Lecture 11
LSdiff evaluation / Focus group study
Mining Software Repositories, Part 1
eRose
Announcement

• Project Checkpoint Due on this thursday.
• I won’t grade them.
• It is not mandatory.
• You are encouraged to submit to seek my feedback.
• Available for both research project, literature survey, and tool evaluation
Today’s Agenda

- LSdiff evaluation
- LSdiff focus group study
- Presentation: Tileli (advocate), Gaurav (skeptic)
- eRose
- Quiz
**Question:** What kinds of rules can LSdiff find?

**Horn Clause:** \[ A(x) \land B(x,y) \land C(y) \Rightarrow D(x,y) \]

<table>
<thead>
<tr>
<th>Rule Styles</th>
<th>High-level Change Patterns</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>past_* (\Rightarrow) deleted_*</td>
<td>dependency removal, feature deletion, etc.</td>
<td>past_calls(m, “DB.exec”) (\Rightarrow) deleted_calls(m, “DB.exec”)</td>
</tr>
<tr>
<td>past_* (\Rightarrow) added_*</td>
<td>consistent updates to clones, etc.</td>
<td>past_accesses(“Log.on”, m) (\Rightarrow) added_calls(m, “Log.trace”)</td>
</tr>
<tr>
<td>current_* (\Rightarrow) added_*</td>
<td>dependency addition, feature addition, etc.</td>
<td>current_method(m, “getHost”, t) (\Rightarrow) added_calls(m, “Log.trace”)</td>
</tr>
<tr>
<td>deleted_* (\Rightarrow) added_*</td>
<td>related code change, API replacement, etc.</td>
<td>deleted_method(m, “getHost”, t) (\Rightarrow) added_inheritedfield(“getHost”, “Svc”, t)</td>
</tr>
</tbody>
</table>
LSdiff Evaluation

- Quantitative Evaluation
- Qualitative Evaluation
- **Focus Group Study**
- Comparison with `diff`
- Comparison with `check-in comments`
- Impact of Input Parameters
LSdiff Evaluation: Research Questions

1. How often do individual changes form systematic change patterns?

2. How concisely does LSdiff describe structural differences in comparison to existing differencing approach at the same abstraction level?

3. How much contextual information does LSdiff find from unchanged code fragments?
LSdiff Evaluation: Research Questions

1. How often do individual changes form systematic change patterns? Measure coverage, # of facts in $\Delta FB$ matched by inferred rules

2. How concisely does LSdiff describe structural differences in comparison to existing differencing approach at the same abstraction level? Measure conciseness, $\Delta FB / (\#\text{ rules} + \#\text{ facts})$

3. How much contextual information does LSdiff find from unchanged code fragments? Measure the number of facts mentioned by rules but are not contained in $\Delta FB$
### Comparison with \( \Delta FB \)

<table>
<thead>
<tr>
<th></th>
<th>( \text{FB}_0 )</th>
<th>( \text{FB}_n )</th>
<th>( \Delta FB )</th>
<th>Rule</th>
<th>Fact</th>
<th>Cvg.</th>
<th>Csc.</th>
<th>Ad'l.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>10 revision pairs in carol (carol.objectweb.org)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>3080</td>
<td>3452</td>
<td>15</td>
<td>1</td>
<td>3</td>
<td>59%</td>
<td>2.3</td>
<td>0.0</td>
</tr>
<tr>
<td>Max</td>
<td>10746</td>
<td>10610</td>
<td>1812</td>
<td>36</td>
<td>71</td>
<td>98%</td>
<td>27.5</td>
<td>19.0</td>
</tr>
<tr>
<td>Med</td>
<td>9615</td>
<td>9635</td>
<td>97</td>
<td>5</td>
<td>16</td>
<td>87%</td>
<td>5.8</td>
<td>4.0</td>
</tr>
<tr>
<td>Avg</td>
<td>8913</td>
<td>8959</td>
<td>426</td>
<td>10</td>
<td>20</td>
<td>85%</td>
<td>9.9</td>
<td>5.5</td>
</tr>
<tr>
<td><strong>29 release pairs in dnsjava (<a href="http://www.dnsjava.org">www.dnsjava.org</a>)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>3109</td>
<td>3159</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0%</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Max</td>
<td>7200</td>
<td>7204</td>
<td>1500</td>
<td>36</td>
<td>201</td>
<td>98%</td>
<td>36.1</td>
<td>91.0</td>
</tr>
<tr>
<td>Med</td>
<td>4817</td>
<td>5096</td>
<td>168</td>
<td>3</td>
<td>24</td>
<td>88%</td>
<td>4.8</td>
<td>0.0</td>
</tr>
<tr>
<td>Avg</td>
<td>5144</td>
<td>5287</td>
<td>340</td>
<td>8</td>
<td>37</td>
<td>73%</td>
<td>8.4</td>
<td>14.9</td>
</tr>
<tr>
<td><strong>10 version pairs in LDiff</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>8315</td>
<td>8500</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0%</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Max</td>
<td>9042</td>
<td>9042</td>
<td>396</td>
<td>6</td>
<td>54</td>
<td>97%</td>
<td>28.9</td>
<td>12.0</td>
</tr>
<tr>
<td>Med</td>
<td>8732</td>
<td>8756</td>
<td>142</td>
<td>1</td>
<td>11</td>
<td>91%</td>
<td>9.8</td>
<td>0.0</td>
</tr>
<tr>
<td>Avg</td>
<td>8712</td>
<td>8783</td>
<td>172</td>
<td>2</td>
<td>17</td>
<td>68%</td>
<td>11.2</td>
<td>2.3</td>
</tr>
<tr>
<td><strong>three data sets above</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Med</td>
<td>6650</td>
<td>6712</td>
<td>132</td>
<td>2</td>
<td>17</td>
<td>89%</td>
<td>7.3</td>
<td>0.0</td>
</tr>
<tr>
<td>Avg</td>
<td>6632</td>
<td>6732</td>
<td>302</td>
<td>7</td>
<td>27</td>
<td>75%</td>
<td>9.3</td>
<td>9.7</td>
</tr>
</tbody>
</table>

\[ (m=3, \ a=0.75, \ k=2) \]
Focus Group Study

- **Why** would you conduct a focus group study?
- **When** do you conduct one?
- **What** can you learn from a focus group?
Focus Group Study

• **Why** would you conduct a focus group study?
  - to explore how customers will respond to a new idea
  - testing new concepts, products, and messages

• **What** can you learn from a focus group?
  - exploratory qualitative research: “thermometer” that allows you to test the “temperature” of consumers’ reactions to your research topics
  - no statistical sampling of the target population
  - less formal than a survey
  - in-depth understanding of the target’s perspectives or opinions
When to Use and When to Avoid

- When the concept or idea you wish to evaluate is new and when the best evaluation comes from letting the target customer view the concept directly.
  - e.g. new advertising campaign
- When not to do this
  - testing consumer reactions when there was no budget to accommodate changes
  - when you ask “how many...?” and “how much...?” questions or need graphs, tables, etc.
  - testing personally sensitive issues: medical conditions, politics, sex, etc.
How to conduct a focus group study

- research objectives
- recruiting profile
- screener questionnaire
- invitation to participate & co-op fee
- discussion guide
- moderator
- audio or video taping
- transcript ==> quotes
Focus Group Study

- Screener questionnaire
- Participants: five professional software engineers
  - industry experience ranging from 6 to over 30 years
  - use *diff* and *diff*-based version control system daily
  - review code changes daily except one who did weekly
- One hour structured discussion
  - I worked as a moderator. We also had a note-taker transcribe the discussion. Discussion was audio-taped and transcribed.
Focus Group Hands-On Trial

Carol Revision 430.

SVN check-in message: Common methods go in an abstract class. Easier to extend/maintain/fix
Author: benoif @ Thu Mar 10 12:21:46 2005 UTC
723 lines of changes across 9 files (2 new files and 7 modified files).

Overview

<table>
<thead>
<tr>
<th>Inferred Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (50/50)</td>
</tr>
<tr>
<td>2 (32/32)</td>
</tr>
<tr>
<td>3 (6/8)</td>
</tr>
<tr>
<td>4 (5/6)</td>
</tr>
<tr>
<td>5 (5/6)</td>
</tr>
<tr>
<td>6 (5/6)</td>
</tr>
</tbody>
</table>

Generated based on LSDiff output.

public class IIOPCosNaming extends AbsRegistry implements NameService {
    /**
     * Default port number (12350 for default)
     */
    static final int DEFAULT_PORT_NUMBER = 12350;
    /**
     * Sleep time to wait
     */
    static final int SLEEP_TIME = 2000;
    /**
     * * port number
     */
    static final int port = DEFAULT_PORT;
    /**
     * Hostname to use
     */
    private String host = null;

    All port fields in the classes that implement NameService interface got deleted except LmiRegistry.
    private int port = DEFAULT_PORT;
    /**
     * Hostname to use
     */
    private String host = null;

    All host fields in the classes that implement NameService interface got deleted except LmiRegistry.
    private String host = null;
Focus-Group Participants’ Comments

“You can’t infer the intent of a programmer, but this is pretty close.”

“This ‘except’ thing is great!”

“This is cool. I’d use it if we had one.”

“This is a definitely winner tool.”
Focus-Group Participants’ Comments

“This looks great for big architectural changes, but I wonder what it would give you if you had lots of random changes.”

“This wouldn’t be used if you were just working with one file.”

“This will look for relationships that do not exist.”
Recap

• Many differencing techniques individually compare code elements at particular granularities using similarity measures.

• Hard to comprehend as a long list of matches

• Difficult to identify exceptions that violate systematic patterns

• LSdiff uses rule-based change representations to explicitly capture systematic changes and automatically infers these rules.
Presentation on eRose

- Tileli
- Guarav
eROSE
Related Changes
(ICSE 2004, TSE 2005)

Tom Zimmermann • Saarland University
Peter Weißgerber • University of Trier
Stephan Diehl • University of Trier
Andreas Zeller • Saarland University
Harry Potter and the Half-Blood Prince (Book 6) (Hardcover)
by J. K. Rowling

List Price: CDN$ 41.00
Our Price: CDN$ 24.60 & eligible for FREE Super Saver Shipping on orders over CDN$ 39. Details
You Save: CDN$ 16.40 (40%)

Availability: In Stock. Ships from and sold by Amazon.ca.

8 used & new available from CDN$ 12.00

See larger image and other views

Find all things Harry in our Harry Potter Store.

Customers who bought this item also bought

- J.K. Rowling: Classic Books from the Library of Hogwarts School Of...
- Harry Potter and the Order of the Phoenix by J.K. Rowling
- Harry Potter and the Order of the Phoenix (Book 5) by J.K. Rowling
- Harry Potter and the Prisoner of Azkaban (Widescreen) DVD ~ Alfonso Cuarón
- Harry Potter and the Goblet of Fire (Book 4) by J.K. Rowling
- Eldest (Inheritance, Book 2) by Christopher Paolini

Explore similar items
eROSE: Guiding Developers

Customers who bought this item also bought...

Developers who changed this function also changed...

Purchase History

Version Archive
public final OverlayPreferenceStore.OverlayKey[] fKeys = new OverlayPreferenceStore.OverlayKey[] {
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, OPEN_STRUCTURE_COMPARE),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, SYNCHRONIZE_SCROLLING),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, SHOW_PSEUDO_CONFLICTS),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, SHOW_MORE_INFO),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, INITIALY_SHOW_ANCESTER_PANE),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, IGNORE_WHITESPACE),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, PREFERENCES_SAVE_ALL_EDITORS),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, USE_SPLINES),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, USE_SINGLE_LINE),
    //new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, USE_RESOLVE_UI),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.STRING, PATH_FILTER)
};

public static void initDefaults(IPreferenceStore store) {
    store.setDefault(OPEN_STRUCTURECOMPARE, true);
    store.setDefault(SYNCHRONIZE_SCROLLING, true);
    store.setDefault(SHOW_PSEUDO_CONFLICTS, false);
    store.setDefault(INITIALY_SHOW_ANCESTER_PANE, false);
    store.setDefault(SHOW_MORE_INFO, false);
    store.setDefault(IGNORE_WHITESPACE, false);
    store.setDefault(PREFERENCES_SAVE_ALL_EDITORS, false);
}
public final OverlayPreferenceStore.OverlayKey[] fKeys = new OverlayPreferenceStore.OverlayKey[] {
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, OPEN_STRUCTURE_COMPARE),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, SYNCHRONIZE_SCROLLING),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, SHOW_PSEUDO_CONFLICTS),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, SHOW_MORE_INFO),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, INITIALY_SHOW_ANCESTOR_PANE),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, IGNORE_WHITESPACE),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, USE_SPLINES),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, USE_SINGLE_LINE),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, USE_RESOLVE_UI),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.STRING, PATH_FILTER),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.STRING, AbstractTextEditor.PREFERENCE),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, AbstractTextEditor.PREFERENCE),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, SHOW_PSEUDO_CONFLICTS),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, SHOW_MORE_INFO),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, INITIALY_SHOW_ANCESTOR_PANE),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, IGNORE_WHITESPACE),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, USE_SPLINES),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, USE_SINGLE_LINE),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.BOOLEAN, USE_RESOLVE_UI),
    new OverlayPreferenceStore.OverlayKey(OverlayPreferenceStore.STRING, PATH_FILTER),
};

public static void initDefaults(IPreferenceStore store) {
    store.setDefault(OPEN_STRUCTURE_COMPARE, true);
    store.setDefault(SYNCHRONIZE_SCROLLING, true);
    store.setDefault(SHOW_PSEUDO_CONFLICTS, false);
    store.setDefault(INITIALY_SHOW_ANCESTOR_PANE, false);
    store.setDefault(SHOW_MORE_INFO, false);
    store.setDefault(IGNORE_WHITESPACE, false);
    store.setDefault(PREF_SAVE_ALL_EDITORS, false);
}
eROSE suggests further locations.
eROSE prevents incomplete changes.
Processing CVS data

1. Comparing files
2. Building transactions
Building Transactions

same author + message + time

2003-02-19 (aweinand): fixed #13332
createGeneralPage()
createTextComparePage()
fKeys[]
initDefaults()
buildnotes_compare.html
PatchMessages.properties
plugin.properties

CVS
150,000
Mining Associations

User changes fKeys[] and initDefaults()
Mining Associations

EROSE finds past transactions

- #756: fKeys[] initDefaults() ...
  plugin.properties
- #6721: fKeys[] initDefaults() ...
  plugin.properties
- #21078: fKeys[] initDefaults() ...
  plugin.properties
- #42432: fKeys[] initDefaults() ...
  plugin.properties
- #51345: fKeys[] initDefaults() ...
  plugin.properties
- #59998: fKeys[] initDefaults() ...
  plugin.properties
- #71003: fKeys[] initDefaults() ...
  plugin.properties
- #87264: fKeys[] initDefaults() ...
  plugin.properties
- #91220: fKeys[] initDefaults() ...
  plugin.properties
- #101823: fKeys[] initDefaults() ...
  plugin.properties
- #104223: fKeys[] initDefaults() ...
  plugin.properties
ERose finds past transactions

### Mining Associations

\{ f\text{Keys}[], \text{initDefaults}() \} \Rightarrow \{ \text{plugin.properties} \}

Support 10, Confidence 10/11 = 0.909
Evaluation: Research Questions

- Given a single change, can ROSE point programmers to entities that should typically be changed, too?
- Does ROSE find the missing change?
- Suppose a transaction is finished, how often does ROSE erroneously suggest that a change is missing?
Evaluation Questions:

• What are differences between course-grained vs. fine-grained suggestions?

• How well does ROSE perform if it is applied to changes without add and delete?

• What are the actual benefit of “add_to” and “del_from” items?

• How much of the version history does ROSE need?

• Would focusing on recent changes improve the quality of recommendations?
Recall: EROSE predicts 33% of all changed entities.

Likelihood: In 70% of all transactions, EROSE’s topmost three suggestions contain a changed entity.

EROSE learns quickly (within 30 days).
Evaluation Measure

Precision: a fraction of the returned items that were expected

\[ P_q = \frac{|A_q \cap E|}{|A_q|} \]

Recall: a fraction of expected items that were returned

\[ R_q = \frac{|A_q \cap E|}{|E|} \]

Where \( A_q \) is a set of items recommended by querying with \( q \).

\( E \) is a set of items in the evaluation data (ground truth).
A recommendation is neutral when a method in \( f \), other than the one that was changed in \( f \), has a direct fine-grained reference to a method, field, or class in \( f \), or a class that was changed in \( f \) has a weak coarse-grained relationship—it indirectly inherits from, or is in the same package or directory that has more than 20 files—with a class that was changed in \( f \).

If \( f \) and \( f \) have more than one relationship, the interestingness value of the recommendation is determined by the interestingness value of the most obvious relationship.

4.2 Validation Settings

Fig. 1 presents some metrics about the Eclipse and Mozilla developments and outlines the portions of the development history we considered in our analysis. In both systems, the training data comprised changes to over 20,000 files and over 100,000 versions to those source files.

Table 2 shows the number of transactions involving different cardinalities of files as well as the total number of transactions. For the period of time that corresponds to the training data, both Eclipse and Mozilla have a similar number of transactions. In both systems, transactions of two items have the highest counts and the number of transactions decreases as the cardinality of the transaction increases.

Table 3 describes the parameters we used in the data mining algorithm. The first column lists the support threshold. The second column indicates whether the data mining algorithm was applied to Eclipse and Mozilla. The third column presents the number of files that were generated from the patterns extracted using the algorithm with the specified parameter applied to either Eclipse or Mozilla.

For the frequent pattern algorithm, the value of the support threshold \( \text{min}\_\text{support} \) was varied so that a reasonably large number of files (over 200) were involved in patterns and the support was not too low (not below 5). Comparing the patterns generated for Eclipse and Mozilla using the frequent pattern algorithm with the same parameter setting (\( \text{min}\_\text{support} \) equals 20 and 15), Mozilla has more than five times more files involved in the change patterns than Eclipse. We were careful to choose thresholds that were neither too restrictive nor too relaxed. An overly restrictive threshold results in too few patterns. This situation affects the recall value as the recommendations do not cover the changes needed for a modification task. An overly relaxed threshold affects the precision since too many patterns result in a number of recommendations, only a few of which are correct.
Evaluation Measure

Feedback: $|Z^*|/|Z|$
the percentage of queries where eRose makes at least one recommendation

$Z^* = \{ q \mid q = (Q, E) \in Z, apply_{R_1}(Q) \neq \emptyset \}$,

Likelihood:
a probability that at least one of the top k recommendations for a query is correct

$L_k = \frac{|\{ q \mid q = (Q, E) \in Z, apply_{R_k}(Q) \cap E \neq \emptyset \}|}{|\{ q \mid q = (Q, E) \in Z, apply_{R_k}(Q) \neq \emptyset \}|}$

Where $A_q$ is a set of items recommended by querying with $q$.
$E$ is a set of items in the evaluation data (ground truth)
Quiz

- You cannot discuss your solution with your classmates
- It will be graded (scale of 0-3)
My general thoughts on eRose & Recap

- eRose uses association rule mining to identify related code elements from a version history data.
- The approach & idea is very novel, though the results are not very impressive.
- One of the first practical system that recovers institutional knowledge from history data
- Trade offs between precision vs. recall is thoroughly investigated.
Announcement

- Project Checkpoint Due on this Thursday.
  - I won’t grade them.
  - It is not mandatory.
  - You are encouraged to submit to seek my feedback.
  - Available for both research project, literature survey, and tool evaluation
Preview for Next Monday


- Focus on how they integrated heterogeneous software artifact repositories

- Look at their user study design: what else would you have done to evaluate this system?

- If time permits, briefly go over BugCache (S.Kim et al. ICSE 2007) & Social Structure Mining (C. Bird et al. FSE 2008) papers.