Code Reuse

SWE 795, Fall 2019
Software Engineering Environments

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

• Part 1 (Lecture)(~60 mins)

• Part 2: (HW 4 Checkpoint presentations)(45 mins)
• Break!

• Part 3 (Discussion)(~45 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><strong>Executing the Context</strong>&lt;br&gt;Look for references to application data objects in the <code>openOn:</code> method.</td>
</tr>
<tr>
<td></td>
<td><strong>Assessing Similarity</strong>&lt;br&gt;Open example application “on” a basic data object from the project.</td>
</tr>
<tr>
<td></td>
<td><strong>Studying Bits of Context</strong>&lt;br&gt;Reason by analogy from familiar syntactic construction, e.g., <code>button!Down:</code></td>
</tr>
<tr>
<td></td>
<td><strong>Deciding to Subclass</strong>&lt;br&gt;Look for use of unmappable instance variables or many messages to “self.”</td>
</tr>
<tr>
<td>Debugging a Usage Context</td>
<td><strong>Getting an Instance Running</strong>&lt;br&gt;Focus first on the <code>openOn:</code> code for starting up a window.</td>
</tr>
<tr>
<td></td>
<td><strong>Borrowing the Context</strong>&lt;br&gt;Use multiple browsers to work from related pieces of context. Carry out step-by-step replacement of message parameters. Edit what does not compile. Develop a method to substitute one data object for another.</td>
</tr>
<tr>
<td></td>
<td><strong>Analysis by Testing</strong>&lt;br&gt;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
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>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Search terms</th>
<th>Natural language related to high-level task</th>
<th>Mix of natural language and code, cross-language analogies</th>
<th>Mostly code (e.g., function names, language keywords)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example search</td>
<td>“ajax tutorial”</td>
<td>“javascript timer”</td>
<td>“mysql_fetch_array”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Num. result clicks</th>
<th>Usually several</th>
<th>Fewer</th>
<th>Usually zero or one</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num. query refinements</td>
<td>Usually several</td>
<td>Fewer</td>
<td>Usually zero</td>
</tr>
</tbody>
</table>

| Types of webpages visited | Tutorials, how-to articles | API documentation, blog posts, articles | API documentation, result snippets on search page |

<table>
<thead>
<tr>
<th>Amount of code copied from Web</th>
<th>Dozens of lines (e.g., from tutorial snippets)</th>
<th>Several lines</th>
<th>Varies</th>
</tr>
</thead>
<tbody>
<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

Joel Brandt, Philip J. Guo, Joel Lewenstein, Mira Dontcheva, and Scott R. Klemmer. 2009. Two studies of
opportunistic programming: interleaving web foraging, learning, and writing code. *Conference on Human Factors
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 run-time 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—inhherent 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
Some techniques for supporting reuse

• New ways to start code search
  • Browse API documentation to find right class for the task
  • Support searching for examples across existing codebases
  • Find the right sequence of methods to complete some task you already started

• Helping understand code examples
Apatite: A New Interface for Exploring APIs

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

Carnegie Mellon University
Indexing OSS projects for code search

Grouping diverse search results

Calcite: Offering virtual methods matching user expectations

## Searching by inputs and outputs

<table>
<thead>
<tr>
<th>Name</th>
<th>Test Cases</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>
<td></td>
</tr>
<tr>
<td>Quote Tokenizer</td>
<td>“this is a test”</td>
<td>“this”, “is”, “a”, “test”</td>
<td></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>
<td></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>
<td></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>
<td></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>
<td></td>
</tr>
<tr>
<td>Log2</td>
<td>0</td>
<td>RuntimeException</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>To Roman</td>
<td>13</td>
<td>xiii</td>
<td></td>
</tr>
<tr>
<td>From Roman</td>
<td>VIII</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xxvi</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Primes</td>
<td>5</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td></td>
<td>59</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>Perfect Numbers</td>
<td>6</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>Day of Week</td>
<td>“08/07/08”</td>
<td>“Thursday”</td>
<td></td>
</tr>
<tr>
<td>Easter</td>
<td>2008</td>
<td>new Date(108,2,23)</td>
<td></td>
</tr>
</tbody>
</table>

Searching by input and output types

<table>
<thead>
<tr>
<th>Programming problem</th>
<th>$\tau_{in}$</th>
<th>$\tau_{out}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read lines from an input stream (Tester)</td>
<td>InputStream</td>
<td>BufferedReader</td>
</tr>
<tr>
<td>Open a named file for memory-mapped I/O (Almanac)</td>
<td>String</td>
<td>MappedByteBuffer</td>
</tr>
<tr>
<td>Get table widget from an Eclipse view (FAQs)</td>
<td>javaworkbench</td>
<td>Table</td>
</tr>
<tr>
<td>Get the active editor (Eclipse FAQs)</td>
<td>ScrollingGraphicalViewer</td>
<td>IEditorPart</td>
</tr>
<tr>
<td>Retrieve canvas from scrolling viewer (Author)</td>
<td>KeyEvent</td>
<td>FigureCanvas</td>
</tr>
<tr>
<td>Get window for MessageBox (Author)</td>
<td>Enumeration</td>
<td>Shell</td>
</tr>
<tr>
<td>Convert legacy class (Author)</td>
<td></td>
<td>Iterator</td>
</tr>
</tbody>
</table>

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**

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

Integrating code search into autocomplete

Commercial tool, extension built for Adobe Flex Builder

Offering autocomplete suggestions based on most frequent completions

Commercial tool: Kite

Intelligent Snippets

Write code faster by tabbing through code snippets that Kite recommends based on your code base. Learn more.

https://kite.com/
Helping understand code examples
(a) A micro-explanation of a CSS selector with an automatically generated natural language explanation and demonstration of an HTML element it matches.

(b) A micro-explanation describing the high-level intent and low-level argument values of a `wget` command.

A. Head, C. Appachu, M. A. Hearst and B. Hartmann, "Tutorons: Generating context-relevant, on-demand explanations and demonstrations of online code," 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Atlanta, GA, 2015, pp. 3-12. doi: 10.1109/VLHCC.2015.7356972
Connecting snippets back to documentation

Figure 1: HyperSource associates Web pages visited by programmers with subsequent code edits.

Figure 2: Web history (L) associated with the current line of code (R). Gutter highlights provide scent.

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

Interactive Extraction of Examples from Existing Code

Figure 1: Extracting example code from existing code with CodeScoop. With CodeScoop, (1) a programmer selects a few lines they want to share from a source program, and CodeScoop helps them build them into a complete, compilable example. To help programmers make complete examples, CodeScoop detects errors and recommends fixes by (2a) pointing to potentially missing code and (2b) suggesting literal values from the program trace that can take the place of variables. (3) It also recommends code the programmer may have overlooked, like past variable uses and nearby control structures.

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