Experiments with JavaScript
Clone Detection

Wai Ting Cheung

Visiting Student, HKUST
@ PLRG, KAIST
Jan 30, 2013
Table of Contents

• Introduction of Code Clones
• Experiment Overview
• Results
• Conclusion
Introduction of Code Clones
Code Cloning

- Copying code fragments and reusing them with or without modification

- 7% to 23% of the code in a typical software system has been cloned. (Baker, WCRE 1995; Roy, WCRE 2008)
Previous Works

- Empirical Study of Code Clones
  - Cai, FASE 2011; Roy, WCRE 2008, Kim, FSE 2005

- Survey of Clone Detection Research
  - Pate, JSEP 2011; Roy, Technical Report 2007

- Evaluation of Clone Detection Tools
  - Roy, SCP 2009; Bellon, TSE 2007

- Study of Clones in Script Languages / Web Applications
  - Roy, IWSC 2010; Basit, ICWE 2005; Calefato, JWE 2004
Use of JavaScript

• 98 out of 100 most popular websites use JavaScript (Guarnieri, ISSTA11)

• Use of dynamic features is evident in websites (Richards, ECOOP11, PLDI10; Zorn, WebApps10)
Experiment Overview
Research Questions

• What are the main differences in code clone properties in Javascript web, Javascript standalone, and Java projects?

• How many consistent and inconsistent code clones are in Javascript web, Javascript standalone, and Java projects?
Experiment Subjects

- 18 subjects in total, 6 each in
  - JavaScript in web pages
  - JavaScript standalones
  - Java projects
- web development framework, GUI framework, build tool, etc.
Tools for Clone Detection

• Tree-based clone detection
  • **SAFE** (for JavaScript)
    • Formal Specification and Implementation of a Scalable Analysis Framework for ECMAScript, FOOL 2012
  • **Deckard** (for Java)
    • Scalable and Accurate Tree-based Detection of Code Clones, ICSE 2007
Metrics

- Clone localization
- Size of cloned code
- Clone coverage
- Files associated with clones
- Consistent / inconsistent function clones / cloned fragments
Clone Localization

• Same file and same directory
• Same directory but different files
• Different directories
• The location of a clone pair is a factor in software maintenance (Kapser, ELISA 2003)
Size of Cloned Code

• Average lines of cloned code
• Maximum lines of cloned code
• Give information about the scale of the cloned code in a system
Clone Coverage

- The ratio of cloned code to the total lines of code
- An increase in the number of clones over time can indicate a decline in the structure and maintainability of a software system (Barbour et al, 2012)
Files Associated with Clones

• A file is associated with clones if it has at least one method that forms a clone pair with another method in the same file or a different file.

• It tells us that whether the clones are from some specific files, or scattered among many files all over the system.

• From a maintenance point of view, a lower value is better, since clones localized to certain specific files may be easier to maintain (Roy, IWSC 2010).
Inconsistent Clones

• A substring $s$ of the code is called an inconsistent clone, if there is another substring $t$ of the code such that their edit distance is below a given threshold and that $t$ has no significant overlap with $s$ (Juergens, ICSE 2009)

• Half of the changes to code clone groups are inconsistent changes (Krinke et al, 2007)

• Inconsistent changes to clone groups are directly related to the maintenance problems (e.g. bug-fixing or update) (Roy et al, 2007)
Inconsistent Clones

gestureTouchesDragged: function(evt, touches) {
    var gestures = this.get("gestures"), idx, len = gestures.length, g;
    for (idx = 0; idx < len; idx++) {
        g = gestures[idx];
        g.unassignedTouchesDidChange(evt, touches);
    }
},

gestureTouchEnd: function(touch) {
    var gestures = this.get("gestures"), idx, len = gestures.length, g;
    for (idx = 0; idx < len; idx++) {
        g = gestures[idx];
        g.unassignedTouchDidEnd(touch);
    }
}
Function Clones

• Entire functions are copied rather than fragments

• A high number of function clones in a software system could increase significantly the cost of maintenance (Lague et al, 1997)

• Finding function clones in scripted web pages for the purpose of eliminating duplicated code can be seen as a first step to introduce refactoring (Calefato et al, 2004)
Results
Clone Localization

<table>
<thead>
<tr>
<th>Subject</th>
<th>Same File, Same Dir.</th>
<th>Same Dir, Diff. Files</th>
<th>Diff. Dir.</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaScript in Web Pages</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JavaScript Standalone</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Java Projects</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Size of Cloned Code

<table>
<thead>
<tr>
<th></th>
<th>Average Lines of Cloned Code</th>
<th>Standard Deviation</th>
<th>Maximum Lines of Cloned Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaScript in Web Pages</td>
<td>10.50</td>
<td>2.95</td>
<td>262</td>
</tr>
<tr>
<td>JavaScript Projects</td>
<td>15.33</td>
<td>11.00</td>
<td>550</td>
</tr>
<tr>
<td>Java Projects</td>
<td>12.33</td>
<td>4.46</td>
<td>299</td>
</tr>
</tbody>
</table>
Clone Coverage

![Box plot showing coverage percentages for different contexts: JavaScript in Web Pages, JavaScript Standalone, and Java Projects.](image_url)
Consistent / Inconsistent Function Clones / Cloned Fragments

Subject
- JavaScript in Web Pages
- JavaScript Standalone
- Java Projects
Threats to Validity

• Representativeness of open source projects and websites
• Only a single configuration is used
• Only two languages are used
Summary

- Most of the clones of JavaScript in web pages are from different directories
- JavaScript standalone has the lowest coverage and files associated with clones
- JavaScript in web pages contains the largest amount of consistent clones
Conclusion

• We have conducted clone detection experiments on properties of different projects and found that they are indeed different.

• The differences are clues to which systems require more efforts in software maintenance.

• Future work: Automatic refactoring.
Thank You