Reuse

SWE 795, Spring 2017
Software Engineering Environments

Partially adapted from Kerry Chang, Finding Code to Reuse, 05-899D Human Aspects of Software Development
Today

• Part 1 (Discussion)(15 mins)
  • Preventing defects reading from last week

• Part 2 (Lecture)(~70 mins)

• Break!

• Part 3 (Discussion)(~60 mins)
  • Discussion of readings
What is reuse?

• Making use of previously written code rather than writing new code

• Often, reuse takes form of reusing a *library* or a *framework*

• Once made choice to reuse a library or framework, need to understand how to achieve specific behavior with library or framework
  • Often finding code *snippets* that achieve desired behavior
Reuse of Uses

- Developers rely extensively on *examples* to understand how to instantiate objects.

<table>
<thead>
<tr>
<th>Reuse Activity</th>
<th>Specific Strategies Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finding a Usage Context</td>
<td>Find senders of messages defined for target class, focusing on “interesting” ones</td>
</tr>
<tr>
<td>Evaluating a Usage Context</td>
<td>Look for references to application data objects in the <code>openOn:</code> method.</td>
</tr>
<tr>
<td>Executing the Context</td>
<td>Open example application “on” a basic data object from the project.</td>
</tr>
<tr>
<td>Assessing Similarity</td>
<td>Reason by analogy from familiar syntactic construction, e.g., <code>button1Down:</code></td>
</tr>
<tr>
<td>Studying Bits of Context</td>
<td>Look for use of unmappable instance variables or many messages to “self.”</td>
</tr>
<tr>
<td>Deciding to Subclass</td>
<td>Focus first on the <code>openOn:</code> code for starting up a window.</td>
</tr>
<tr>
<td>Getting an Instance Running</td>
<td>Adapt or develop the method identified in the notification “message not understood.”</td>
</tr>
</tbody>
</table>

Some possible reuse strategies

• Read the documentation

• Find tutorials

• Find StackOverflow snippets

• Find similar code in your own codebase

• Call API functions, see what they return
Opportunistic vs. systematic

• Opportunistic developers more likely to start with example code

• Systematic developers more likely to read the documentation first

Steven Clarke.
Reuse topics today

• How do developers do it?
  • What are the key steps? What challenges do developers face?
  • Challenges working with APIs

• Techniques for supporting software reuse
  • Browsing APIs
  • Code search
Example reuse process

B: read tutorials, articles, projects to understand domain

D: searched Google, often seeking descriptions in API of specific classes & methods to use

E: looked for examples of how to use specific methods

Types of reuse

<table>
<thead>
<tr>
<th>WEB SESSION INTENTION:</th>
<th>LEARNING</th>
<th>CLARIFICATION</th>
<th>REMINDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reason for using Web</td>
<td>Just-in-time learning of unfamiliar concepts</td>
<td>Connect high-level knowledge to implementation details</td>
<td>Substitute for memorization (e.g., language syntax or function usage lookup)</td>
</tr>
<tr>
<td>Web session length</td>
<td>Tens of minutes</td>
<td>~ 1 minute</td>
<td>&lt; 1 minute</td>
</tr>
<tr>
<td>Starts with web search?</td>
<td>Almost always</td>
<td>Often</td>
<td>Sometimes</td>
</tr>
<tr>
<td>Search terms</td>
<td>Natural language related to high-level task</td>
<td>Mix of natural language and code, cross-language analogies</td>
<td>Mostly code (e.g., function names, language keywords)</td>
</tr>
<tr>
<td>Example search</td>
<td>“ajax tutorial”</td>
<td>“javascript timer”</td>
<td>“mysql_fetch_array”</td>
</tr>
<tr>
<td>Num. result clicks</td>
<td>Usually several</td>
<td>Fewer</td>
<td>Usually zero or one</td>
</tr>
<tr>
<td>Num. query refinements</td>
<td>Usually several</td>
<td>Fewer</td>
<td>Usually zero</td>
</tr>
<tr>
<td>Types of webpages visited</td>
<td>Tutorials, how-to articles</td>
<td>API documentation, blog posts, articles</td>
<td>API documentation, result snippets on search page</td>
</tr>
<tr>
<td>Amount of code copied from Web</td>
<td>Dozens of lines (e.g., from tutorial snippets)</td>
<td>Several lines</td>
<td>Varies</td>
</tr>
<tr>
<td>Immediately test copied code?</td>
<td>Yes</td>
<td>Not usually, often trust snippets</td>
<td>Varies</td>
</tr>
</tbody>
</table>

Types of reuse

• Learning—relies on selecting highest quality tutorials tutorials
  • e.g., “update web page without reloading php”
• Clarification—learning syntax based on exiting understanding of the domain concepts
  • e.g., reminding use of syntax of HTML forms
  • Often search by analogy to domain concepts in other languages / frameworks
    • e.g., Perl has a function to format dates as strings, what’s the one for PHP?
• Reminder—using web as external memory aid
  • e.g., forgot a word in a long function name
  • e.g., 6 lines of code necessary to connect and disconnect from MySQL database copied hundreds of times by individual

Design implications

• Web tutorials used for just in time learning
  • —> Tutorials should be tightly coupled to code, where developers can play in sandbox then read tutorial content to understand problems when do not work
• Web search used as translator from intention to terminology & syntax
  • —> tools could compare code from search results to users code to automatically locate errors
  • —> search should be integrated into autocomplete
• Developers delay testing, esp for routine functionality
  • —> Tools should assist with adaption by highlighting variables and literals in reused snippets & provide link back to original source
Challenges with reuse

- Mapping an abstract conceptual solution into the appropriate elements
  - “How do I create a rectangle? Why is there no Rectangle tool?”
- Understanding control & data flow, hidden dependencies due to runtime binding or inheritance, between classes in the API
  - “I’m over-riding SelectionTool, and in particular mouseDown() so that when the figure is clicked the box is drawn. This bit works, however when trying to drag the figure, if I do something similar the rectangle flickers like mad.”
- Understanding how functionality works
  - “How does ... work?”, “What does ... do?” or, “Where is ... defined/created/called?”
- Making changes consistent w/ architectural constrains of API
  - Violating constraints of MVC architecture by passing references in prohibited ways

Challenges with reuse

• **Design** barriers—inhernent cognitive difficulties of the programming problem, separate from notation used
  • I don’t know what I want the computer to do

• **Selection** barriers—finding programming interfaces available to achieve a particular behavior
  • I don’t know what to use

• **Coordination** barriers—constraints governing how languages & libraries can be combined
  • I don’t know how to make them work together

• **Use** barriers—determining how API how to use API
  • I don’t know how to use it

• **Understanding** barriers—environment properties such as compile & runtime errors that prevent seeing behavior
  • It didn’t do what I expected

• **Information** barriers—environment properties that prevent understanding runtime execution state
  • I think I know why didn’t behave as expected, but don’t know how to check
Vocabulary problem

• Developers are familiar with concepts using one set of terminology.
• API, tutorials, or other resources use different terminology
• How do developers find the right concepts with alternative terms?
Challenges may vary by context

• Size of desired snippet
  • Reusing a line of code? A whole algorithm?

• Alternatives
  • How many alternatives are there? How important is it to find the best alternative?

• Integration
  • What libraries or frameworks does a snippet require? How can they be integrated?
Challenges working with API documentation

• Goal: Parse a Java source file w/ Eclipse
• Answer:

```java
IFile file = ...;
ICompilationUnit cu =
    JavaCore.createCompilationUnitFrom(file);
ASTNode ast = AST.parseCompilationUnit(cu, false);
```

• Challenges
  • Want to work with files and ASTNodes, but key class is JavaCore
  • No connection from what you might know about ASTNode and IFile to JavaCore
Techniques for reuse

• Browsing API documentation

• Searching with …. 

• Integrating search into autocomplete

• Enabling sample adaption & exploratory programming w/ examples
Filtering & browsing documentation

Apatite: A New Interface for Exploring APIs

Daniel S. Eisenberg, Jeffrey Stylos, and Brad A. Myers

Carnegie Mellon University
Indexing OSS projects in code search

## Searching by inputs and outputs

<table>
<thead>
<tr>
<th>Name</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Tokenizer</td>
<td>“this is a test”</td>
<td>[“this”, “is”, “a”, “test”]</td>
</tr>
<tr>
<td>Quote Tokenizer</td>
<td>“this is a test”</td>
<td>[“this”, “is”, “a”, “test”]</td>
</tr>
<tr>
<td></td>
<td>“this is a ‘test with’ quoted &quot;string types&quot; in it”</td>
<td>[“this”, “is”, “a”, “test with”, “quoted”, “string types”, “in”, “it”]</td>
</tr>
<tr>
<td>Robots.txt</td>
<td>“<a href="http://www.cs.brown.edu/people/spr%E2%80%9D">http://www.cs.brown.edu/people/spr”</a></td>
<td>true</td>
</tr>
<tr>
<td></td>
<td>“<a href="http://www.cnn.com/topics%E2%80%9D">http://www.cnn.com/topics”</a></td>
<td>true</td>
</tr>
<tr>
<td></td>
<td>“<a href="http://www.nytimes.com/college/students%E2%80%9D">http://www.nytimes.com/college/students”</a></td>
<td>false</td>
</tr>
<tr>
<td>Log2</td>
<td>0</td>
<td>RuntimeException</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>5</td>
</tr>
<tr>
<td>To Roman</td>
<td>13</td>
<td>xiii</td>
</tr>
<tr>
<td>From Roman</td>
<td>VIII</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>xxvi</td>
<td>26</td>
</tr>
<tr>
<td>Primes</td>
<td>5</td>
<td>true</td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>false</td>
</tr>
<tr>
<td></td>
<td>59</td>
<td>true</td>
</tr>
<tr>
<td>Perfect Numbers</td>
<td>6</td>
<td>true</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>false</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>true</td>
</tr>
<tr>
<td>Day of Week</td>
<td>“08/07/08”</td>
<td>“Thursday”</td>
</tr>
<tr>
<td>Easter</td>
<td>2008</td>
<td>new Date(108,2,23)</td>
</tr>
</tbody>
</table>

LaToza

GMU SWE 795 Spring 2017

Look for: METHOD In Remote Archives Using Google

Description: roman numeral

Method

Declaration: java.lang.String convert(int n)

Tests:

Find it!

Results:

Order By: Code Size Format Using: Brown

Source: programs/roman-numerals/bio98q1.java @ http://www.olympiad.org.uk

static String[] hundreds = {"", "c", "cc", "ccc", "cd", "d", "dc", "dccc", "cm"};

static String[] tens = {"", "x", "xx", "xxx", "xl", "lx", "lxx", "lxxx", "xc"};

static String[] thousands = {"", "m", "mm", "mmm"};

static String[] units = {"", "i", "ii", "iii", "iv", "v", "vi", "vii", "viii", "ix"};

public static String convert(int n)
{
    return (thousands[(n/1000)] + hundreds[(n/100)%10] + tens[(n/10)%10] + units[(n%10)].toUpperCase());
}

Source: XQuisitor/saxonb8-4+Folder/net/sf/saxon/number/Numberer_en.java @ http://www.cafeconleche.org/xquisitor/xquisitor-1.0a5.zip

License
Searching by input and output types

Mine Jungoloids describing paths by which types can be converted

Searching by output

Figure 1. With d.mix, users browse web sites through a proxy that marks API-accessible content. Users select marked elements they wish to copy. Through a site-to-service map, d.mix composes web service calls that yield results corresponding to the user’s selection. This code is copied to the d.mix wiki for editing and hosting.

http://dl.acm.org/citation.cfm?doid=1294211.1294254

Searching for instantiation snippets

• Classes are often created through factories rather than constructors, making construction snippets harder to find

• Integrate construction snippet search into autocomplete

Labeling snippets with keywords

- Problem: how do you ensure that there’s high quality labels explaining the intention of code snippets?
- Idea: enable search from keywords to code and from code to keywords
- Log associations to support future queries

Grouping diverse search results

Adapt snippets

SnipMatch Demonstration

Doug Wightman¹, Zi Ye¹, Joel Brandt², Roel Vertegaal¹

¹Human Media Lab, Queen’s University
Kingston, ON, K7L 3N6, Canada
{wightman, zi, roel}@cs.queensu.ca

²Advanced Technology Labs, Adobe
San Francisco, CA 94103
joel.brandt@adobe.com

Adapt snippets

Link code back to snippets

https://www.youtube.com/watch?v=_PYvPlv4OQw