Blending Conceptual and Evolutionary Couplings to Support Change Impact Analysis in Source Code

Huzefa Kagdi  Winston-Salem State, NC
Malcom Gethers and Denys Poshyvanyk  The College of William and Mary, VA
Michael Collard  The University of Akron, OH
Research Goal

• Develop a new and improved Impact Analysis (IA) approach by combining existing solutions
  – Information Retrieval (IR), Latent Semantic Indexing (LSI)
  – Mining Software Repositories (MSR), Itemset Mining

• Employ single + multiple version analysis

```c++
/**
 * Append a new page, creating followup frames (but not headers/footers),
 * and return the page number.
 * @param masterPageName the name of the master page to use for this new page.
 */
KWPage appendPage(const QString &masterPageName = QString());
```


Changed paths:
  M /trunk/KDE/kdegames/ktron/ktron.cpp
  M /trunk/KDE/kdegames/ktron/main.cpp

move scope
Impact Analysis in Source Code

Fix a bug

Estimates

Approach

Impact Set

```c
// reallocate.
void Vector::realloclate(unsigned newsize, bool copy)
{
    // resize to the given size
    double* temptr = new double[newsize];
}
```
Combined Approach for IA

- **Step 1: Select** the first software entity for which IA needs to be performed

- **Step 2: Compute** conceptual couplings with IR methods from the release (source code comments and identifiers) of a software system in which the first entity is selected

- **Step 3: Mine** evolutionary couplings from the source code change history (commits in the source code repository), occurred before the release in Step 2

- **Step 4: Compute** the estimated impact set from the disjunctive combination (union) of couplings computed in steps 3 and 4
Motivating Example

- In *Apache httpd* commit# 888310 addresses the bug#47087
- Three source code files were changed
  - /modules/http/http_filters.c
  - /modules/http/http_protocol.c
  - /server/protocol.c

<table>
<thead>
<tr>
<th>Rank</th>
<th>Conceptual</th>
<th>Evolutionary</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>/server/protocol.c</td>
<td>/modules/http/byterange_filter.c</td>
<td>/server/protocol.c</td>
</tr>
<tr>
<td>2</td>
<td>/modules/proxy/mod_proxy_http.c</td>
<td>/modules/http/http_protocol.c</td>
<td>/modules/proxy/mod_proxy_http.c</td>
</tr>
<tr>
<td>3</td>
<td>/modules/debugging/mod_bucateer.c</td>
<td>/modules/proxy/mod_proxy_ftp.c</td>
<td>/modules/http/byterange_filter.c</td>
</tr>
<tr>
<td>4</td>
<td>/server/core_filters.c</td>
<td>/server/core.c</td>
<td>/modules/http/http_protocol.c</td>
</tr>
<tr>
<td>5</td>
<td>/modules/http/byterange_filter.c</td>
<td>/include/ap_mmn.h</td>
<td>/server/core_filters.c</td>
</tr>
</tbody>
</table>
Extracting Semantic Information with LSI

- Source code (release) -> Corpus (doc = method)
- Preprocessing: split_identifiers & splitIdentifiers
- Vector space = term-by-document matrix
- Singular Value Decomposition -> LSI subspace

<table>
<thead>
<tr>
<th>document</th>
<th>shape</th>
<th>set</th>
<th>frame</th>
<th>view</th>
</tr>
</thead>
<tbody>
<tr>
<td>KWDocument::addShape</td>
<td>1 10 6 15 5</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KWDocument::removeShape</td>
<td>1 7 5 16 6</td>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Computing Conceptual Similarity

Cosine between document vectors [Poshyvanyk, Marcus]

<table>
<thead>
<tr>
<th></th>
<th>KWDocument::addShape</th>
<th>KWDocument::removeShape</th>
<th>KWDocument::insertPage</th>
</tr>
</thead>
<tbody>
<tr>
<td>KWDocument::addShape</td>
<td>1</td>
<td>0.78</td>
<td>0.24</td>
</tr>
<tr>
<td>KWDocument::removeShape</td>
<td>0.78</td>
<td>1</td>
<td>0.27</td>
</tr>
<tr>
<td>KWDocument::insertPage</td>
<td>0.24</td>
<td>0.27</td>
<td>1</td>
</tr>
</tbody>
</table>

void KWDocument::addShape(KWShape* shape)
{
    // Add a shape to the document.
    // ...}

void KWDocument::removeShape(KWShape* shape)
{
    // Remove a shape from the document.
    // ...}

void KWDocument::insertPage()
{
    // Insert a page into the document.
    // ...}

⇒
0.78
Development Versions

\[ \delta_i \rightarrow \delta_k \rightarrow \delta_n \]

Fix a bug

Commit

\[ \text{Change-set} \]

\[
\begin{align*}
\text{Fix a bug} & \quad \Rightarrow \\
\delta_i & \quad = \\
\delta_k & \quad = \\
\delta_n & \quad = \\
\end{align*}
\]

\[
\begin{align*}
\text{r472031} & \quad | \quad \text{bmeyer} \quad | \quad 2005-10-19 \\
11:14:47 & \quad -0400 \quad (\text{Wed, 19 Oct} \quad 2005) \quad | \quad 1 \quad \text{line}
\end{align*}
\]

\[ \text{Changed paths:} \]

- \[ /\text{trunk/KDE/kdegames/ktron/ktron.cpp} \]
- \[ /\text{trunk/KDE/kdegames/ktron/main.cpp} \]
Evolutionary Couplings

Software entities that typically co-change in the version archives [Gall’98, Zimmermann’04]

Evolutionary Pattern

{Button.java, GUI.java}

co-changed in 3 commits (support value)

with itemset mining

Change prediction rules from association rules

{GUI.java} ⇒ {Button.java, GUI.java} with a 75% probability, i.e., confidence value
Procedure for Computing Impact Set

DisjIA

Start Entity

Cut Point/2.0

Cut Point/2.0

ConceIA

CC

EvolIA

EC

EI

EM

| EI U EM |
< cut point

Yes

No

Conceptual Couplings (CC) are ranked by cosine similarity values
Evolutionary Couplings (EC) are ranked by confidence and support values
Case Study

• Four open source software systems, namely *Apache httpd, ArgoUML, iBatis*, and *KOffice*

• Widely used metrics precision (false positives) and recall (false negatives) used for accuracy measure

• Commits in the version history used as ground truth, i.e., actual impact sets

• Accuracy assessment performed for various estimate impact set sizes
Research Questions

• **RQ1**: Does combining conceptual and evolutionary couplings improve the accuracy of IA when compared to the two standalone techniques?

• **RQ2**: Does the choice of granularity, *i.e.*, file or method, affect the accuracy of IA of standalone techniques and their combination?
## Evaluation Dataset

<table>
<thead>
<tr>
<th>System</th>
<th>Release History</th>
<th>LSI Indexed Release</th>
<th># of Commits Train (test)</th>
<th># of Entities Train (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache (httpd)</td>
<td>2.2.9-2.3.5</td>
<td>2.2.3</td>
<td>1736 (287)</td>
<td>2086 (982)</td>
</tr>
<tr>
<td>ArgoUML</td>
<td>0.24-0.28</td>
<td>0.28</td>
<td>3375 (773)</td>
<td>4217 (621)</td>
</tr>
<tr>
<td>iBatis</td>
<td>3.0.0-190_b1 - 3.0.0-240_b10</td>
<td>3.0.0-216</td>
<td>108 (40)</td>
<td>461 (118)</td>
</tr>
<tr>
<td>KOffice 2.0.91</td>
<td>2.0.0-2.0.91</td>
<td>2.0.91</td>
<td>2749 (522)</td>
<td>5580 (1072)</td>
</tr>
<tr>
<td>KOffice 2.0.1</td>
<td>2.0.0-2.0.2</td>
<td>2.0.1</td>
<td>763 (255)</td>
<td>1233 (533)</td>
</tr>
<tr>
<td>KOffice 2.0.1*</td>
<td>2.0.0-2.0.2</td>
<td>2.0.1</td>
<td>577 (192)</td>
<td>5530 (1438)</td>
</tr>
</tbody>
</table>

* Denotes method level granularity processed with srcDiff [Collard’03]
Null Hypotheses

- $H_{0\text{CP}}$: Combining conceptual and evolutionary couplings does not significantly improve precision results of impact analysis compared to conceptual couplings.
- $H_{0\text{CR}}$: Combining conceptual and evolutionary couplings does not significantly improve recall results of impact analysis compared to conceptual couplings.
- $H_{0\text{EP}}$: Combining conceptual and evolutionary couplings does not significantly improve precision results of impact analysis compared to evolutionary couplings.
- $H_{0\text{ER}}$: Combining conceptual and evolutionary couplings does not significantly improve recall results of impact analysis compared to evolutionary couplings.
## Results of Wilcoxon Signed-Rank Test

<table>
<thead>
<tr>
<th>System</th>
<th>Granularity</th>
<th>$H_{0\ CP}$</th>
<th>$H_{0\ CR}$</th>
<th>$H_{0\ EP}$</th>
<th>$H_{0\ ER}$</th>
<th>Null Hypothesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache (httpd)</td>
<td>File</td>
<td>0.0002</td>
<td>0.0003</td>
<td>0.0001</td>
<td>0.0003</td>
<td>Rejected</td>
</tr>
<tr>
<td>ArgoUML</td>
<td>File</td>
<td>0.0050</td>
<td>0.0039</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>Rejected</td>
</tr>
<tr>
<td>iBatis</td>
<td>File</td>
<td>0.0126</td>
<td>0.0126</td>
<td>0.0001</td>
<td>0.0002</td>
<td>Rejected</td>
</tr>
<tr>
<td>KOffice 2.0.91</td>
<td>File</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>Rejected</td>
</tr>
<tr>
<td>KOffice 2.0.1</td>
<td>File</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>Rejected</td>
</tr>
<tr>
<td>KOffice 2.0.1*</td>
<td>Method</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>Rejected</td>
</tr>
</tbody>
</table>

In all cases considered for our dataset we obtained a p-value less than 0.05, indicating that the improvement in accuracy obtained is not by chance.
Threats to Validity

• Commits used as gold standard for accuracy computation
  – Not all the entities in a commit maybe related to a single change request
  – All the entities related to a single change request maybe present in a single commit
  – Developer established actual change-sets

• Granularity levels of file and method

• Statistically significant results for the four open source system may not generalize
Related Work

- Various dependency-analysis methods are already investigated in the literature
  - Call graphs, program slicing, hidden dependency analysis [Chen, Rajlich, Yu], Lightweight static analysis approaches [Moonen], concept analysis [Tonella], dynamic analysis [Law], hypertext systems, documentation systems, UML models [Briand], and Information Retrieval [Antoniol]
  - Coupling measures have been also used to support impact analysis in OO systems [Briand, Wilkie]
  - Comparison of different impact analysis algorithms [Osro]

- An IR (single version) and MSR (multiple version) combination has not been investigated for IA previously
Conclusions

• Combining conceptual and evolutionary couplings does improve accuracy of IA

• Recall improvements of up to 20% over the conceptual technique in *KOffice* and up to 45% over the evolutionary technique in *iBatis*

• Varying granularity levels does impact accuracy of individual methods; however, combining conceptual and evolutionary couplings maintains the accuracy gains
Future Work

• Devise and empirically validate other combinations (e.g., weighed contributions of entities from each coupling based on the amount of change history considered)
• Include static and dynamic analysis information, and application of IR on multi-version artifacts (e.g., commit messages and bug reports)
• Provide IA support beginning from a high-level textual change request
• Comparative studies with other approaches (e.g., structural metrics)
• Work supported by NSF CCF-1016868 and NSF CCF-1016887 grants

• Questions?