Using Information Retrieval to Support Software Maintenance Tasks

Denys Poshyvanyk
Addressed Problem

- The software developer has to maintain large software systems with:
  - Little or no domain knowledge
  - Absence of the original developer
  - Badly organized, missing, or out of date documentation
Software Maintenance Requires

- **Structural Information** - the structural aspects of the source code (e.g., dependencies, architecture)

- **Dynamic information** - behavioral aspects of the program (e.g., feature implementation overlap)

- **Textual Information** - nature of the problem domain derived from names - identifiers, comments, documentation, and other artifacts

- **Historical Information** - history of changes (co-changes) of different software artifacts (e.g., CVS logs - rationale for changes)
Multitude of Software Artifacts

Source code

Call graphs

Design docs

Bug Reports

Documentation

Execution Traces
Research Focus

• *Structural Information* - the structural aspects of the source code (e.g., dependencies, architecture)

• *Dynamic information* - behavioral aspects of the program (e.g., feature implementation overlap)

• *Textual Information* - nature of the problem domain derived from names - identifiers, comments, documentation, and other artifacts

• *Historical Information* - history of changes (co-changes) of different software artifacts (e.g., CVS logs - rationale for changes)
Extracting and Indexing Lexical Information with Information Retrieval

- Parsing source code and extracting documents
  - corpus is a collection of documents (e.g., methods)

- Removing non-literals and stop words
  - common words in English, programming language keywords

- Preprocessing: split_identifiers and SplitIdentifiers

- Indexing and retrieving semantic information with Latent Semantic Indexing
Latent Semantic Indexing (LSI)

• Vector space model based Information Retrieval method [Deerwester 90, Dumais 94, Berry 95]

• It has been used successfully as a method to represent aspects of the meaning of words and passages in natural language

• Overcomes problems with polysemy and synonymy

• Programming (and natural) language independent

• Can be applied on a variety of software artifacts, including source code
Parsing Source Code and Extracting Documents

- Corpus is a collection of documents (e.g., methods, classes, files)

```java
public void run(IProgressMonitor monitor) throws InvocationTargetException, InterruptedException
if (n_iFlag == 0)
processCorpus(monitor, checkUpdate());
else if (n_iFlag == 2)
processCorpus(monitor, nD_UPDATECORPUS);
else
processQueryString(monitor);
if (monitor.isCanceled())
throw new InterruptedException("The long running");

public void run(IProgressMonitor monitor) throws InvocationTargetException, InterruptedException
if (n_iFlag == 0)
processCorpus(monitor, checkUpdate());
else if (n_iFlag == 2)
processCorpus(monitor, nD_UPDATECORPUS);
else
processQueryString(monitor);
if (monitor.isCanceled())
throw new InterruptedException("The long running");
```

• Corpus is a collection of documents (e.g., methods, classes, files)
public void run(IProgressMonitor monitor)
    throws InvocationTargetException, InterruptedException {
    if (m_iFlag == 0)
        processCorpus(monitor, checkUpdate());
    else if (m_iFlag == 2)
        processCorpus(monitor, UD_UPDATECORPUS);
    else
        processQueryString(monitor);

    if (monitor.isCanceled())
        throw new InterruptedException("The long running");
}

public void run(IProgressMonitor monitor) throws InvocationTargetException, InterruptedException {
    if (m_iFlag == 0)
        processCorpus(monitor, checkUpdate());
    else if (m_iFlag == 2)
        processCorpus(monitor, UD_UPDATECORPUS);
    else
        processQueryString(monitor) if (monitor.isCancelled())
            throw new InterruptedException("The long running");
Removing Stop Words

- Common words in English, programming language keywords

```java
public void run(IProgressMonitor monitor) throws InvocationTargetException, InterruptedException {
    if (m_iFlag == the) {
        processCorpus(monitor).checkUpdate();
    } else if (m_iFlag == processCorpus(monitor).UD_UPDATECORPUS) {
        // Code
    } else if (processQueryString(monitor).isCancelled()) {
        throw new InterruptedException();
    }
    long running = // Code
```
### Splitting Identifiers

```java
public void run(IProgressMonitor monitor) throws InvocationTargetException, InterruptedException {
    if (m_iFlag) {
        processCorpus(monitor, checkUpdate);
    } else if (m_iFlag) {
        processCorpus(monitor, UD_UPDATECORPUS);
    } else {
        processQueryString(monitor);
        if (monitor.isCancelled()) {
            throw new InterruptedException()
        }
    }
}
```

- `IProgressMonitor = i progress monitor`
- `InvocationTargetException = invocation target exception`
- `m_iFlag = m i flag`
- `UD_UPDATECORPUS = ud updatecorpus`
Indexing Source Code with Latent Semantic Indexing

```java
public void run(IProgressMonitor monitor)
    throws InvocationTargetException, InterruptedException{
    if ( m_iFlag == 0 )
        processCorpus(monitor,checkUpdate());
    else if ( m_iFlag == 2 )
        processCorpus(monitor, UD_UPDATECORPUS);
    else
        processQueryString(monitor);
    if (monitor.isCanceled())
        throw new InterruptedException("The long running
```
Combining Structural, Dynamic and Textual Information

Software Artifacts

IR Tool LSI

Analysis Tool 1 Parser

Analysis Tool N Tracer

Structural Information
AST
Call Graph
Data Flow
Control Flow

Semantic Similarities
System Representation
Contributions on the Thesis

• Feature (or concept) location in source code

• Coupling measurement and impact analysis in source code

• Cohesion measurement and fault prediction in source code
Concept Location in Source Code

- Change request
- Concept Location
- Impact Analysis
- Implementation
- Change Propagation
- Testing
Concept Location with Regular Expressions

```
BSTNode* BST::insert(KeyType key, ValueType value)
// lookup key, create key-value if not present
// return pointer to node with key
{
    BSTNode* node = new Node(key, value);
    BSTNode* root = NULL;
    if (root == NULL)
    {
        root = node;
        return root;
    }
    if (root->key < key)
        root->right = insert(key, value);
    else
        root->left = insert(key, value);
    return root;
}
```
Our Motivation
### JIRiSS

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera</td>
<td>setScreenParams</td>
<td>0.813055</td>
</tr>
<tr>
<td>ScaleObjectTool</td>
<td>mouseDragged</td>
<td>0.807305</td>
</tr>
<tr>
<td>Camera</td>
<td>setScreenParamsParalle</td>
<td>0.80328</td>
</tr>
<tr>
<td>Camera</td>
<td>setSize</td>
<td>0.788526</td>
</tr>
<tr>
<td>ViewerCanvas</td>
<td>setScale</td>
<td>0.723998</td>
</tr>
<tr>
<td>GLCanvasDrawer</td>
<td>prepareView3D</td>
<td>0.705305</td>
</tr>
<tr>
<td>ViewerCanvas</td>
<td>scaleChanged</td>
<td>0.700639</td>
</tr>
<tr>
<td>JitterModule</td>
<td>setZScale</td>
<td>0.690434</td>
</tr>
<tr>
<td>JitterModule</td>
<td>setXScale</td>
<td>0.690117</td>
</tr>
<tr>
<td>ImageModule</td>
<td>setXScale</td>
<td>0.690104</td>
</tr>
</tbody>
</table>

### Google Eclipse Search

Enter the search string:

- **Animation Preview**

**Selected matches:**

- ActorEditorWindow.java - Art of Illusion/ArtOfIllusion/src/artofillusion/animation (11 matches)
- AnimationPreviewer.java - Art of Illusion/ArtOfIllusion/src/artofillusion/animation (19 matches)
- CSSObject.java - Art of Illusion/ArtOfIllusion/src/artofillusion/object (2 matches)
- CustomDistortionTrack.java - Art of Illusion/ArtOfIllusion/src/artofillusion/animation/distortion (10 matches)
- CylindricalMapping.java - Art of Illusion/ArtOfIllusion/src/artofillusion/texture (12 matches)
- LinearMapping3D.java - Art of Illusion/ArtOfIllusion/src/artofillusion/texture (12 matches)
- ObjectInfo.java - Art of Illusion/ArtOfIllusion/src/artofillusion/object (13 matches)
- ObjectTextureDialog.java - Art of Illusion/ArtOfIllusion/src/artofillusion/animation (42 matches)
- ProceduralPositionTrack.java - Art of Illusion/ArtOfIllusion/src/artofillusion/animation (9 matches)

### IRiSS

#### IRiSS - persistent cache

<table>
<thead>
<tr>
<th>Method</th>
<th>confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCrystalTextView::WrapLineCached</td>
<td>0.87</td>
</tr>
<tr>
<td>CCrystalTextView::InvalidateLineCache</td>
<td>0.84</td>
</tr>
<tr>
<td>CCrystalTextView::DrawSingleLine</td>
<td>0.82</td>
</tr>
<tr>
<td>CCrystalTextView::OnDraw</td>
<td>0.79</td>
</tr>
<tr>
<td>CCrystalTextView::SetFont</td>
<td>0.68</td>
</tr>
<tr>
<td>CCrystalTextView::OnSize</td>
<td>0.59</td>
</tr>
<tr>
<td>CCrystalTextView::CalculateActualOffset</td>
<td>0.57</td>
</tr>
</tbody>
</table>
Feature Location with Information Retrieval and Formal Concept Analysis

[Poshyvanyk’07]
Top 160 results of at least 73,100 retrieved for the query **icsm** (details)

**Did you mean:** **pics**

**KEl Kinnaman Electric** - Shipboard Electrical Supplies Dynalet-Symbols-Breakers-Panels-etc.

1. **ICS.M- The Intergovernmental Committee on Surveying and Mapping**
   Representatives from all the Australian States, Territories, and the Commonwealth and New Zealand.

2. **ICS.M 2008: Home - 24th IEEE International Conference on Software ...**
   Welcome. See you again next year in Edmonton, Canada! ICS.M is the premier international event.

   ICS.M Limited provide I.T. support and services throughout the South West of England.
   [www.icsm.co.uk](http://www.icsm.co.uk) - [cache] - Bing, Ask

4. **ICS.M: InstitutDeChimieSeparative**
   Website of ICS.M, Separative Chimie Institute of Marcoule ... This institute is a mixed research unit.
   [icsm.trienglish](http://icsm.trienglish) - [cache] - Bing, Ask

5. **ICS.M 2002**
   International Conference on Software Maintenance - ICS.M 2002 - Montreal, Canada, Software Main.
Cluster International Conference contains 53 documents.

1. **ICSM 2008: Home - 24th IEEE International Conference on Software ...**
   Welcome. See you again next year in Edmonton, Canada! ICSM is the premier international event in the so:

2. **ICSM 2007 - 23rd International Conference on Software Maintenance ...**
   ICSM 2007 is now over. Some post-conference information is now available. Gail Murphy keynote present:
   icsm07.ai.univ-paris8.fr - [cache] - Bing, Ask

3. **International Conference on Software Maintenance (ICSM) - Resources**
   The ICSM conference is the premier international (world wide) event in the maintenance field.
   conferences.computer.org/icsm - [cache] - Ask

4. **ICSM 2009: The 2009 IEEE International Conference on Software ...**
   International Conference on Software Maintenance - ICSM 2009 - Edmonton, Canada, Software Mainte
   icsm2009.cs.ualberta.ca - [cache] - Bing

5. **IEEE International Conference on Software Maintenance**
   IEEE International Conference on Software Maintenance ICSM Resources site 24. ICSM 2008: Beijing,
   www.informatik.uni-erlangen.de/ley/di/conf/icsm/index.html - [cache] - Bing, Ask

6. **ICSM - What does ICSM stand for? Acronyms and abbreviations by the ...**
   Acronym Definition, ICSM International Conference on Software Maintenance. ICSM: Imperial College
Results: Locating 105 Features in Rhino
Shortcomings of Static Concept Location

- Highly dependent on naming conventions and the developer’s experience to write good queries
- Ignores other existing relationships between software components (such as, dependencies)
- May miss important parts of the source code
Feature Location with Scenario-based Probabilistic Ranking

Scenario NOT exercising the feature (trace 1)
- `readAndDispatch` -- `org.eclipse.swt.widgets.Display`
- `checkDevice` -- `org.eclipse.swt.widgets.Display`
- `isDisposed` -- `org.eclipse.swt.graphics.Device`
- `drawMenuBars` -- `org.eclipse.swt.widgets.Display`
- `runPopups` -- `org.eclipse.swt.widgets.Display`
- `filterMessage` -- `org.eclipse.swt.widgets.Display`
- `windowProc` -- `org.eclipse.swt.widgets.Display`
- `WM_TIMER` -- `org.eclipse.swt.widgets.Control`
- `windowProc` -- `org.eclipse.swt.widgets.Display`
- `WM_TIMER` -- `org.eclipse.swt.widgets.Control`
- `windowProc` -- `org.eclipse.swt.widgets.Display`
- `windowProc` -- `org.eclipse.swt.widgets.Control`

Scenario exercising the feature (trace 2)
- `checkDevice` -- `org.eclipse.swt.widgets.Display`
- `isDisposed` -- `org.eclipse.swt.graphics.Device`
- `drawMenuBars` -- `org.eclipse.swt.widgets.Display`
- `runPopups` -- `org.eclipse.swt.widgets.Display`
- `filterMessage` -- `org.eclipse.swt.widgets.Display`
- `windowProc` -- `org.eclipse.swt.widgets.Display`
- `windowProc` -- `org.eclipse.swt.widgets.Control`
- `checkDevice` -- `org.eclipse.swt.widgets.Display`
- `isDisposed` -- `org.eclipse.swt.graphics.Device`
- `drawMenuBars` -- `org.eclipse.swt.widgets.Display`
- `runPopups` -- `org.eclipse.swt.widgets.Display`
- `runAsyncMessages` -- `org.eclipse.swt.widgets.Display`
- `removeFirst` -- `org.eclipse.swt.widgets.Synchronizer`

Knowledge-based filtering

Probabilistic ranking of events
Shortcomings of Dynamic Concept Location

- Execution traces are large even for small systems
- Selecting multiple scenarios may be difficult
- Filtering the traces is equally problematic - best filtering methods still return hundreds of methods
Probabilistic Ranking Of METHODS and Information Retrieval

[TSE’07]

Scenario NOT exercising the feature (trace 1)
readAndDispatch -- org.eclipse.swt.widgets.Display
checkDevice -- org.eclipse.swt.widgets.Display
isDisposed -- org.eclipse.swt.graphics.Device
drawMenuBars -- org.eclipse.swt.widgets.Display
filterMessage -- org.eclipse.swt.widgets.Display
runPopups -- org.eclipse.swt.widgets.Display
drawMenuBars -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control
WM_TIMER -- org.eclipse.swt.widgets.Control

Scenario exercising the feature (trace 2)
checkDevice -- org.eclipse.swt.widgets.Display
isDisposed -- org.eclipse.swt.graphics.Device
drawMenuBars -- org.eclipse.swt.widgets.Display
drawMenuBars -- org.eclipse.swt.widgets.Display
drawMenuBars -- org.eclipse.swt.widgets.Display
drawMenuBars -- org.eclipse.swt.widgets.Display
drawMenuBars -- org.eclipse.swt.widgets.Display
windowProc -- org.eclipse.swt.widgets.Control
Feature Location with PROMESIR

1. Query
2. Scenarios
3. Events
4. Execution traces
5. Indexes
6. Ranks
7. Results
Example of using PROMESIR

- Locating a feature in JEdit
- Feature: “showing white-space as a visible symbol in the text area”
- Steps:
  - Run two scenarios
  - Run query
  - Explore results
First Scenario Exercising the Feature in JEdit

Start Tracing
First Scenario Exercising the Feature in JEdit

Stop Tracing
Second Scenario **NOT** Exercising the Feature in JEdit

```java
// public class JEditTextArea extends JComponent {

    // constructor.
    /**
     * Creates a new JEditTextArea.
     */
    public JEditTextArea(View view) {
      // add by lzq.
      NormalCursor = new Cursor(TEXT_CURSOR);
      HandCursor = new Cursor(HAND_CURSOR);
      enableEvents(AWTEvent.FOCUS_EVENT_MASK | AWTEvent.KEY_EVENT_MASK);
      this.view = view;

      // initialize some misc. stuff.
      // add by lzq.
      urlSelection = new Vector();
      selection = new Vector();
      chunkCache = new ChunkCache(this);
      painter = new TextAreaPainter(this);
      painter.addExtension(new ToolTipTextAreaExtension(this));
      gutter = new Gutter(view, this);
      listenerList = new EventListenerList();
      caretEvent = new NotifyCaretEvent();
    }
```
Example of using PROMESIR - Results

- Number of methods identified by SPR - 284
- The position of the first relevant method according to IR ranking - 56
- Position of the first relevant method according to PROMESIR - 7
Case Studies

• Locating features associated with bugs:

<table>
<thead>
<tr>
<th></th>
<th>Mozilla</th>
<th>Eclipse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classes</td>
<td>4,853</td>
<td>7,648</td>
</tr>
<tr>
<td>Methods</td>
<td>53,617</td>
<td>89,341</td>
</tr>
<tr>
<td>Words</td>
<td>85,439</td>
<td>56,861</td>
</tr>
</tbody>
</table>

• Case study objectives:
  - Compare PROMESIR with stand-alone feature location approaches: LSI and SPR
Contributions on the Thesis

- Feature (or concept) location in source code

- Coupling measurement and impact analysis in source code

- Cohesion measurement and fault prediction in source code
Impact Analysis in Source Code
Comparing Structural and Conceptual Coupling Measures for Impact Analysis [ESE’09]

- Conceptual coupling of classes (CoCC)
  - Measures overlap of shared names in comments and identifiers
- Eleven coupling (two conceptual and nine structural) measures are used to rank classes during impact analysis
- Precision/recall values are computed for 1490 classes in 391 bug reports (between Mozilla 1.6 and 1.7)
- Conceptual coupling measure CoCC is the best predictor of actually changed classes
Contributions on the Thesis

- Feature (or concept) location in source code
- Coupling measurement and impact analysis in source code
- Cohesion measurement and fault prediction in source code
Fault Prediction in Source Code

[TSE’08 ]
Research

• Dynamic analysis
• Static analysis
• Information Retrieval

Problems

• Feature location
• Impact analysis
• Bug prediction
Research

- Dynamic analysis
- Static analysis
- Information Retrieval

Problems

- Feature location
- Impact analysis
- Bug prediction
Research

- Dynamic analysis
- Static analysis
- Information Retrieval

Problems

- Feature location
- Impact analysis
- Bug prediction
Research

- Dynamic analysis
- Static analysis
- Information Retrieval

Problems

- Feature location
- Impact analysis
- Bug prediction
Research

- Dynamic analysis
- Static analysis
- Information Retrieval

Problems

- Feature location
- Impact analysis
- Bug prediction
Current Research

- Dynamic analysis
- Static analysis
- Information Retrieval
- **Mining software repositories**

Problems

- Feature location
- Impact analysis
- Bug prediction
Current Research

- Dynamic analysis
- Static analysis
- Information Retrieval
- Mining software repositories

New Problems

- Feature location
- Impact analysis
- Bug prediction
- Identifying experts
Current Research

- Dynamic analysis
- Static analysis
- Information Retrieval
- Mining software repositories

New Problems

- Feature location
- Impact analysis
- Bug prediction
- Identifying experts
- Retrieving relevant applications from open-source repositories
Sneak Preview of Current Research

- Internet-scale engine for retrieving highly relevant applications
- The team:
  - Mark Grechanik, Accenture, U of Illinois in Chicago
  - Chen Fu, Accenture
  - Qing Xie, Accenture
  - Collin McMillan, W&M
  - Denys Poshyvanyk, W&M
  - Chad Cumby, Accenture

Check it out:
www.xemplar.org
Acknowledgements

• Andrian Marcus, advisor @ Wayne State U
• Václav Rajlich @ Wayne State U
• Jonathan Maletic @ Kent State U
• Giuliano Antoniol @ École Polytechnique de Montréal
• Yann-Gaël Guéhéneuc @ École Polytechnique de Montréal
Conclusions

• Proposed and implemented novel methods for feature location, impact analysis, bug prediction and software measurement based on combination of:
  - Dynamic program analysis, static program analysis and Information Retrieval

• Developed practical tools for the proposed approaches

• Evaluated proposed methods and tools on large open-source systems