A Catalog of Refactorings to Discipline Preprocessor-Based Annotations

Márcio Ribeiro
http://www.ic.ufal.br/marcio
marcio@ic.ufal.br
@marciomribeiro
```
this.dragon = Resources.getImage(Resources.IMG_DRAGON);
```

```
private int[] xClouds = new int[]{20, 80, 40, 60};

private int[] xClouds = new int[]{40, 140, 80, 110};
```
$2^n$

\[ n = \{\text{CLOUD, RAIN}\} \]

Products = [{}, {CLOUD}, {RAIN}, {CLOUD, RAIN}]
One different product to each person in the world

Sven Apel, Classifying and Unifying Product-Line Analyses
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit. <?> for Help, </> for Search. Legend: [*] built-in  [] excluded  <M> module  < > module capable

[*] 64-bit kernel
   General setup --->
   [*] Enable loadable module support --->
   [*] Enable the block layer --->
      Processor type and features --->
      Power management and ACPI options --->
      Bus options (PCI etc.) --->
      Executable file formats / Emulations --->
   [*] Networking support --->
      Device Drivers --->
      Firmware Drivers --->
      File systems --->
      Kernel hacking --->

v(+)
How to Implement?
void test() {
   // code 1
   #ifdef CLOUD
      // code 2
   #endif
   #ifdef RAIN
      // code 3
   #endif
   // code 4
}

!CLOUD and !RAIN
CLOUD and !RAIN
!CLOUD and RAIN
Simple technique

Solve portability problems

Maintenance problems

 Poor understandability

Subtle to errors

No overheads

#ifdefs
Undisciplined vs. Disciplined Annotations

```c
#if (WIN32)
if ((ready = select(0, NULL, tp))
#else
if ((ready = select(max_fd + 1, NULL, tp))
#endif
== -1) {
    ngx_log_error(NGX_LOG_ALERT);
    return NGX_ERROR;
    if (ready == -1) {
        err = ngx_socket_errno;
    } else {
        err = 0;
    }
}

#endif (WIN32)
ready = select(0, NULL, tp);
#else
ready = select(max_fd + 1, NULL, tp);
#endif
if (ready == -1) {
    ngx_log_error(NGX_LOG_ALERT);
    return NGX_ERROR;
    if (ready == -1) {
        err = ngx_socket_errno;
    } else {
        err = 0;
    }
}
```
Previous work...
ness and response time. Our results indicate that the discipline of annotations has no influence on program comprehension and maintenance, neither for correctness nor for performance (in terms of response time). Although we observed some tendencies, they are not supported by our statistical analysis. However, our experiment
#if defined(FEAT_XCLIPBOARD) || defined(USE_XSMP) || defined(FEAT_MZSCHEME)
    static int busy = FALSE;
#endif

#if defined(HAVE_GETTIMEOFDAY) && defined(HAVE_SYS_TIME_H)
    if (msec > 0 && (
        #ifdef FEAT_XCLIPBOARD
            xterm_Shell != (Widget)0
        #ifdef USE_XSMP
            ||
        #endif
        #ifdef FEAT_MZSCHEME
            ||
        #endif
        #ifdef USE_XSMP
            xsmp_icefd != -1
        #ifdef FEAT_MZSCHEME
            ||
        #endif
        #endif
        #ifdef FEAT_MZSCHEME
            (mzthreads_allowed() && p_mzq > 0)
        #endif
    
    gettimeofday(&start_tv, NULL);
#endif

#if defined(HAVE_GETTIMEOFDAY) && defined(HAVE_SYS_TIME_H)
    if (msec > 0 && (
        #ifdef FEAT_XCLIPBOARD
            xterm_Shell != (Widget)0
        #ifdef USE_XSMP
            ||
        #endif
        #ifdef FEAT_MZSCHEME
            ||
        #endif
        #ifdef USE_XSMP
            xsmp_icefd != -1
        #ifdef FEAT_MZSCHEME
            ||
        #endif
        #endif
        #ifdef FEAT_MZSCHEME
            (mzthreads_allowed() && p_mzq > 0)
        #endif
    
    gettimeofday(&start_tv, NULL);
#endif

    if (busy)
        return 0;
#endif
"Prefer to compile out entire functions, rather than portions of functions or portions of expressions."
We found the opposite
Disciplined

Undisciplined

Minutes

20 40 60 80 100

Significant difference!
Significant difference!
Refactoring Undisciplined Annotations
1. #if defined (USE_ISPTS_FLAG)
2.  if (result) {
3.  #endif
4.   result = open("/dev/ptmx");
5.  if (!result)
6.    strcpy(ttydev);
7. #ifdef USE_ISPTS_FLAG
8.  }
9. #endif

1. #if defined (USE_ISPTS_FLAG)
2.  if (result) {
3.   result = open("/dev/ptmx");
4.  if (!result)
5.    strcpy(ttydev);
6.  }
7. #else
8.   result = open("/dev/ptmx");
9.  if (!result)
10.   strcpy(ttydev);
11. #endif

❌ Code clone
❌ Lines of code
## Catalogue of Refactorings

<table>
<thead>
<tr>
<th>Bad Smells</th>
<th>Refactoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Else inside directives</td>
<td>Else if wrappers</td>
</tr>
<tr>
<td>Incomplete if and while statements</td>
<td>Conditional statement wrappers</td>
</tr>
<tr>
<td>Case inside directives</td>
<td>Case wrappers</td>
</tr>
<tr>
<td>Alternative choice</td>
<td>Alternative statement wrappers</td>
</tr>
<tr>
<td>Incomplete else</td>
<td>If statements ending with an else</td>
</tr>
<tr>
<td>Incomplete conditions</td>
<td>Conditions</td>
</tr>
<tr>
<td>Incomplete commands</td>
<td>Returns and expressions</td>
</tr>
<tr>
<td>Incomplete arrays and enums</td>
<td>Data structure definitions</td>
</tr>
<tr>
<td>Incomplete function definitions</td>
<td>Function definitions</td>
</tr>
</tbody>
</table>

...  

14 refactorings
1. `if (condition1`  
2. `#ifdef` `expression1`  
3. `&& condition2`  
4. `#endif`  
5. `) {`  
6. `// Statements..`  
7. `}`

Precondition: original code is not using the variable `test` in this scope

---

1. `bool test = condition1;`  
2. `#ifdef` `expression1`  
3. `test = test && condition2;`  
4. `#endif`  
5. `if (test) {`  
6. `// Statements..`  
7. `}`

Code clone

Lines of code
Precondition: original code is not using the variable test in this scope

- Green checkmark: Code clone
- Green checkmark: Lines of code +2
1. type function(
2.   #ifdef expression1
3.   type param_id
4.   #endif
5. ) {
6.   // Statements..
7. }

Precondition: original code does not define the macro PARAM in this scope

1. #ifdef expression1
2.   #define PARAM type param_id
3. #else
4.   #define PARAM ""
5. #endif
6. type function(PARAM) {
7.   // Statements..
8. }

✔ Code clone
✔ Lines of code
❓ Code understanding
Evaluating the Refactorings

- Frequency of application possibilities
- Opinion of developers
- Behavior preservation
- Code cloning

63 Program Families
<table>
<thead>
<tr>
<th></th>
<th>1972</th>
<th>905</th>
<th>357</th>
<th>178</th>
<th>270</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>73</td>
<td>87</td>
<td>23</td>
<td>31</td>
<td>19</td>
</tr>
</tbody>
</table>

7K application possibilities
mfp = open(mf
#ifdef TS
    , (mode_t) 0600
#else
    , IR | IW
#endif
#endif
mfp = open(mf, (mode_t) 0600);
#else
mfp = open(mf_name, IR | IW);
#endif

Strongly prefer A  | Prefer A  | Doesn't matter  | Prefer B  | Strongly prefer B
1                    | 14        | 8               | 73        | 94
A

```c
bool exec = (bit < 8);
#ifdef TS
exec = exec && (r != NULL);
#endif
if (exec) {
    // STMTS
}
```

B

```c
if (bit < 8
#ifdef TS
    && r != NULL
#endif
) {
    // STMTS
}
```

67% vs 20%
```c
#ifdef TS
#define PARAM Ct server,
#else
#define PARAM
#endif
void msgNetbeans(PARAM XT client) {
    // STMTS
}

void msgNetbeans(Ct server,
#ifndef TS
#define PARAM
#else
#endif
XT client) {
    // STMTS
}
```

**Survey Results**

- **Strongly Prefer A**: 56
- **Prefer A**: 53
- **Doesn't matter**: 23
- **Prefer B**: 45
- **Strongly Prefer B**: 13

**Overall**

- **57%** for A
- **30%** for B
We received positive feedback from developers when submitting pull requests to discipline undisciplined annotations
110 submitted pull requests. 99 decided!

110 different systems! (one per system!)
Total Acceptance Rate of 71%!

Might Accept + Accept
Thanks, merged

The change is an improvement

Stylistic changes

Deprecated

Bad variable name

Breaks the code

That's much better

Sorry, more serious issues to deal with

I want to make as few changes as possible
Token sequence

```c
#ifdef EXP
    call (p1, p2);
#else
    call (p1);
#endif
```

#define EXP

```
call <> p1 <> p2
```

#undef EXP

```
call <> p1
```

Safe

Fail
Program A

Configurations

C1

C2

Test Cases

#define EXP

Test results

Config | Input | Output
---|---|---
A | !EXP | 10 | 20
A' | !EXP | 10 | 20

Configurations'

C1'

C2'

Program A'

Safe

Fail
We found and fixed a behavioral change in one refactoring of the catalogue

<table>
<thead>
<tr>
<th></th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid programs</td>
<td>7746</td>
<td>7723</td>
<td>14448</td>
<td>6700</td>
</tr>
<tr>
<td>Invalid programs</td>
<td>2254</td>
<td>2277</td>
<td>5452</td>
<td>2300</td>
</tr>
<tr>
<td>Valid refactorings</td>
<td>7746</td>
<td>7723</td>
<td>14448</td>
<td>6700</td>
</tr>
<tr>
<td>Behavioral changes</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

We found and fixed some behavioral changes in the implementation of refactorings
<table>
<thead>
<tr>
<th>Project</th>
<th>Undisciplined</th>
</tr>
</thead>
<tbody>
<tr>
<td>apache</td>
<td>178</td>
</tr>
<tr>
<td>bc</td>
<td>6</td>
</tr>
<tr>
<td>dia</td>
<td>31</td>
</tr>
<tr>
<td>expat</td>
<td>31</td>
</tr>
<tr>
<td>flex</td>
<td>16</td>
</tr>
<tr>
<td>fvwm</td>
<td>61</td>
</tr>
<tr>
<td>ghostscript</td>
<td>87</td>
</tr>
<tr>
<td>gnuchess</td>
<td>2</td>
</tr>
<tr>
<td>gzip</td>
<td>19</td>
</tr>
<tr>
<td>lighttpd</td>
<td>23</td>
</tr>
<tr>
<td>lua</td>
<td>6</td>
</tr>
<tr>
<td>mptris</td>
<td>17</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>477</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clone</th>
<th>LOC</th>
<th>Directives</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0,18%</td>
<td>2,21%</td>
</tr>
<tr>
<td>0</td>
<td>0,12%</td>
<td>0%</td>
</tr>
<tr>
<td>0</td>
<td>0,31%</td>
<td>4,06%</td>
</tr>
<tr>
<td>0</td>
<td>0,44%</td>
<td>3,87%</td>
</tr>
<tr>
<td>0</td>
<td>0,09%</td>
<td>0%</td>
</tr>
<tr>
<td>0</td>
<td>0,11%</td>
<td>3,35%</td>
</tr>
<tr>
<td>0</td>
<td>0,01%</td>
<td>0,95%</td>
</tr>
<tr>
<td>0</td>
<td>0,02%</td>
<td>0%</td>
</tr>
<tr>
<td>0</td>
<td>0,64%</td>
<td>4,03%</td>
</tr>
<tr>
<td>0</td>
<td>0,08%</td>
<td>1,18%</td>
</tr>
<tr>
<td>0</td>
<td>0,12%</td>
<td>3,11%</td>
</tr>
<tr>
<td>0</td>
<td>0,78%</td>
<td>3,05%</td>
</tr>
<tr>
<td>0</td>
<td>0,04%</td>
<td>2,1%</td>
</tr>
</tbody>
</table>
Ongoing work...
"If my memory serves correctly, it was just a refactoring to simplify the code and make it a little more understandable."

"Because using conditional for only part of one high level instruction makes it less understandable and is likely to raise issue over long time maintenance."
The Discipline of Preprocessor-Based Annotations

Does #ifdef TAG n’t #endif Matter

and let TAG be disabled!
A Catalog of Refactorings to Discipline Preprocessor-Based Annotations

Márcio Ribeiro
http://www.ic.ufal.br/marcio
marcio@ic.ufal.br
@marciomribeiro