Code Reuse

CS 695 / SWE 699: Programming Tools
Fall 2023
Today

- Part 1 (Lecture) (~45 mins)
- Part 2: Tech Talks (60 mins)
  - Two tech talks
  - 10 min break
  - Two tech talks
- Part 3: (Short In-Class Activity) (30 mins)
Logistics

- HW 4 checkpoint due 11/1
- HW 4 due 11/29
- Tech talks should now all be on the dates originally scheduled
Overview

• What is code reuse, how developers do it, and what makes it hard?
• Tools to support code reuse
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><strong>Finding a Usage Context</strong></td>
<td>Find senders of messages defined for target class, focusing on “interesting” ones</td>
</tr>
<tr>
<td><strong>Evaluating a Usage Context</strong></td>
<td></td>
</tr>
<tr>
<td>Executing the Context</td>
<td>Look for references to application data objects in the openOn: method.</td>
</tr>
<tr>
<td>Assessing Similarity</td>
<td>Open example application “on” a basic data object from the project.</td>
</tr>
<tr>
<td>Studying Bits of Context</td>
<td>Reason by analogy from familiar syntactic construction, e.g., button1Down:</td>
</tr>
<tr>
<td>Deciding to Subclass</td>
<td>Look for use of unmappable instance variables or many messages to “self.”</td>
</tr>
<tr>
<td><strong>Debugging a Usage Context</strong></td>
<td></td>
</tr>
<tr>
<td>Getting an Instance Running</td>
<td>Focus first on the openOn: code for starting up a window.</td>
</tr>
<tr>
<td>Borrowing the Context</td>
<td>Use multiple browsers to work from related pieces of context. Carry out step-by-step</td>
</tr>
<tr>
<td></td>
<td>replacement of message parameters. Edit what does not compile. Develop a method to substitute</td>
</tr>
<tr>
<td></td>
<td>one data object for another.</td>
</tr>
<tr>
<td>Analysis by Testing</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
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 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— inherent 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
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 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>
</tr>
</thead>
</table>
| **Simple Tokenizer** | “this is a test”  

Input: [“this”, “is”, “a”, “test”]  

Output: [“this”, “is”, “a”, “test”]  

Quote Tokenizer | “this is a test”  

Input: [“this”, “is”, “a”, “test”]  

Output: [“this”, “is”, “a”, “test”]  

Input: “this is a ‘test with’ quoted "string types!" in it”  

Output: [“this”, “is”, “a”, “test with”, “quoted”, “string types”, “in”, “it”]  

Robots.txt | “http://www.cs.brown.edu/people/spr”  

Output: true  

“http://www.cnn.com/topics”  

Output: true  

“http://www.nytimes.com/college/students”  

Output: false  

Log2 | 0  

Output: RuntimeException  

1  

2  

5  

To Roman | 13  

Output: xiii  

VIII  

8  

xxvi  

26  

From Roman |  

Primes | 5  

Output: true  

39  

false  

59  

true  

Perfect Numbers | 6  

Output: true  

12  

false  

28  

true  

Day of Week | “08/07/08”  

Output: “Thursday”  

Easter | 2008  

Output: new Date(108,2,23)

---

Look for: METHOD In Remote Archives Using Google

Description: roman numeral

Method

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

Tests:

Result:

Order By: Code Size Format Using: Brown

Source: programs/roman-numerals/2008st.java @ http://www.olympus.org.uk

static String[] hundreds = { "", "x", "xx", "xxx", "cd", "d", "dc", "dxc", "cm" };
static String[] tens = { "", "x", "xx", "xxx", "xl", "l", "lx", "lxx", "lx", "lxx" };
static String[] thousands = { "", "m", "mm", "mmm" };
static String[] units = { "", "i", "ii", "iii", "iv", "v", "vi", "vii", "viii", "ix" };

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

Source: X:scans/saxon-8-4+Folder/net/saxon/number/Numberer_en.java @ http://www.cafe-conetti.com/aquistor/aquistor-1.0a2.zip
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

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

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.

Use $m$ suffix for megabytes (-limit-rate=$1m$). The above command will limit the retrieval rate to 50KB/s. It is also possible to specify disk quota for automatic retrievals to avoid disk DoS attack. The following command will be aborted when the quota is (100MB+) exceeded.

```
$ wget -c -b -o /tmp/download.log -i /tmp/download.txt --quota=100m
```

You found a `wget` command.

`wget` is a Terminal command you run to download a page from the Internet. Here, it downloads content from URLs from the file '/tmp/download.txt'.

It uses these options:
- --continue (-c): resume getting a partially-downloaded file.
- --background (-b): go to background after startup.
- --output-file (-o): log messages to /tmp/download.log.
- --input-file (-i): download URLs found in local or external /tmp/download.txt.
- --quota (-q): set retrieval quota to 100m.

(b) A micro-explanation describing the high-level intent and low-level argument values of a `wget` command.
if (value == null) {
    return def;
} else {
    try {
        return Integer.parseInt(value.trim());
    } catch (NumberFormatException nf) {
        ...
    }
}
```java
public class ExtractedExample {
    public static void main(String[] args) {
        Cursor cursor = database.cursor();
        List titles = new ArrayList();
        try {
            cursor.execute("SELECT id, title, year, num_pages FROM table WHERE 
            int COLUMN_INDEX_ID = 0;
            int COLUMN_INDEX_TITLE = 1;
            int COLUMN_INDEX_YEAR = 2;
            int COLUMN_INDEX_NUM_PAGES = 3;
            boolean DEBUG = true;
            Database database = new Database("Lou", "PASSWORD", "https://acme- 
            Cursor cursor = database.cursor();
            Booklist booklist = new Booklist();
            List titles = new ArrayList();
            try {
                cursor.execute("SELECT id, title, year, num_pages FROM table WHERE 
                if (cursor.rowCount() > 0) {
                    int rowNumber = 0;
                    while (finished == false) {
                        int rowCount = cursor.rowCount();
                        for (int i = 0; i < Math.min(rowCount, maxBooks); ++i) {
                            cursor.fetchone();
                            int id = cursor.getInt(COLUMN_INDEX_ID);
                            String title = cursor.getString(COLUMN_INDEX_TITLE);
                            int year = cursor.getInt(COLUMN_INDEX_YEAR);
                            int num_pages = cursor.getInt(COLUMN_INDEX_NUM_PAGES);
                            Book book = new Book(id, title, year, num_pages);
                            if (title != null) {
                                titles.add(title);
                            }
                            if (id != -1) {
                                boolean bestseller = isBestseller(book.getId());
                            }
                        }
                    }
                }
            } catch (ConnectionException e) {
```

← Do you want any of those uses of "cursor"? No
Tech Talks