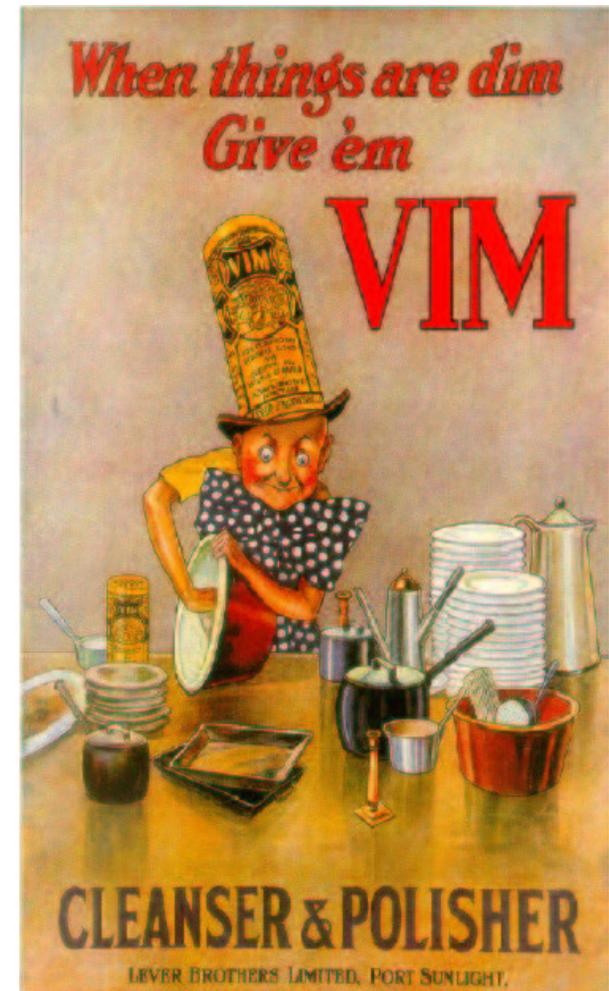
Further readings

- Martin Fowler. “The Refactoring homepage”: <http://www.refactoring.com/>
- CMU SEI: “Software architecture”.
http://www.sei.cmu.edu/ata/ata_init.html
- KMLab. “On goal oriented software engineering”.
http://www.cs.utoronto.ca/km/goal_oriented

Case Study I. VIM

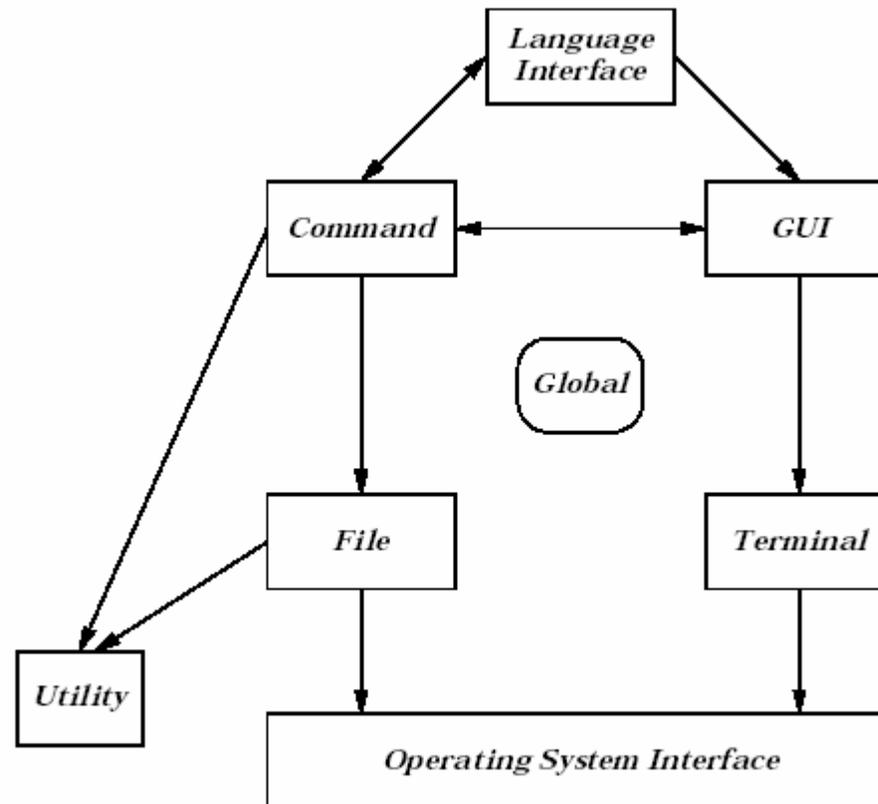


- VIM stands for Vi-IMproved
<http://www.vim.org>
- Are you a VIMer?
- Current version 6.3
- Bram Moolenaar
- Developed in C
- 172 KLOC



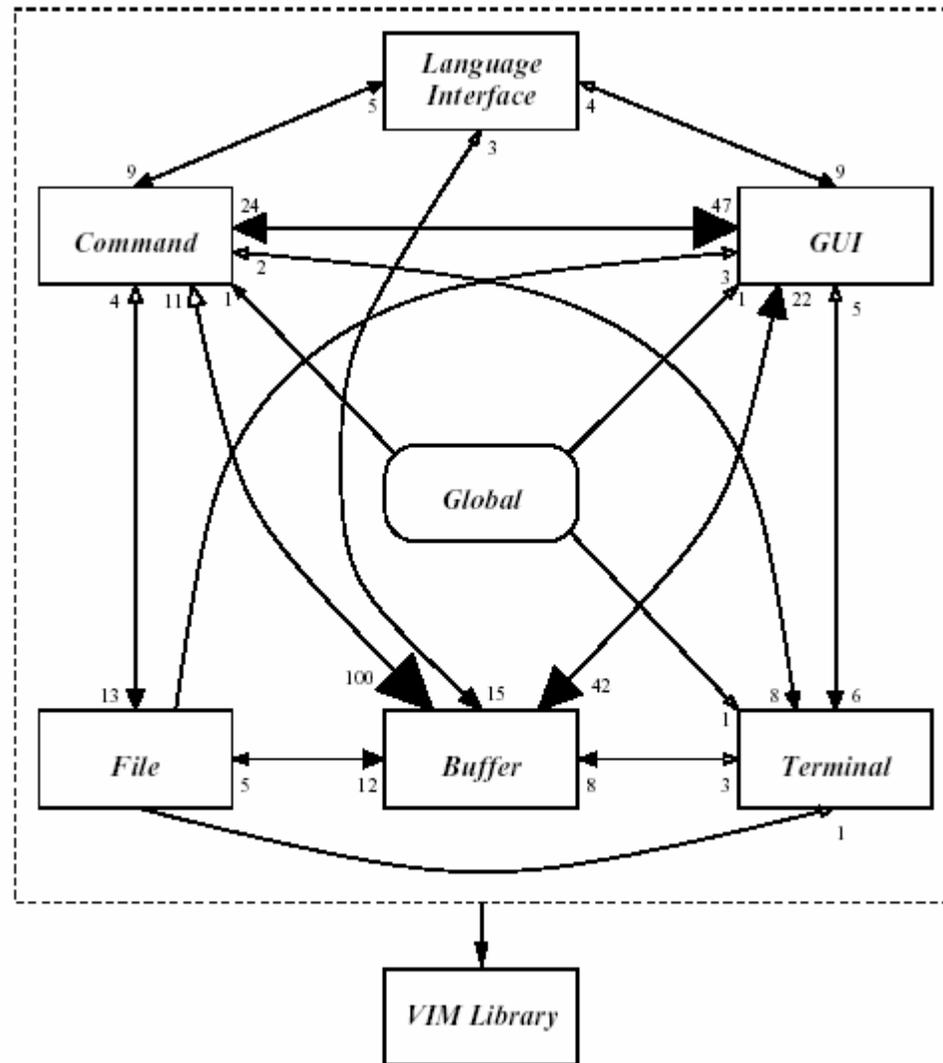
Understanding the architecture of VIM

- Lee's initial VIM architecture



John Tran et al. "Architectural Repair of Open Source Software". IWPC 2000.

- Vim 5.3

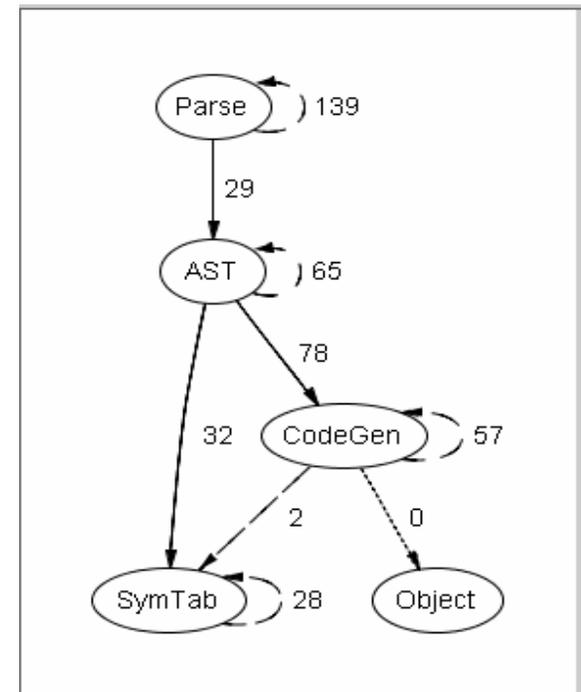


G. Murphy, et al. "Software Reflexion Models: Bridging the gap between design and implementation", IEEE Trans. On Software Engineering 27(4):364-380, 2001.

- Reflexion model (jRMTool)

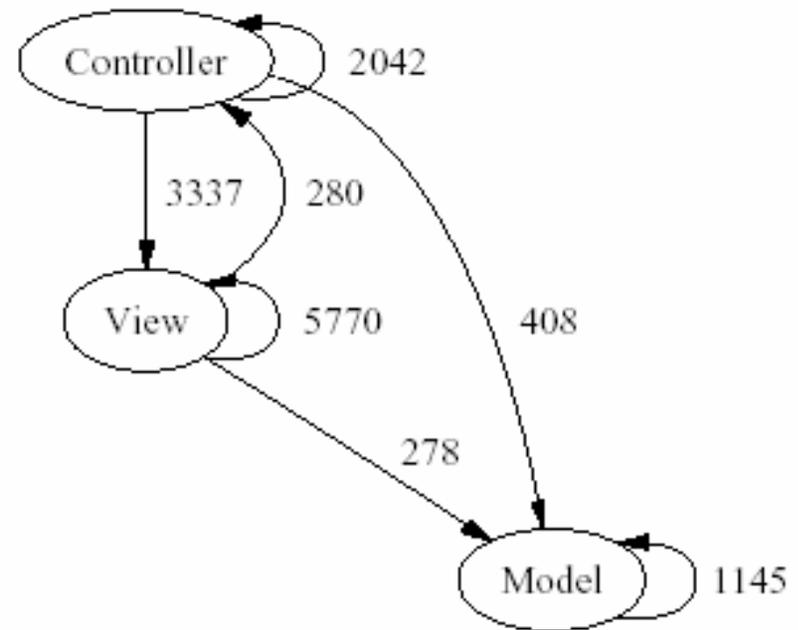
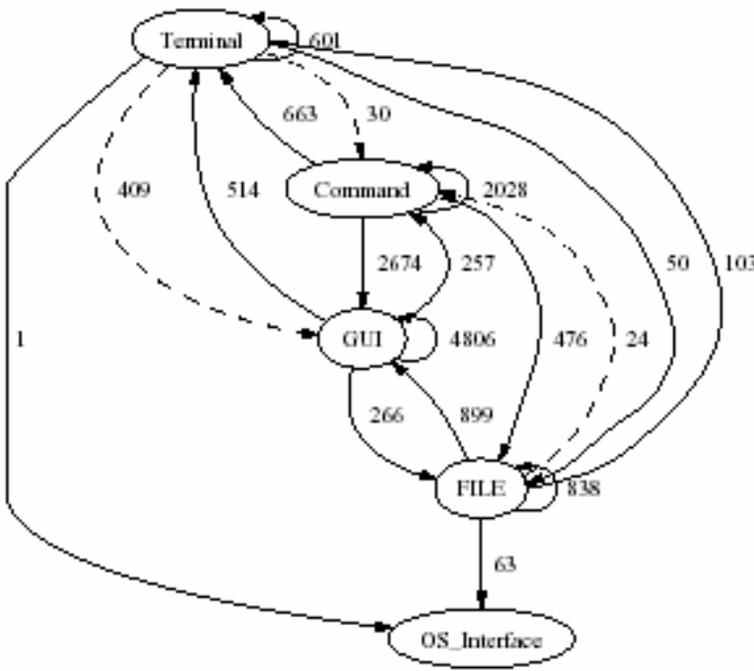


- High-level model (HLM) multi-graph
- Source model (SM) multi-graph (source code or trace)
- Mapping from SM to HLM is defined by regular expressions
- Identify three kinds of edges:
 - Convergence 
 - Divergence 
 - Absence 



H. Dayani-Fard, Y. Yu, J. Mylopoulos, P. Andritsos. “*Improving the build architecture of legacy C/C++ software systems*”, Fundamental Approaches to Software Engineering, April 2005. to appear

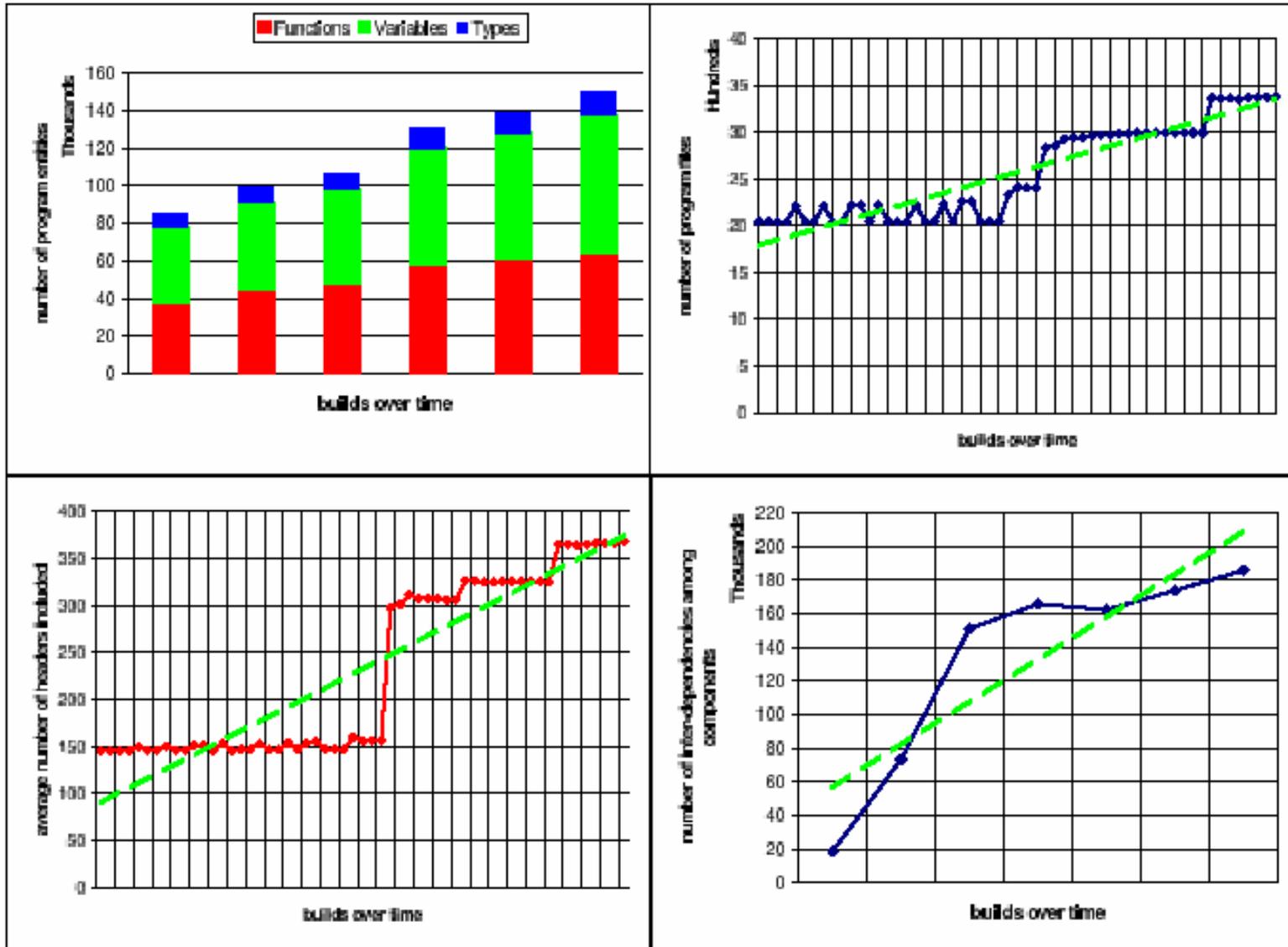
- <http://www.cs.toronto.edu/~yijun/literature/paper/dayani-fard05fase.pdf>
- VIM 6.2



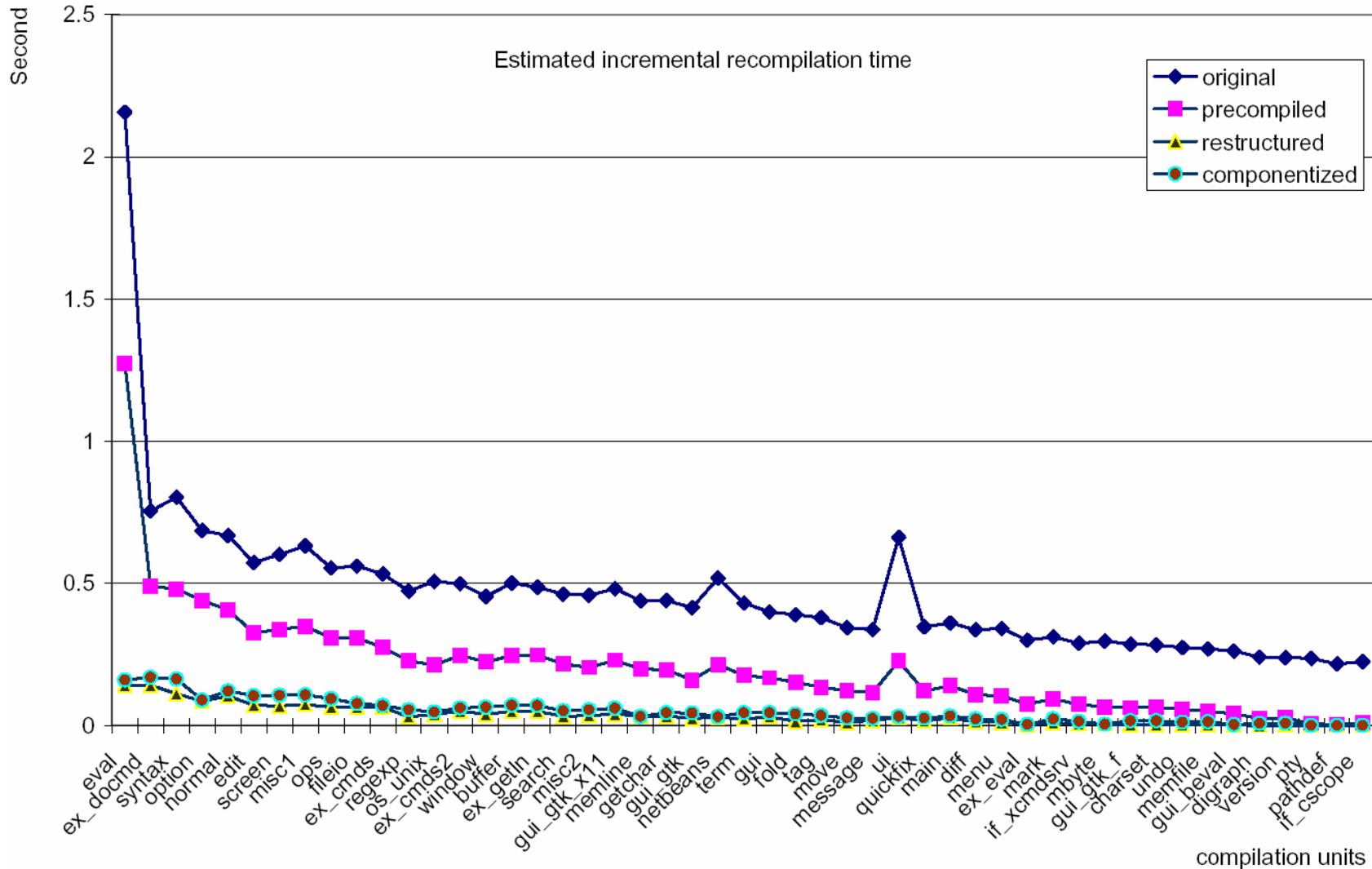
Restructuring headers

- Components provides and uses interfaces
- In C/C++, such as VIM, interfaces are written in Headers
- “Abstraction and information hiding” is a good principle in SE, thus we should do the componentization ...
- “Large-cohesion and Low coupling” is the modularity principle of SE
- The inclusion of the headers may violate this principle
 - Too much entities included leads to redundancies, and also
 - False dependencies
- It is an advanced topic to show how to restructure the program to remove all false dependencies
- And also componentize the program to minimize the number of interfaces.
- Implementation in the adapted version of GCC 3.4.0
- Applications to IBM database product and potentially a Wind River product

Motivation: Decaying metrics of an industrial product



Build performance results



Spring 2005

ECE450H1S

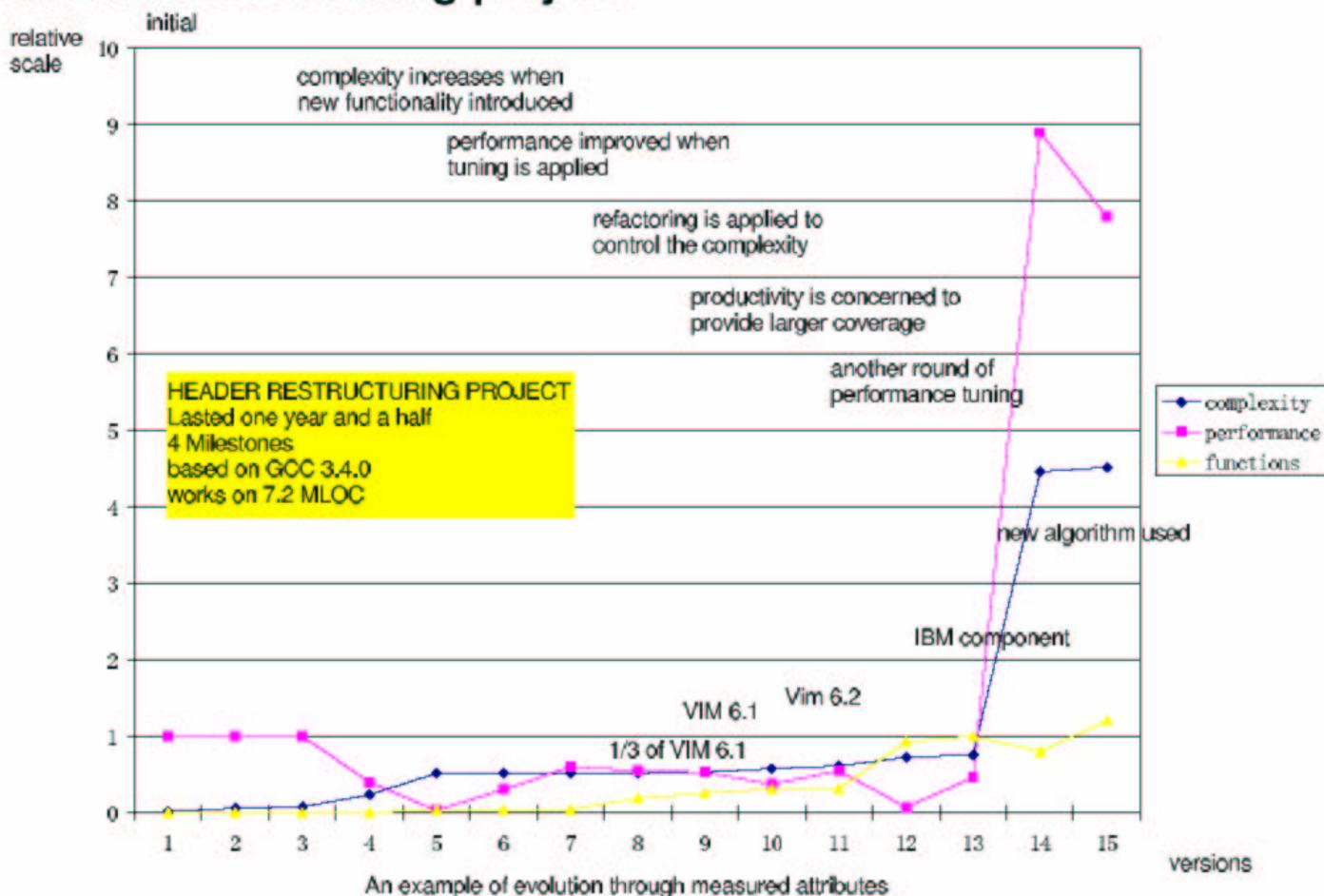
Software Engineering II

compilation units

Quality-driven software refactoring

- Refactoring is a technique to reveal hidden structure of the system. It helps maintainability by reducing complexity, but may hurt performance...

Header restructuring project



Case Study II. osCommerce

osCommerce - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Google Search Links

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

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 \$79.99

Shopping Cart

0 items

Bestsellers

01. Hewlett Packard LaserJet 1100Xi

Specials

Courage Under Fire
~~\$30.99~~
\$29.99

Reviews

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

Languages

Internet

Motivation

- PHP, 65 KLOC
- It is an parallel implementation of the Media Shop, an information system example in Goal-oriented Requirements Engineering
- It has been studied by clone detection
- We want to show the connection of goal models with aspect elicitation
Y. Yu, J.C. Leite, J. Mylopoulos. **“From Goals to Aspects: Discovering Aspects from Requirements Goal Models”**, RE 2004. 38-47.

Aspect-Orientation changes the way of thinking



z:\colleagues\julio\oscommerce-2.2ms2\catalog*.*

Name	Ext	Size	Date
↑...			
[admin]			
[download]			
[images]			
[includes]			
[install]			
[pub]			
! .htaccess			
account	php		
account_edit	php		
account_history	php		
account_history_info	php		
account_newsletters	php		
account_notifications	php		
account_password	php		
address_book	php		
address_book_process	php		
advanced_search	php		
advanced_search_result	php		
checkout_confirmation	php		
checkout_payment	php		
checkout_payment_address	php		
checkout_process	php		
checkout_shipping	php		
checkout_shipping_address	php		
checkout_success	php		
conditions	php		
contact_us	php		
cookie_usage	php		
create_account	php		
create_account_success	php		
download	php		
index	php		
info_shopping_cart	php		
login	php		
logoff	php		
password_forgotten	php		
popup_image	php		
popup_search_help	php		
privacy	php		
product_info	php		
product_reviews	php		
product_reviews_info	php		
product_reviews_write	php		
products_new	php		
redirect	php		
reviews	php		
shipping	php		
shopping_cart	php		
specials	php		
ssl_check	php		
stylesheet	css		
tell_a_friend	.		

To right

ACCOUNT_MGMT

ADVANCED_SEARCH

CHECK_OUT

LOGIN/OUT

PRIVACY_INFO

PRODUCT_INFO

SHOPPING_CART

SSL

File

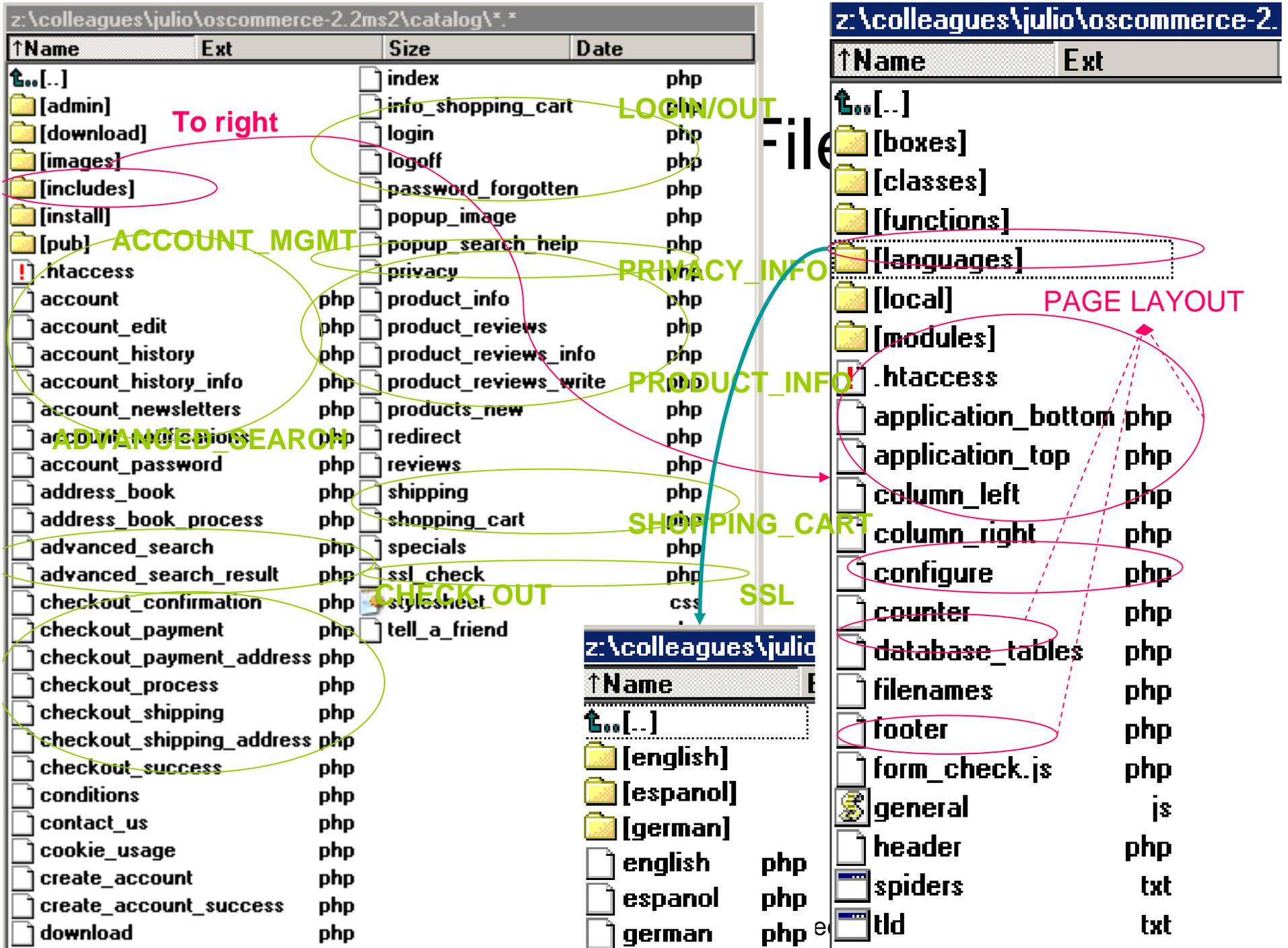
z:\colleagues\julio\oscommerce-2.2ms2\languages

Name	Ext
↑...	
[english]	
[espanol]	
[german]	
english	php
espanol	php
german	php

z:\colleagues\julio\oscommerce-2.2ms2\application

Name	Ext
↑...	
[boxes]	
[classes]	
[functions]	
[languages]	
[local]	
[modules]	
! .htaccess	
application_bottom	php
application_top	php
column_left	php
column_right	php
configure	php
counter	php
database_tables	php
filenames	php
footer	php
form_check.js	php
general	js
header	php
spiders	txt
tld	txt

PAGE LAYOUT

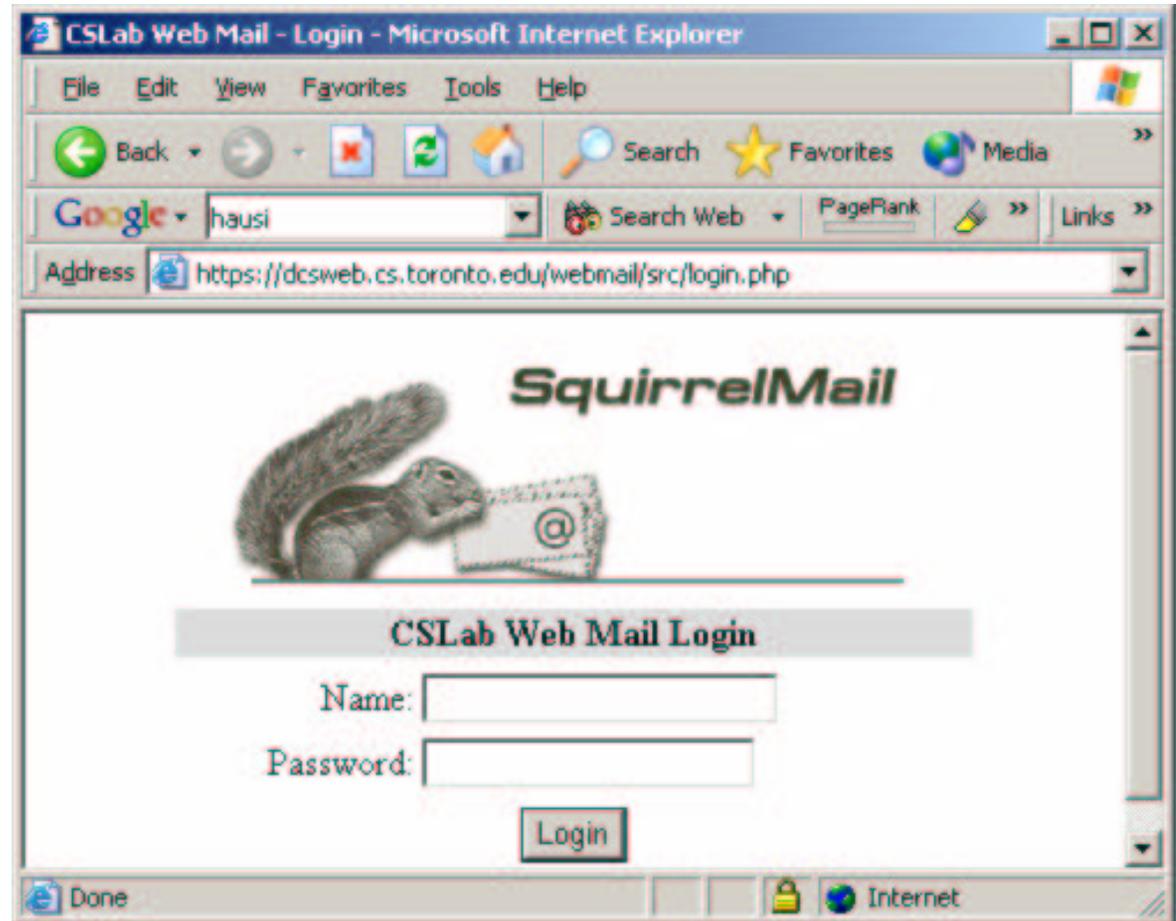


Case Study III. Squirrel Mail

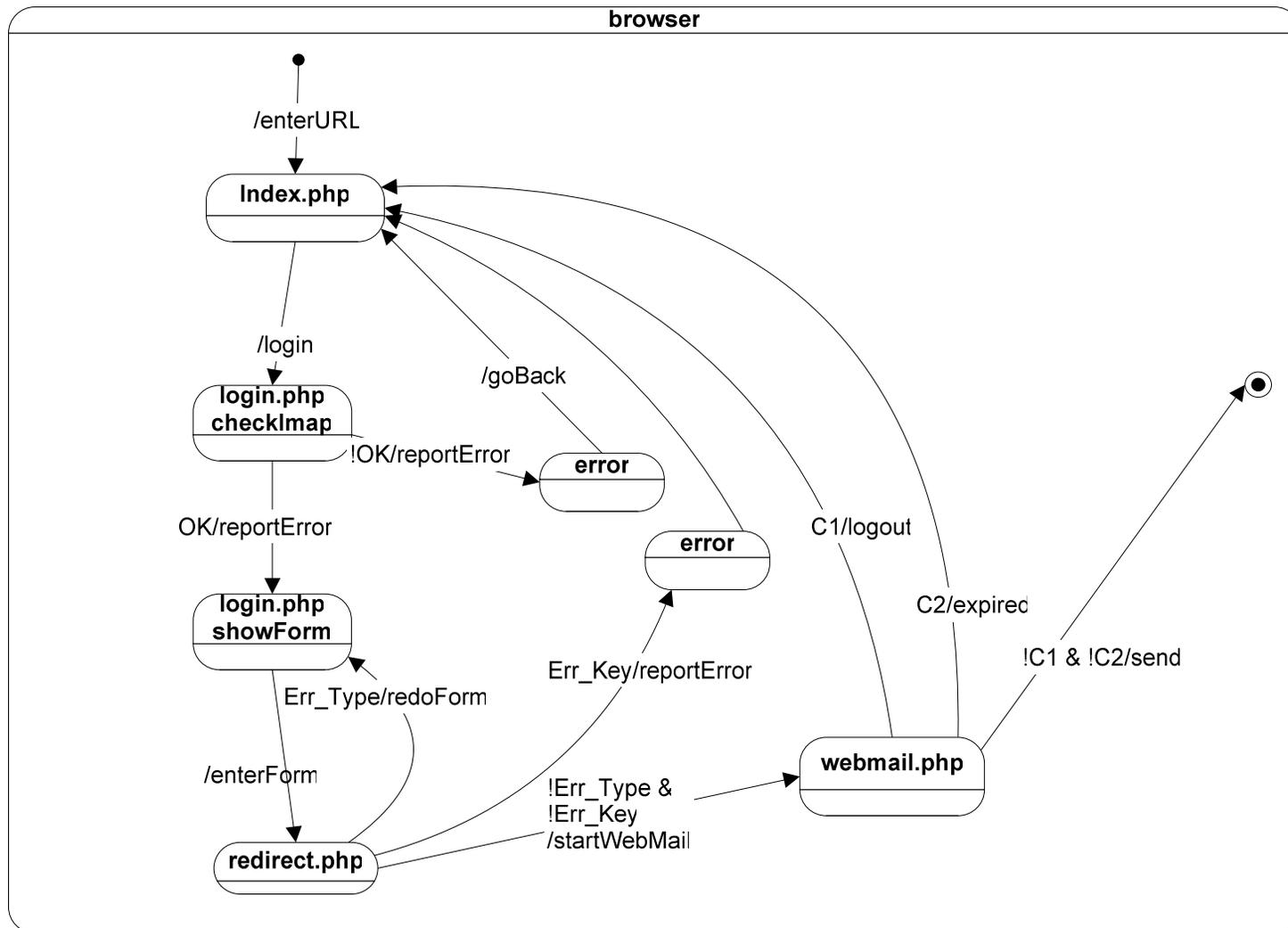
- It is a web-based email system used by the CS department
- We will explore the steps on how to refactor it to reveal the intention of developers: Code -> Statechart -> Goals
- The research is on-going on building the tool support. It will be associated with a tutorial on Eclipse tools

The appearance of the system

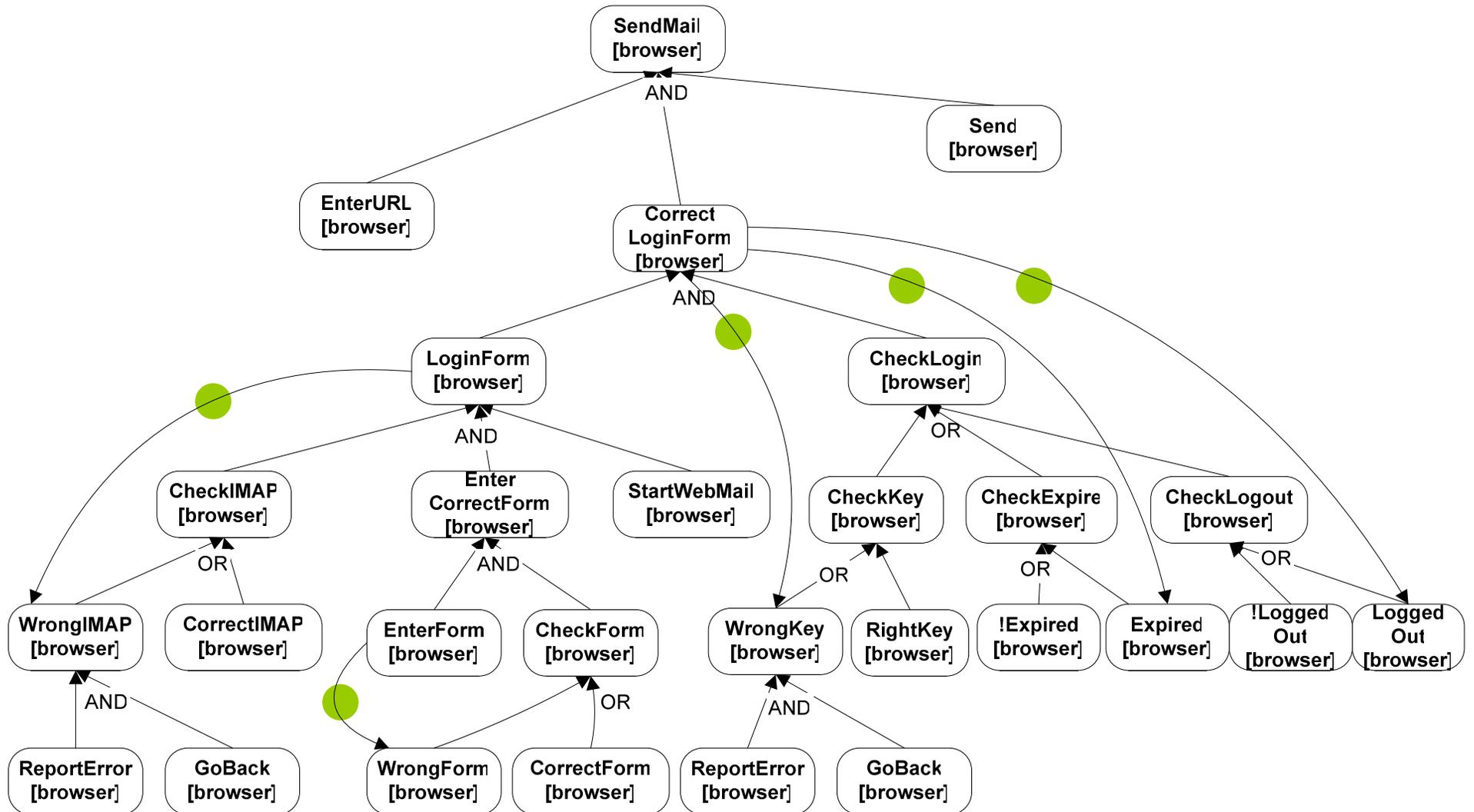
SquirrelMail 1.5.0
Open source
70 KLOC
PHP + HTML



A result of the refactoring



A result goal model



Summary

- Reengineering is a hot topic in the software engineering research
- Case studies show some ways to understand a legacy software
- We will use several tutorials to explore further on individual case studies, explaining advanced topics on:
 - The concepts of software architecture (components, service-oriented architecture, build architecture), aspect-oriented paradigm, software refactoring
 - The software engineering tools for these tasks, including code fact extraction, reflexion model, Eclipse, aspectJ etc.
 - How to apply them to our course project

What's next ...

- A Tutorial on Web Services
- Next lecture will give you some examples of requirement specifications and project documents
- Do we cover the material you want to learn? If no, please send me email and see whether the course can motivate your study ...