

# SWEVO

The McGill Software Evolution Research Group

Past - Present - Future

Martin Robillard et al.

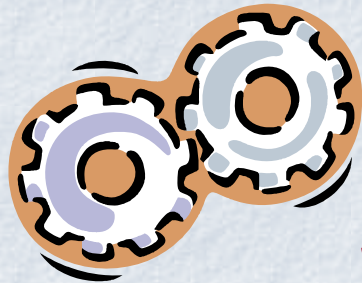


**McGill**

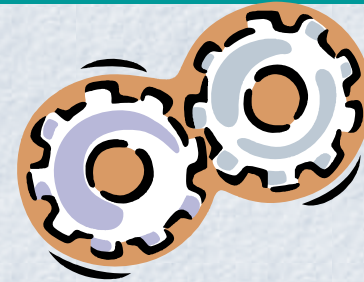
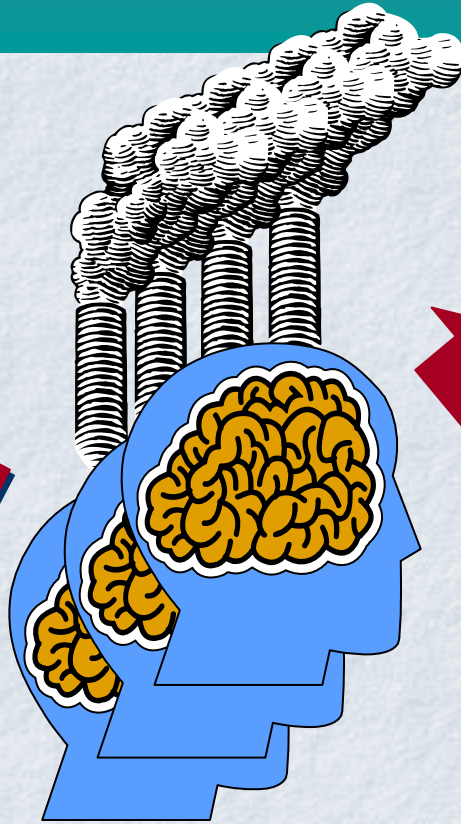
École d'Informatique  
School of Computer Science

# Context: Software in Evolution

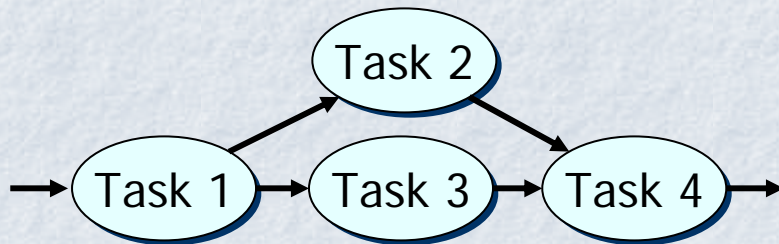
Big problem:  
Inefficiencies



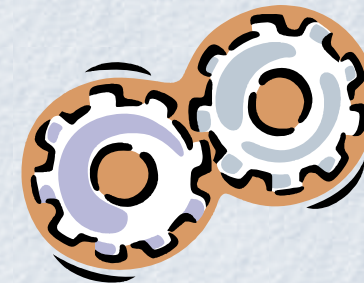
*Understand  
the application  
domain*



*Find and  
understand the  
relevant  
source code*

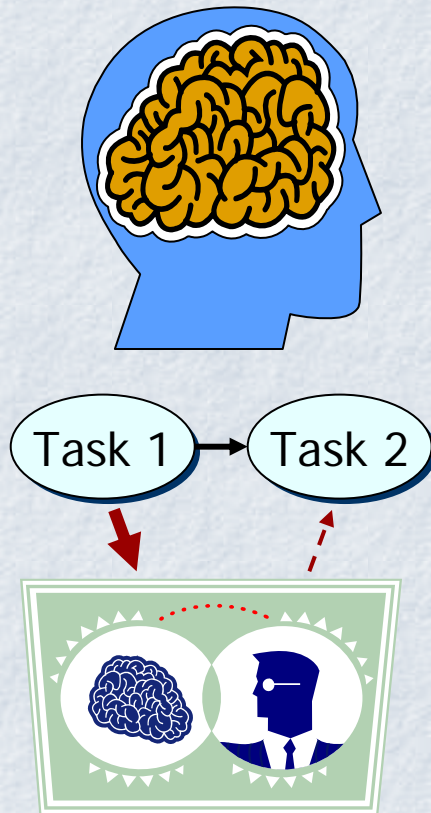


*Change the  
source code*



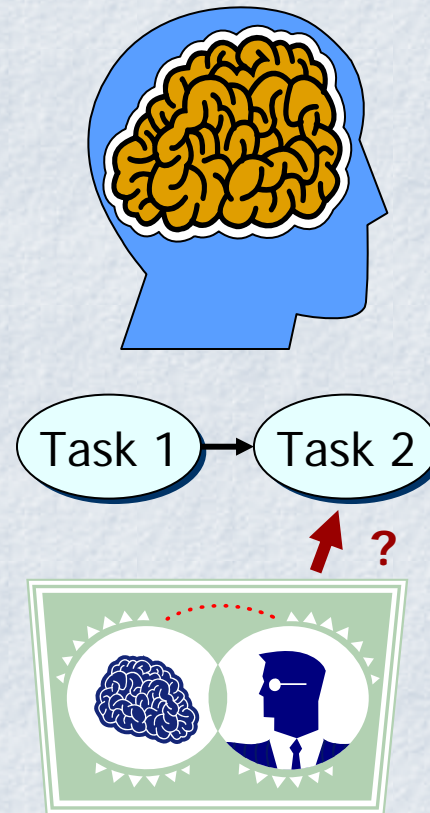
# Solution Areas

Producing new knowledge



- Design Rationale
- Tacit idioms
- Feature location
- API usage tricks
- And so on

Maintaining existing knowledge



# Specific Solution Areas

## Producing new knowledge

- Software navigation analysis
- Feature coupling analysis
- Crosscutting refactoring assessment
- Software topology analysis

## Maintaining existing knowledge

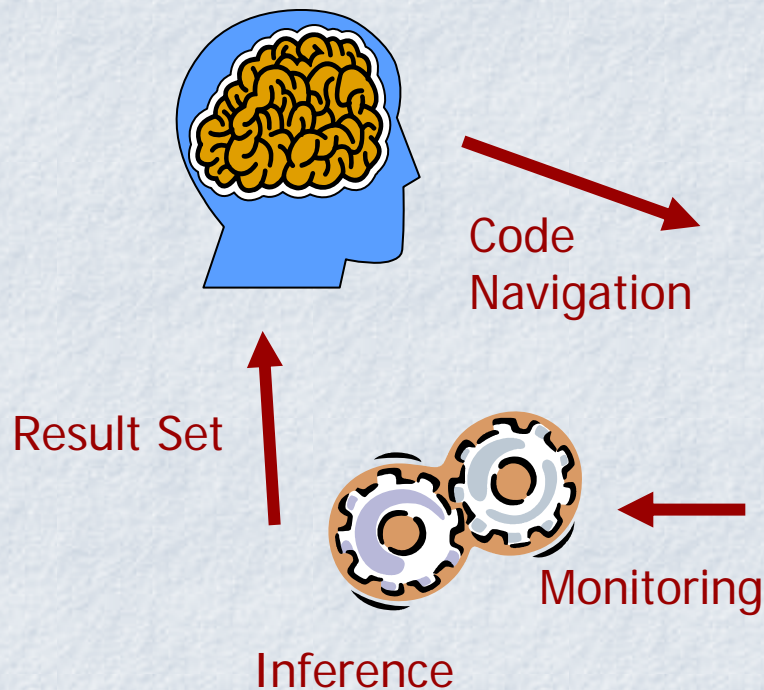
- Concern modeling
- Concern interface management
- API evolution tracking
- Clone tracking

# Software Navigation Analysis

Producing New Knowledge

**Premise:** Developers look at a lot of code when doing a change task.

**Question:** Can we automatically infer the important segments of the code based on their investigation?



```
* @param path The file path
* @since JEdit 2.4pre1
*/
public static Buffer openFile(View view, String path)
{
    return openFile(view, null, path);
}

//{{{ openFile() method
/**
 * @deprecated The openFile() method should not be used. The
 * openFile(String path, boolean readOnly) method should be used.
 */
public static Buffer openFile(View view, String path, boolean readOnly)
{
    return openFile(view, path, null, readOnly);
}

//{{{ openFile() method
/**
 * @deprecated The openFile() method should not be used. The
 * openFile(String path, boolean readOnly, Hashtable props) method
 * should be used.
 */
public static Buffer openFile(View view, String path, Hashtable props)
{
    return openFile(view, path, null, props);
}

//{{{ openFile() method
/**
 * Opens a file. This may return null if the buffer could not be
```

# Software Navigation Analysis

Producing New Knowledge



**Martin** Robillard, Gail Murphy.  
Automatically Inferring Concern  
Code from Program Investigation  
Activities, ASE 2003.



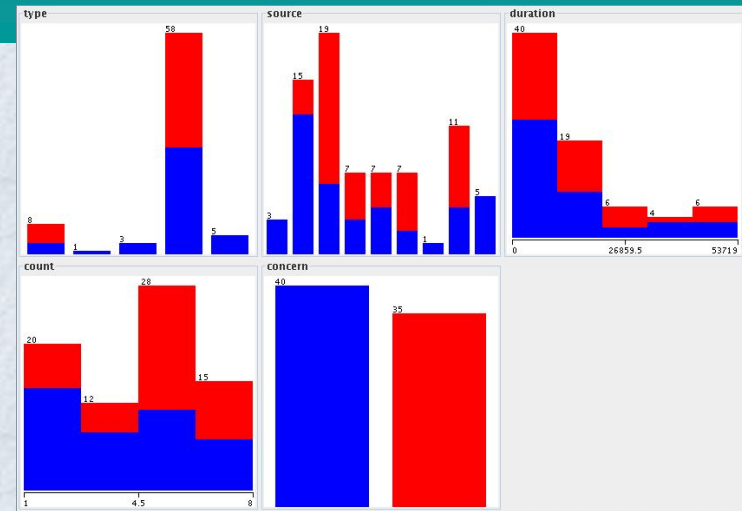
**Imran** Majid, Martin Robillard.  
NaCIN - An Eclipse Plug-In for  
Program Navigation-based  
Concern Inference, ETX 2005.

*Now at Microsoft*



**Punit** Agrawal, Martin  
Robillard, Doina Precup.  
Program Navigation Analysis, A  
Machine Learning Approach.

*Current M.Sc.*



## SONA 2006

Bellairs, Barbados  
Workshop on Software  
Navigation Analysis

## UW/MSR USC

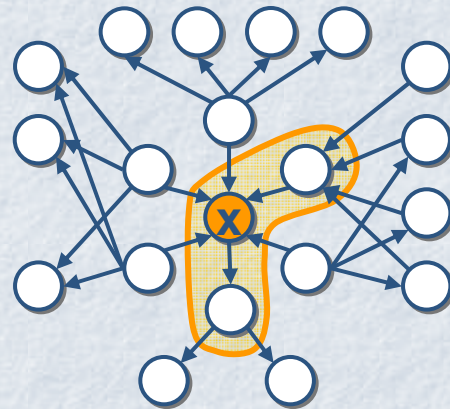
Columbia Gorge, WA, USA  
Summer Institute on the Human  
Side of Computing

# Topology-Based Searches for Software Investigation

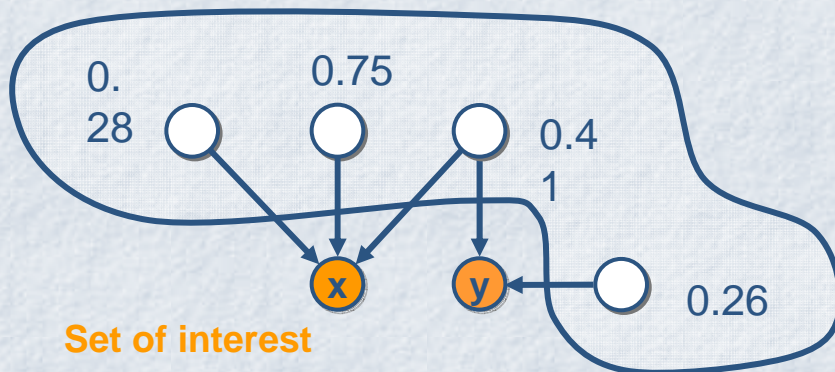
Producing New Knowledge

**Premise:** Sometimes developers have no clue where to look

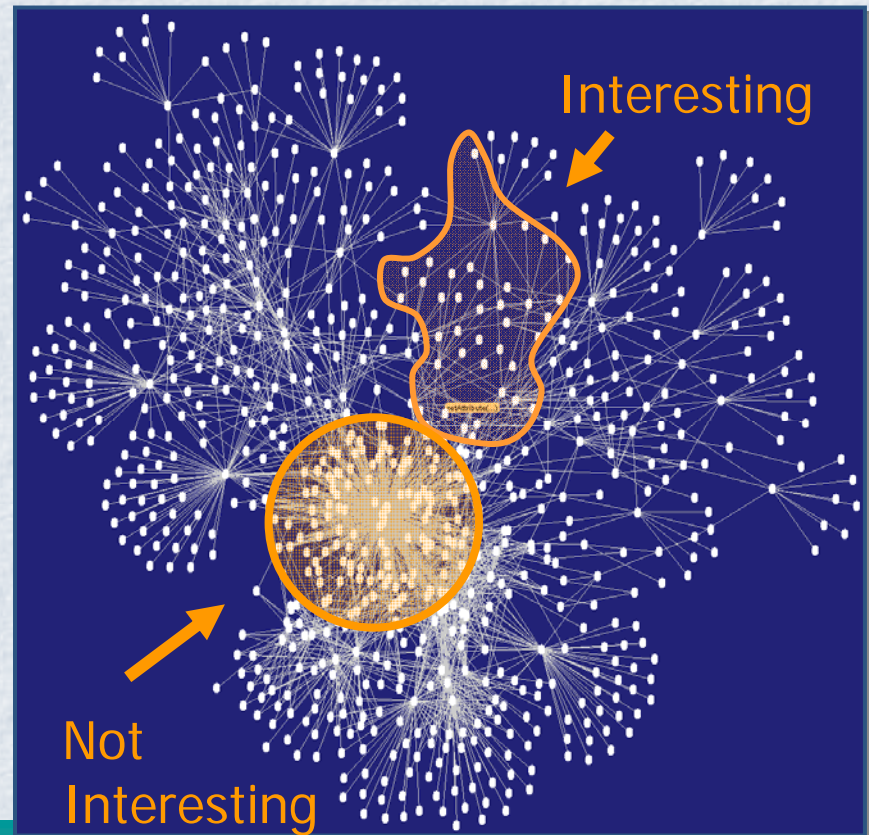
**Question:** Can the topology of software dependencies can help identify elements that are more likely to be relevant than others?



Suggestion set



Set of interest





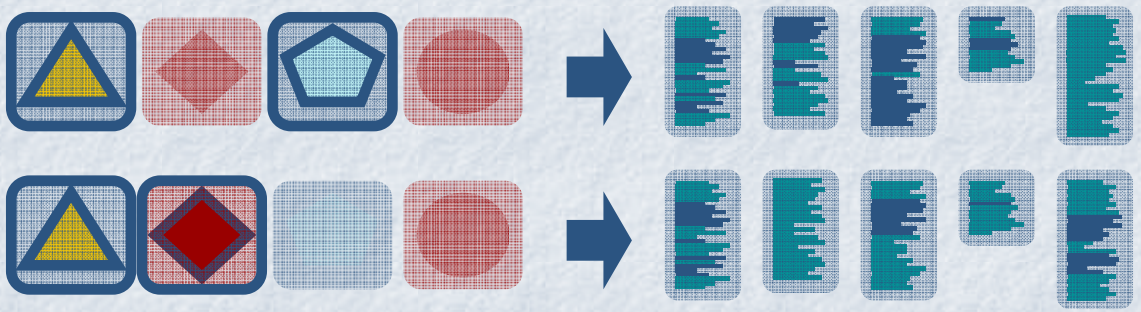


# Feature Coupling Analysis

Producing New Knowledge

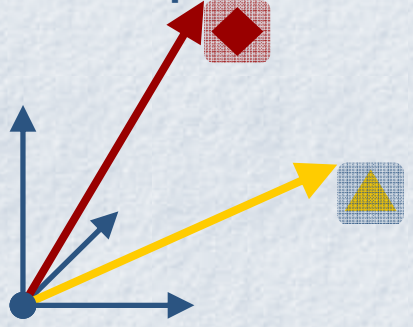
**Premise:** Code always seems to get worse (to “decay”).

**Question:** Can we automatically detect code decay when we run regression tests?

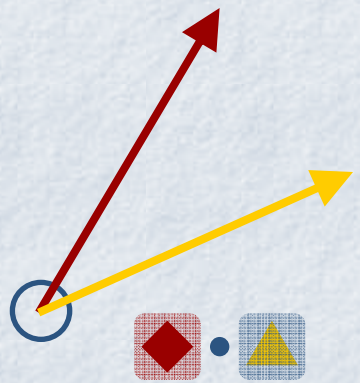


Olivier Giroux  
*Now at NVIDIA*

Implementation Space



Associations



Olivier Giroux and Martin P. Robillard. Detecting Increases in Feature Coupling using Regression Tests. *FSE 2006.*

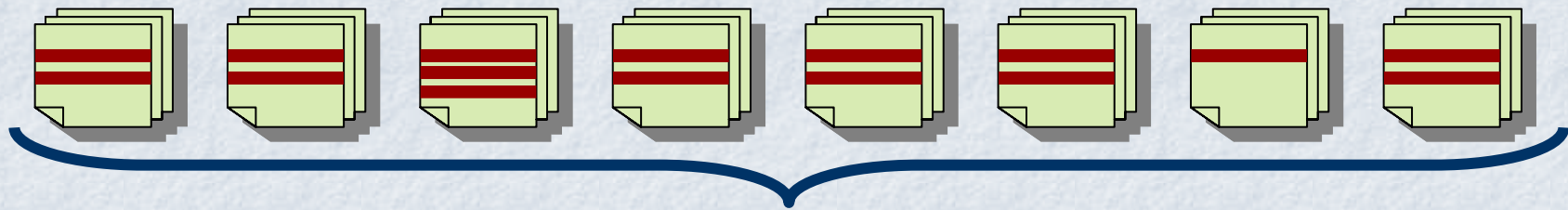
# Refactoring Assessment

Producing New Knowledge

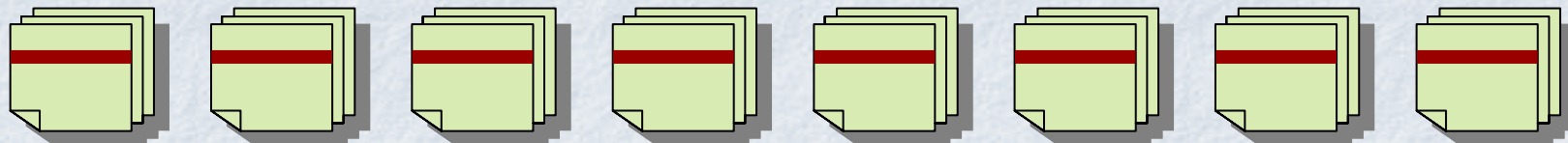
**Premise:** Some code really could benefit from refactoring.

**Question:** How can we easily refactor crosscutting concerns into better abstractions?

Aspect Mining



Refactoring



*Current M.Sc.*

**Isaac** Yuen, Martin Robillard.  
Bridging the Gap between  
Aspect Mining and Refactoring.  
*AOSD 2007 LATE Workshop*

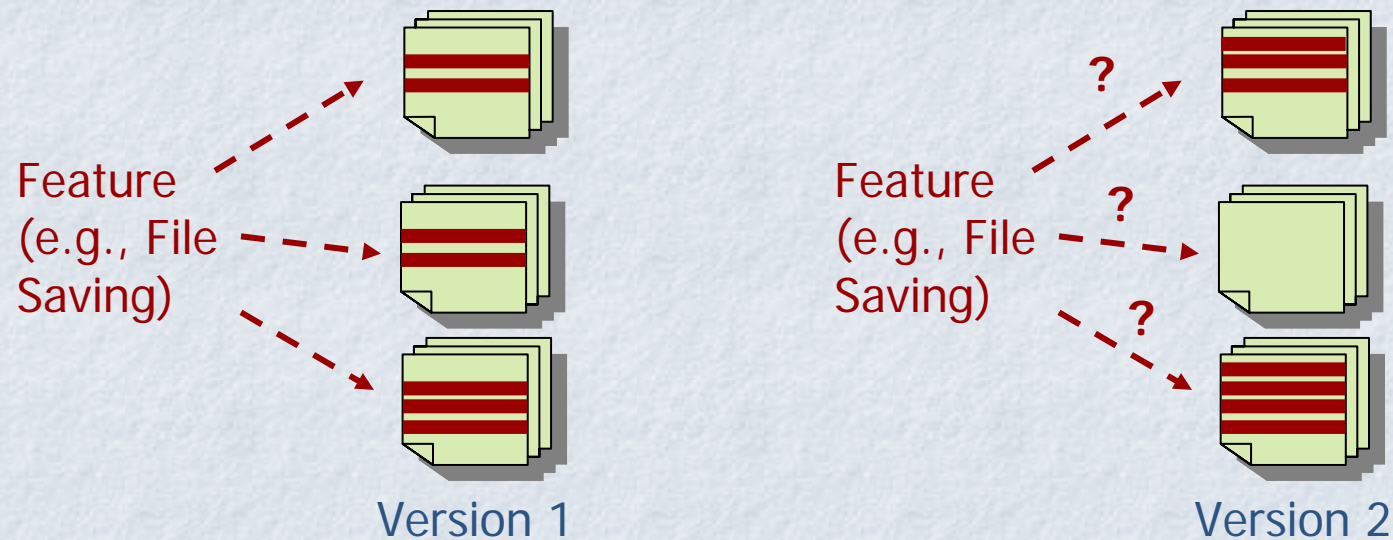


# Concern Modeling

Maintaining Existing Knowledge

**Premise:** We have a concern-code association (e.g., a feature location)

**Question:** How can we maintain this association as the code evolves?



# Concern Modeling: Take 1

Maintaining Existing Knowledge

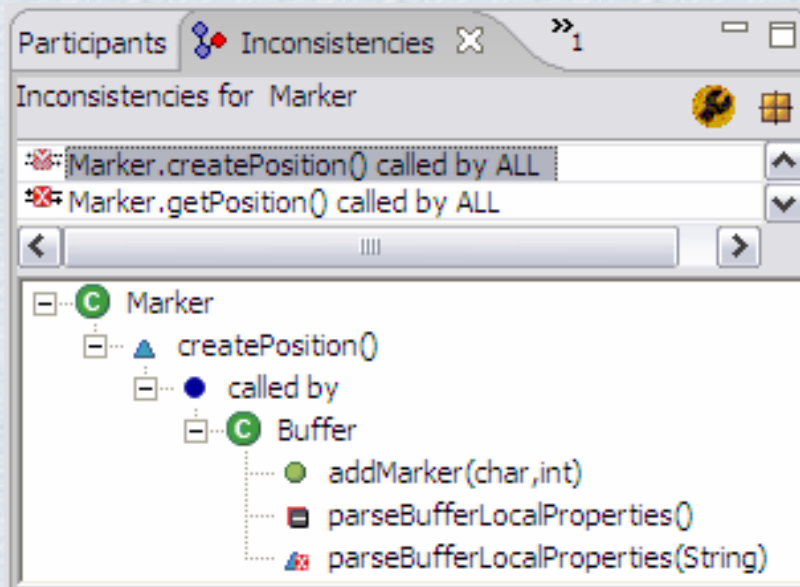
## Concern Graphs + FEAT

<CG> := <name><fragment>\*

<fragment> := <intension><extension>

Example intension:

Marker.createPosition() called by ALL



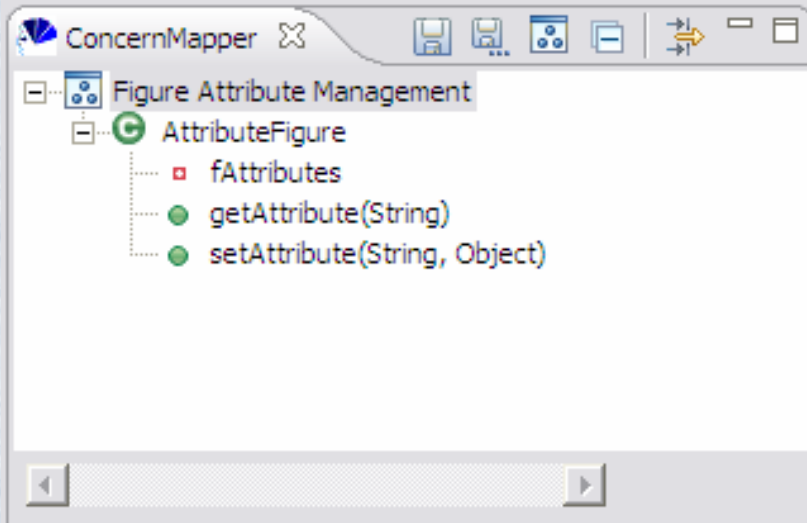
Robillard & Murphy,  
TOSEM 16 (1), 2007.

Problem: Hard to write  
good specs.

# Concern Modeling: Take 2

Maintaining Existing Knowledge

ConcernMapper – [www.cs.mcgill.ca/~martin/cm](http://www.cs.mcgill.ca/~martin/cm)



- Drag and drop elements
- Save the model
- Et voila!
- **Good:** Easy to use
- **Bad:** Limited tolerance to evolution

Actively maintained and enhanced since July 2007 by



Martin  
Robillard

*Talk maker upper*



Frédéric  
Weigan Warr

*McGill Ugrad*



Usman  
Ahmed

*At Electronic Arts*



Putra  
Manggala

*Summer Intern*

# Managing Concern Interfaces: JMantlet

Maintaining Existing Knowledge



**Premise:** Interfaces include implementation classes.

**Problem:** To check that interface conventions are not broken"

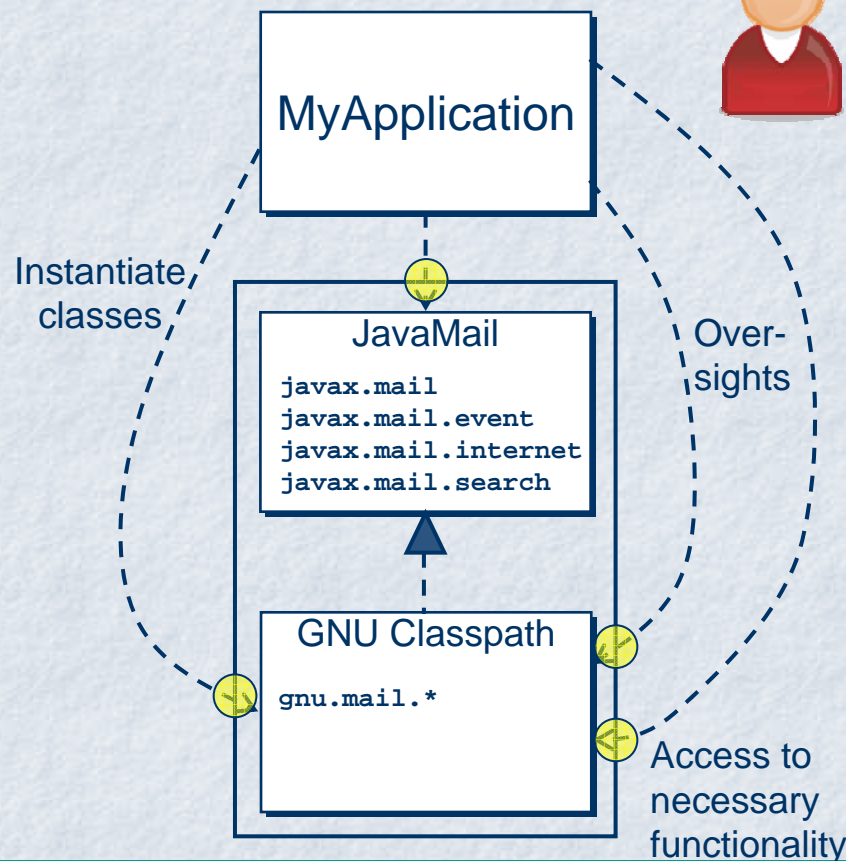


Jean-Sébastien  
Boulanger  
*Intern 2006*

Specify Concern  
Interface &  
Implementation

Check Project  
for Conformance

View Concern  
Model



```
package example.main;

public class Main {

    protected void connectToStore(Session session)
        throws MessagingException {

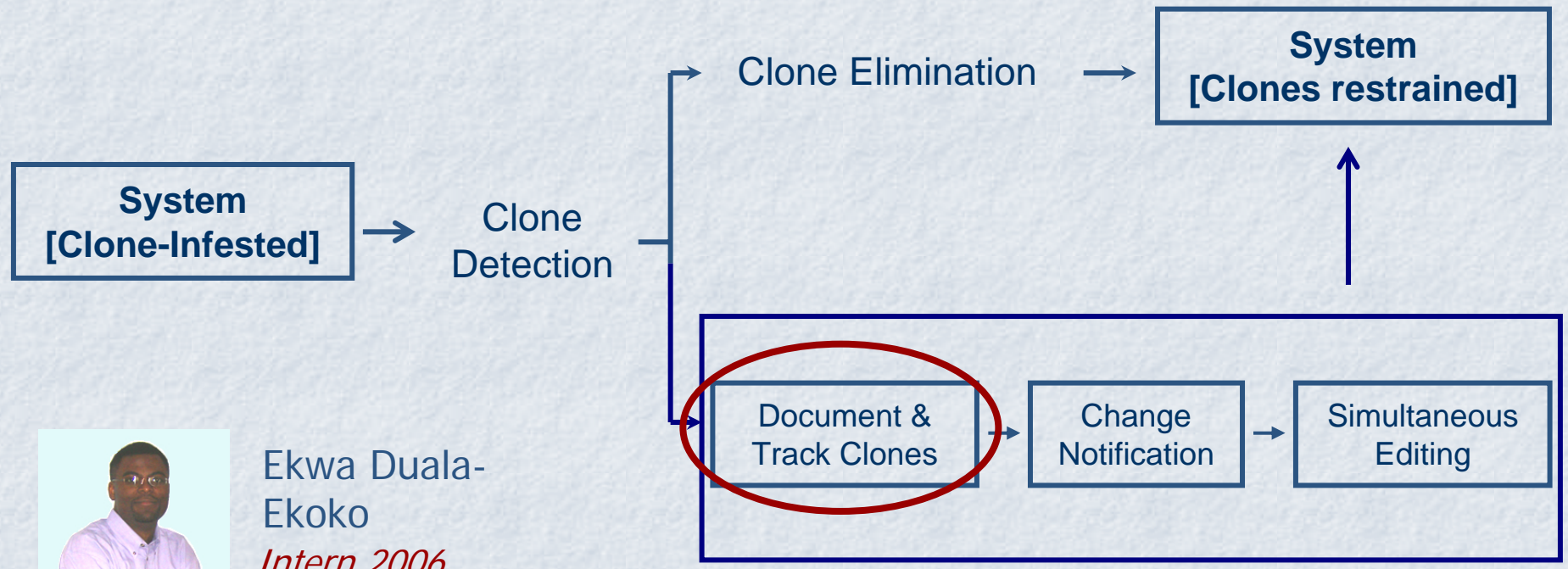
        MaildirStore store;
        store = new MaildirStore(null, null);
    }
}
```

# Tracking Code Clones in Evolving Software

Maintaining Existing Knowledge

**Premise:** Some clones cannot be refactored.

**Problem:** To keep track of critical or key clone groups as the code evolves.



Ekwa Duala-Ekoko  
*Intern 2006*

## Clone Management

Ekwa Duala-Ekoko and Martin P. Robillard. Tracking Code Clones in Evolving Software. To appear in ICSE 2007.

By the Way, don't Miss...

# ETX 2007

Eclipse Technology eXchange  
@ OOPSLA 2007

Montréal, 21-22 October 2007

[www.cs.mcgill.ca/~martin/etx2007/](http://www.cs.mcgill.ca/~martin/etx2007/)