What is Reuse?

It is not copy and paste... its Reuse!
Always Reuse?

Been here... done this before!
Always Reuse?

Been here... done this before!

But...
Reuse Requirements

• Creating and maintaining a reuse repository - takes commitment, effort and investment.

• Enabling software developers to build new software systems with components from the reuse repository.
Cognition Issues in Reuse

Task-relevant Information
Cognition Issues in Reuse

L4: Entire Information Space

L4-L3: Unanticipated Information

Task-relevant Information
Cognition Issues in Reuse

L4: Entire Information Space

L3: Belief

L2: Vaguely Known

L1: Well Known

L4-L3: Unanticipated Information

Task-relevant Information
CodeBroker delivers task relevant and personalised components...
CodeBroker delivers task relevant and personalised components...
CodeBroker delivers task relevant and personalised components...
CodeBroker delivers task relevant and personalised components...

- Interface end
- Back-end search engine

responds to comments

e.g. `// generate random numbers`
CodeBroker delivers task relevant and personalised components...

Interface end

Back-end search engine

responds to comments
e.g. // generate random numbers

clues in method signatures
e.g. void randomNumberGenerator (int x)
Comment: Create a random number between two limits

Signature: int <- int x int

Inferred queries

system updating

User model

Discourse model

L4: Component Repository

L3

L2

left-click

right-click

Get Int

public static int getInt(int lo, int hi)

Generate a random number using the default generator

Get Long

See Also:

Java documentation
UltiGPX
A tool for visualising GPS data collected by hikers
UltiGPX
A tool for visualising GPS data collected by hikers
UltiGPX

UltiGPX is a tool for visualizing Global Positioning System (GPS) data. It is designed to help developers visualize and analyze GPS data, making it easier for them to make informed decisions. The tool is particularly useful for managing and analyzing large amounts of GPS data, as it provides a visual interface for navigating and manipulating the data.

The UltiGPX interface includes several key features, such as a toolbar for file management, a display area for visualizing GPS data, and a properties panel for analyzing the data. The display area can show a variety of visualizations, including maps and graphs, that can help developers understand the data and make better decisions.

Overall, UltiGPX is a powerful tool for developers who need to visualize and analyze GPS data. Its intuitive interface and comprehensive features make it an ideal choice for anyone looking to make the most of their GPS data.
Azureus
Azureus
Azureus
Azureus

Should I reuse or not?
Classic Code Searching
Azureus
Azureus

By manual inspection, user falls upon SpeedGraphic class (322 LOC) in package org.gudy.azureus2.ui.swt.components.graphics.
By manual inspection, user falls upon SpeedGraphic class (322 LOC) in package org.gudy.azureus2.ui.swt.components.graphics.

- 14 different types references by drawChart() (82 LOC).
- 7 were common to both, Azureus and UltiGPX.
- Remaining 7 types need to be investigated carefully to identify their dependencies.
What decisions were made?
Reuse Process
Reuse Process

Identification
Reuse Process

Identification

Delineation
Reuse Process

Identification

Delineation

Extraction
Gilligan

a tool to support software reuse...
What does Gilligan do?
What does Gilligan do?

Supports navigation

Annotate structurally dependent code fragments
What does Gilligan do?
What does Gilligan do?

Feasibility Assessment
How does Gilligan work?
How does Gilligan work?

Find project
How does Gilligan work?

1. Find project
2. Select starting point
How does Gilligan work?

1. Investigate feature dependencies.
2. Evaluate reuse plan.
How does Gilligan work?

1. Investigate feature dependencies.
2. Evaluate reuse plan.
How does Gilligan work?

1. Investigate feature dependencies.
2. Evaluate reuse plan.

Find project
Select starting point
Gillian
Perform reuse plan
How does Gilligan work?

1. Investigate feature dependencies.
2. Evaluate reuse plan.

Find project
Select starting point
Perform reuse plan
Abort Investigation
How does Gilligan work?

1. Investigate feature dependencies.
2. Evaluate reuse plan.

Find project → Select starting point → Gillian

Abort Investigation

Perform reuse plan
How does Gilligan work?

1. Investigate feature dependencies.
2. Evaluate reuse plan.

Abort Investigation

Find project
Select starting point

Gillian

Re-implement feature

Perform reuse plan
How does Gilligan work?
How does Gilligan work?
How does Gilligan work?

Search for dependencies

- Population size
  - I+ (Immigrated+)
  - I- (Immigrated-)
  - I+ (Immigrated+)
- Born-
- Dead-
- Emigrated-
How does Gilligan work?
How does Gilligan work?
How does Gilligan work?

Starting point
(package, class or method)

Relevant dependencies
Screenshot of Gilligan
Screenshot of Gilligan
Azureus Reuse Plan
Azureus Reuse Plan

Figure 4. Azureus graphics feature reuse plan. (Screenshot is annotated for greyscale reproduction.)

4.1. Past research topic [7].

29th International Conference on Software Engineering (ICSE'07)
Azureus Reuse Plan

Extraction and integration required 708 LOC (reused). Within these, 2 methods and 5 LOC were commented.
<table>
<thead>
<tr>
<th>#</th>
<th>Question</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Somewhat Agree</th>
<th>Neutral</th>
<th>Somewhat Disagree</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I have reused source code</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>I have reused whole classes</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>I have reused whole features</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>I would rather reimplement a feature than reuse an existing one</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>I reuse code to save time</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>I reuse code to increase reliability</td>
<td>4</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>Keeping track of the relevant details of a piece of source code while</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>navigating its text can be difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Understanding what dependencies a feature has on its context is</td>
<td>7</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>important for me to determine whether I should reuse it</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>My organization has a large amount of code available to be reused</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>Portions of features I am developing already exist</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Evaluation - Case Study

From SWT framework (65 kloc in 458 classes), extract...

BMP implementation

PNG implementation
Evaluation - Case Study

BMP extraction

PNG extraction
Evaluation - Case Study

BMP extraction

PNG extraction
Evaluation - Case Study

- Started with `WinBMPFileFormat`.
- Led him to `LEDataInputStream` and `ImageData`.
- Reused 497 LOC.
- Of 14,081 nodes in SWT, only 60 visited.
- 38 accepted, 16 rejected, remapped 2 and 4 already provided.

BMP extraction

PNG extraction
Evaluation - Case Study

BMP extraction

- Started with WinBMPFileFormat.
- Led him to LEDataInputStream and ImageData.
- Reused 497 LOC.
- Of 14,081 nodes in SWT, only 60 visited.
- 38 accepted, 16 rejected, remapped 2 and 4 already provided.

PNG extraction

- Started with a 471 LOC class.
- Of 92 nodes on screen, 20 were relevant to him.
- He marked 20 accepted, 2 rejected and 1 remapped.
- Eventually he reused 23 classes (approx 3 kloc).
- Changed mind often!
### Java™ Platform, Standard Edition 6

API Specification

This document is the API specification for version 6 of the Java™ Platform, Standard Edition.

See: [Description](#)

#### Packages

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.applet</td>
<td>Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.</td>
</tr>
<tr>
<td>java.awt</td>
<td>Contains all of the classes for creating user interfaces and for painting graphics and images.</td>
</tr>
<tr>
<td>java.awt.color</td>
<td>Provides classes for color spaces.</td>
</tr>
<tr>
<td>java.awt.datatransfer</td>
<td>Provides interfaces and classes for transferring data between and within applications.</td>
</tr>
<tr>
<td>java.awt.dnd</td>
<td>Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.</td>
</tr>
<tr>
<td>java.awt.event</td>
<td>Provides interfaces and classes for dealing with different types of events fired by AWT components.</td>
</tr>
<tr>
<td>java.awt.font</td>
<td>Provides classes and interface relating to fonts.</td>
</tr>
<tr>
<td>java.awt.geom</td>
<td>Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.</td>
</tr>
<tr>
<td>java.awt.im</td>
<td>Provides classes and interfaces for the input method framework.</td>
</tr>
<tr>
<td>java.awt.im.spi</td>
<td>Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.</td>
</tr>
<tr>
<td>java.awt.image</td>
<td>Provides classes for creating and modifying images.</td>
</tr>
</tbody>
</table>
To ease the burden of the developer, Strathcona uses the structure of the source code to find relevant examples in a repository.
Example Task

The graphical overview and textual rationale description can be used by the developer to quickly locate the message object and the context of the error. The Eclipse IDE has the ability to display the error, along with additional information such as the file name, line number, and error code. This allows the developer to quickly and easily fix the error.

Fig. 2. API documentation for IStatusLineManager.
Example Task

API Documentation for IStatusLineManager

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
</table>
| ```
void setMessage(String message)

Sets the message text to be displayed on the status line.
``` |
Client: Determining Structural Context
Client: Determining Structural Context

Extract structural contexts
Client: Determining Structural Context

- declaring type is View
- declares method updateStatusMessage
- extends org.eclipse.ui.part.ViewPart
- ...

Extract structural contexts
Server Repository

Conditions for addition

- Code must be parsable by Eclipse compiler.
- Code must represent good usage of API.

Structural contexts are stored in PostGRES tables.
Output Examples
Code Search Engines

SPARS

koders

Google

Docjar
MAPO
MAPO

query
MAPO

open source repositories

query

I. code search engine
MAPO

open source repositories → search engine → source files

query
MAPO

1. code search engine

2. code analyser

open source repositories

source files

call sequences

query
MAPO

1. code search engine

2. code analyser

3. sequence processor

open source repositories

query

source files

call sequences

query
MAPO

1. code search engine

2. code analyser

3. sequence processor

open source repositories

query

call sequences

call sequences

source files
MAPO

1. Code search engine

2. Code analyser

3. Sequence processor

4. Frequent sequence miner

Open source repositories

Query

Source files

Call sequences

Call sequences
MAPO

1. code search engine
2. code analyser
3. sequence processor
4. frequent sequence miner

open source repositories

query

frequent sequences

call sequences

source files

call sequences

query

open source repositories
MAPO

1. code search engine

2. code analyser

3. sequence processor

4. frequent sequence miner

5. frequent sequence processor

open source repositories

query

frequent sequences

call sequences
MAPO

1. code search engine
2. code analyser
3. sequence processor
4. frequent sequence miner
5. frequent sequence processor

open source repositories

query

frequent sequences

source files

call sequences

API Usages

call sequences