Tutorial 4
More on Refactoring

How to refactoring unstructured code?
How to apply refactoring in Eclipse?
Last lecture…

On refactoring

• We explained what is refactoring, what is software refactoring
• How are they related to other restructuring techniques?
• Examples of refactoring
• Refactoring structured source code into goal models
• …
Today…

1. How to refactoring unstructured code into goal models?
2. How to use Eclipse to do refactoring?
3. Discussions
4. Relation to your course project
1. Refactoring an unstructured program

• The subject is called “Squirrel Mail”
• It has 70 KLOC
• Developed in PHP
  Function call
  Foo.php: <?php include("bar.php") ?>
• Why it is unstructured?
  Foo.php: <a href="bar.php”/>  
    <a href="moo.php”/>  
    <?php echo “I won super 7!” ?>

Any idea?
Why a PHP program is unstructured?

• Every Hyperlink generated from the PHP is an “exit” in the current PHP program
• It may call other PHP routines, other web pages, etc. when user click at them
• Non-deterministic, how could you tell which link will the user click?
• Even “go back” button will change the behaviour of the program
• So …
The process

- Structured program is easier to understand

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Extract Methods Refactor</td>
</tr>
<tr>
<td>2.</td>
<td>Construct Hammock graph</td>
</tr>
<tr>
<td>3.</td>
<td>Extract States/Transitions Refactoring</td>
</tr>
<tr>
<td>4.</td>
<td>Eliminate GOTO</td>
</tr>
<tr>
<td>5.</td>
<td>Parse</td>
</tr>
<tr>
<td>6.</td>
<td>Restructure</td>
</tr>
<tr>
<td>7.</td>
<td>Test to identify NFRs</td>
</tr>
<tr>
<td>8.</td>
<td>Analyze Quality metrics to identify softgoals</td>
</tr>
</tbody>
</table>

Legacy source code → Refactored source code

- Yes: Is structured?
- No: Construct Hammock graph

AST (annotated Goal graph) → Structured program → High-level Statechart → Statecharts

Goal Model → Goal Model with NFRs → Goal Model with Softgoals

Spring 2005 ECE450H1S Software Engineering II
1.1 Refactoring based on comments

Variables defined before the entry of the block
\[ I = (I_1 \cup I_2) \cap \{ v | \text{def} (v, p) \land p \leq p_{\text{entry}} \} \neq \emptyset \]

Variables defined in the block that will be used after the exit
\[ O = (O_1 \cup O_2) \cap \{ v | \text{use} (v, p) \land p_{\text{exit}} \leq p \} \neq \emptyset \]
Example

/** Path for SquirrelMail required files. */
define('SM_PATH','../');
require_once($SM_PATH . 'functions/strings.php');
require_once($SM_PATH . 'config/config.php');
require_once($SM_PATH . 'functions/i18n.php');
require_once($SM_PATH . 'functions/plugin.php');
require_once($SM_PATH . 'functions//constants.php');
require_once($SM_PATH . 'functions/page_header.php');
require_once($SM_PATH . 'functions/html.php');
require_once($SM_PATH . 'functions/global.php');
require_once($SM_PATH . 'functions/imap_general.php');

$SM_PATH=set_path ();
Further ...

```php
<?php /* login.php */
$SM_PATH=set_path();
$SM_lang=setup_language();
$base_uri = findout_base_URI();
$logindisabled = detect_imap_server($base_uri);
if ($logindisabled) {
    explain_situation();
    exit;
}
do_hook('login_cookie');
$header = onload_function("redirect.php");
display_header($header);
load_theme($theme[$theme_default]);
do_hook('login_top');
show_logo();
show_form($loginname, $mailto, $key);
do_hook('login_form');
do_hook('login_bottom');
?>
```
1.2 Convert into statechart

- Statecharts concisely describe behaviour of a system.
- No comments now, but we need to understand its behaviour, therefore …

```bash
$SM_PATH=set_path();
```

- $SM_PATH is undefined
- $SM_PATH is set
- /set_path

```
$SM_PATH = set_path();
```

- /set_path
- $SM_PATH
1.3 Statechart refactoring

Extract Method -> Extract States and Transitions based on Hammock graphs

What’s new here? You are refactoring behaviour rather than structures!
1.4 Put it together … the high-level statechart of the unstructured program
1.4 Now convert statechart back into a program with GOTO’s

- FORTRAN
  
  call EnterURL
  10 call Login
     if (wrongIMAP) goto 30
  20 call ShowForm
     if (wrongKey) goto 20
     call EnterForm
     if (wrongForm) goto 30
  call StartWebMail
     if (loggedOut) goto 10
     if (expired) goto 10
  call Send
  Stop
  30 call ReportError
  call GoBack
  goto 10
  end

- Rule of thumb: every state is a basic block; adding a label to states with multiple incoming transitions; adding GOTO statements for all outgoing transitions except one; line-up the basic blocks
1.4 Eliminate GOTO’s

- FPT (Fortran parallelizing transformer, developed at ELIS, Ghent University, Belgium)
- Result of goto elimination:

CALL EnterURL
REPEAT
REPEAT
CALL Login
IF (.not.wrongIMAP) THEN
REPEAT
CALL ShowForm
UNTIL (.not.wrongKey)
CALL EnterForm
IF (.not.wrongForm) THEN
CALL StartWebmail
ENDIF
ENDIF
UNTIL (.not.loggedOut.or .not.expired.or.wrongIMAP.or.wrongForm)
IF(wrongIMAP.or.wrongForm)
THEN
CALL ReportError
CALL GoBack
ENDIF
UNTIL (.not.wrongIMAP.and.not.wrongForm)
CALL Send
END
1.5 Turning structured program into an annotated goal model
1.6 Turning it into “pure” goal model (AND/OR graph)

- call A
- call B

- if (x) then
  - call A
- else
  - call B
- end if

- call A
- repeat
- call C
- until s
- call B

Spring 2005  ECE450H1S  Software Engineering II
1.7 Introducing softgoals

- Identify NFRs
- Add softgoals to categorize why there are the NFRs
- If possible, one can measure the degree of satisfaction for the softgoals
2. How to refactoring in Eclipse

• If you are developing in Java, you are LUCKY!
• The Eclipse IDE, JBuilder IDE are very comprehensive
• Refactoring was developed in Smalltalk, now moved to Java in Eclipse, it has been told in C# for Visual Studio, etc.
• It should not be long to see open-source programming languages to have them supported, such as PHP
• Examples, developed by Jing Su
Example 1 – extract method

```java
void f() {
    ...
    // Compute score
    score = a * b + c;
    score -= discount;
}
```

```java
void f() {
    ...
    computeScore();
}
```

```java
void computeScore() {
    score = a * b + c;
    score -= discount;
}
```
public class ExtractMethodExample {
  void f() {
    int a, b, c, score, discount;
    a = b = c = 10;
    discount = 20;
    // Compute score
    score = a * b + c;
    score -= discount;
  }
}
Enter a name according to the comment
```java
public class ExtractMethodExample {
  void f() {
    int a, b, c, score, discount;
    a + b + c = 10;
    discount = 20;
    ComputeScore(a, b, c, discount);
  }

  private void ComputeScore(int a, int b, int c, int discount) {
    int score;
    // Do something
    score = a * b + c;
    score -= discount;
  }
}
```
Example 2 – move method

```java
class Jar {
    
    
}

class RoboPacker {
    private bool isFragile(Jar foo) {
        switch(foo.material) {
            case GLASS: return true;
            case WOOD: return true;
            case TIN: return false;
        }
    }
}

class Jar {
    bool isFragile() {
        switch(material) {
            case GLASS: return true;
            case WOOD: return true;
            case TIN: return false;
        }
    }

class RoboPacker {
    private bool isFragile(Jar foo) {
        return foo.isFragile();
    }
}
```
```java
/*
 * Created on Nov 18, 2004
 */

/**
 * @author Jing Su
 */

public class MoveMethodExample {
    class Jar {
        private static final int GLASS = 0;
        private static final int BOTTLE = 1;
        private static final int TIN = 2;
        public int material;
    }

    class RoboFisher {
        
        private boolean isFragile(Jar foo) {
            switch(foo.material) {
            case foo.GLASS: return true;
            case foo.BOTTLE: return true;
            case foo.TIN: return false;
            }
            return false;
        }
    }
}
```
```java
public class RoboPacker {
    private boolean isFragile(Jar foo) {
        switch (foo.material) {
            case Jar.GLASS: return true;
            case Jar.WOOD: return true;
            case Jar.TIN: return false;
        }
    }
}
```

```java
public class RoboPacker {
    private boolean isFragile(Jar foo) {
        return foo.isFragile();
    }
}
```
```java
/*
 * Created on Nov 18, 2004
 */

/*
 * Author: Jing Su
 */

public class MoveMethodExample {
    class Jar {
        private static final int GLASS = 0;
        private static final int WOOD = 1;
        private static final int Tin = 2;

        public int material;

        private boolean isFragile() {
            switch (material) {
            case MoveMethodExample.Jar.GLASS: return true;
            case MoveMethodExample.Jar.WOOD: return true;
            case MoveMethodExample.Jar.TIN: return false;
            }
            return false;
        }
    }

    class RoboDasher {

        private boolean isFragile(Jar foo) {
            return foo.isFragile();
        }
    }
}
```
Example 3 – lift method

class Jar {
    bool isFragile() {
        switch(material) {
            case GLASS:
                // complex glass calculation
            case WOOD:
                // complex wood calculation
            case TIN:
                // complex tin calculation
        }
    }
}

class Jar {
    bool isFragile() {
        return material.isFragile();
    }
}

interface Material {
    ...}

class GlassMaterial:Material {
    ...}
class WoodMaterial:Material {
    ...}
class TinMaterial:Material {
    ...}
```java
public class MoveMethodExample {
    interface Material {
        public boolean isFragile();
    }

    class Glass implements Material {
        public boolean isFragile() {
            return true;
        }
        
    } Glass Wood implements Material {
        public boolean isFragile() {
            return true;
        }
    }

    class Tin implements Material {
        public boolean isFragile() {
            return false;
        }
    }

    class Jar {
        public Material material;
        public boolean isFragile() {
            return material.isFragile();
        }
    }

    class RobotForker {
        private boolean isFragile(Jar foo) {
            return foo.isFragile();
        }
    }
}```
3. Think about these …

1. How to extend refactoring tool support to other programming languages such as PHP?

2. Can you extend refactoring to documents, such as in various formats: diagrams, textual, xml, etc.?

3. How can know a function is NFR? Can you measure the impact of a NFR on a quality attribute?
4. Relation to your project

• Opportunities:
  – You may add junit test cases to the code base to reveal bugs (publish it to the bug tracking system) and fix them (+5%)
  – *You may apply design patterns, refactoring techniques on this legacy code base, showing as an improved complexity metrics (+2.5%)*
  – You may tune the performance of the system to speed up the display, load/save for scalable graphs (+2.5%)

• Don’t forget your major project task (up to 100%!)
  – To study the editor methods in the OpenOME and adapt them to the OmniGraphEditor web service.