Automated Software Transplantation

QRS 2016
Talk by Mark Harman
PhD work by Alexandru Marginean

Collaborators
Earl Barr, Yue Jia, Justyna Petke
CREST, University College London
Automated Software Transplantation

QRS 2016
Talk by Mark Harman
PhD work by Alexandru Marginean

Collaborators
Earl Barr, Yue Jia, Justyna Petke
CREST, University College London
Why Autotransplantation?

Why not handle H.264?

Check open source repositories

Start from scratch

~100 players

Video Player
Why Autotransplantation?

C Call Graph?

C Indentation?

Start from scratch

Check open source repositories
BBC World Service Interview
WIRED article
Many shares on Social Media
ACM Distinguished Paper Award at ISSTA '15

Recognition for Our Tool μScalpel

Automated Transplantation of Call Graph and Layout Features into Kate

Alexandru Marginean, Earl T. Barr, Mark Harman, Yue Jia
UCL, Department of Computer Science, CREST Centre

Abstract. We report the automated transplantation of two features currently missing from Kate: call graph generation and automatic layout for C programs, which have been requested by users on the Kate development forum. Our approach uses a lightweight annotation system with Search Based techniques augmented by static analysis for automated transplanting. The results are promising: on average, our tool requires 101 minutes of standard desktop machine time to transplant the call graph feature, and 31 minutes to transplant the layout feature. We repeated each experiment 20 times and validated the resulting transplants using unit, regression and acceptance test suites. In 34 of 40 experiments conducted our search-based autotransplantation tool, μSCALPEL, was able to successfully transplant the new functionality, passing all tests.
Related Work

Clone Detection
Code Migration
Dependence Analysis
Feature Location
Code Salvaging
Feature Extraction
In-Situ Code Reuse
Synchronising Manual Transplants
Automatic Replay Copy-Paste
Automatic Error Fixing
In-Situ Code Reuse
Manual Transplants
Related Work

Miles et al.: In situ reuse of logically extracted functional components
Related Work

Sidiroglou-Douskos et al.: Automatic Error Elimination by Multi-Application Code Transfer
Related Work

Autotransplantation
Human Organ Transplantation
Automated Software Transplantation

Manual Work:
- Organ Entry
- Organ’s Test Suite
- Implantation Point

Donor
- ENTRY
- Organ

Host
- Implantation Point

Organ Test Suite
μTrans

Stage 1: Static Analysis

Stage 2: Genetic Programming

Stage 3: Organ Implantation

Host

Donor

Implantation Point

Organ Entry

Organ Test Suite

Host Beneficiary
Stage 1 — Static Analysis

Donor: int X -> Host: int A, B, C

Donor: int X

Host

Implantation Point

Matching Table

Dependency Graph

Vein

ENTRY

OE

Organ
Stage 2 — GP

Matching Table

Donor Variable ID | Host Variable ID (set)
--- | ---
\(V_1^D\) | \(V_1^H\)  
\(V_2^D\) | \(V_3^H\)  
\(\ldots\) | \(V_{n}^H\)

Genetic Programming

Individual

Var Matching

M1: \(V_1^D\)  
\(\rightarrow\)  
\(V_1^H\)  
M2: \(V_2^D\)  
\(\rightarrow\)  
\(V_4^H\)

Statements

\(S_1\)  
\(S_7\)  
\(S_{73}\)  
\(\ldots\)

Fitness Function:

Week Proxies  
Strong Proxies  
Compilation
Stage 2 - Gp Operators

Matching Table

Donor Variable ID: $V_1^D$  Host Variable ID (set): $V_1^H$ $V_2^H$

Replace Mapping

Var Matching

M1: $V_1^D$ $V_1^H$
M2: $V_2^D$ $V_4^H$

Statements

$S_1$ $S_7$ $S_{73}$ $\ldots$
Stage 2 - Gp Operators

Replace Statement

<table>
<thead>
<tr>
<th>Statements</th>
<th>Var Matching</th>
<th>Individual</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$M_1$: $V_1^D$</td>
<td>$V_1^H$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$M_2$: $V_2^D$</td>
<td>$V_4^H$</td>
</tr>
<tr>
<td>$S_3$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_4$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_5$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_n$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Stage 2 - Gp Operators

\[ S_1, S_2, S_3, S_4, S_5, \ldots, S_n \]

Remove Statement

<table>
<thead>
<tr>
<th>Var Matching</th>
<th>Individual</th>
</tr>
</thead>
<tbody>
<tr>
<td>M_1: ( V_1^D ) -&gt; ( V_1^H )</td>
<td></td>
</tr>
<tr>
<td>M_2: ( V_2^D ) -&gt; ( V_4^H )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_1 ( \times ) S_73 ( \ldots )</td>
</tr>
</tbody>
</table>
Stage 2 - Gp Operators

Add Statement

Individual

Var Matching

M₁: $V_1^D \rightarrow V_1^H$

M₂: $V_2^D \rightarrow V_4^H$

Statements

$S_1$ $S_7$ $S_{73}$ $\ldots$
Stage 2 - Gp Operators

Individual 1
- M₁
- M₂
- S₁
- S₇

Individual 2
- M₃
- M₄
- S₃
- S₉

Crossover Operator

Offspring 1

Offspring 2

Offspring 3

Random Mapping Selection
Host

Donor

Acceptance Tests

Research Questions

RQ1: Do we break the initial functionality?
RQ2: Have we really added new functionality?
RQ3: How about the computational effort?
RQ4: Is autotransplantation useful?
Research Questions

RQ1: Do we break the initial functionality?

RQ2: Have we really added new functionality?

RQ3: How about the computational effort?

RQ4: Is autotransplantation useful?

Empirical Study

15 Transplantations
300 Runs
5 Donors
3 Hosts

Case Studies:

H.264 Encoding
Transplantation;
Kate - call graph generation & C indentation;

RQ4: Is autotransplantation useful?
Validation

Regression Tests

Augmented Regression Tests

RQ1.1

RQ1.2

RQ2

Donor Acceptance Tests

Acceptance Tests

Host Beneficiary

Manual Validation

UCL
Subjects

<table>
<thead>
<tr>
<th>Subjects</th>
<th>Type</th>
<th>Size (KLOC)</th>
<th>Reg. Tests</th>
<th>Organ Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idct</td>
<td>Donor</td>
<td>2.3</td>
<td>-</td>
<td>3-5</td>
</tr>
<tr>
<td>Mytar</td>
<td>Donor</td>
<td>0.4</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>Cflow</td>
<td>Donor</td>
<td>25</td>
<td>-</td>
<td>6-20</td>
</tr>
<tr>
<td>Webserver</td>
<td>Donor</td>
<td>1.7</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>TuxCrypt</td>
<td>Donor</td>
<td>2.7</td>
<td>-</td>
<td>4-5</td>
</tr>
<tr>
<td>Pidgin</td>
<td>Host</td>
<td>363</td>
<td>88</td>
<td>-</td>
</tr>
<tr>
<td>Cflow</td>
<td>Host</td>
<td>25</td>
<td>21</td>
<td>-</td>
</tr>
<tr>
<td>SoX</td>
<td>Host</td>
<td>43</td>
<td>157</td>
<td>-</td>
</tr>
</tbody>
</table>

Case Study

<table>
<thead>
<tr>
<th>Subjects</th>
<th>Type</th>
<th>Size (KLOC)</th>
<th>Reg. Tests</th>
<th>Organ Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLC</td>
<td>Host</td>
<td>422</td>
<td>27</td>
<td>-</td>
</tr>
<tr>
<td>Kate</td>
<td>Host</td>
<td>50</td>
<td>238</td>
<td>-</td>
</tr>
<tr>
<td>x264</td>
<td>Donor</td>
<td>63</td>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td>Cflow</td>
<td>Donor</td>
<td>22</td>
<td>-</td>
<td>13</td>
</tr>
<tr>
<td>Indent</td>
<td>Donor</td>
<td>26</td>
<td>-</td>
<td>7</td>
</tr>
</tbody>
</table>

Minimal size: 0.4k
Max size: 422k
Average Donor: 16k
Average Host: 213k
Experimental Methodology and Setup

- **Host**
  - Implantation Point

- **Donor**
  - OE

- **Organ Test Suite**

Count LOC
CLOC

Coverage Information:
Gcov

*GNU Time*

- x 20

- 64 bit Ubuntu 14.10
  - 16 GB RAM
  - 8 threads

**Validation Test Suites**
## Empirical Study

### RQ1.2

<table>
<thead>
<tr>
<th>Donor</th>
<th>Host</th>
<th>All Passed</th>
<th>Regression</th>
<th>Regression++</th>
<th>Acceptance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idct</td>
<td>Pidgin</td>
<td>16</td>
<td>20</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>Mytar</td>
<td>Pidgin</td>
<td>16</td>
<td>20</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>Web</td>
<td>Pidgin</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>Cflow</td>
<td>Pidgin</td>
<td>15</td>
<td>20</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>Tux</td>
<td>Pidgin</td>
<td>15</td>
<td>20</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>Idct</td>
<td>Cflow</td>
<td>16</td>
<td>17</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Mytar</td>
<td>Cflow</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>20</td>
</tr>
<tr>
<td>Web</td>
<td>Cflow</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>Cflow</td>
<td>Cflow</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Tux</td>
<td>Cflow</td>
<td>14</td>
<td>15</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>Idct</td>
<td>SoX</td>
<td>15</td>
<td>18</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>Mytar</td>
<td>SoX</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>20</td>
</tr>
<tr>
<td>Web</td>
<td>SoX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>Cflow</td>
<td>SoX</td>
<td>14</td>
<td>16</td>
<td>15</td>
<td>14</td>
</tr>
<tr>
<td>Tux</td>
<td>SoX</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>14</td>
</tr>
</tbody>
</table>

**TOTAL**

<table>
<thead>
<tr>
<th>All Passed</th>
<th>Regression</th>
<th>Regression++</th>
<th>Acceptance</th>
</tr>
</thead>
<tbody>
<tr>
<td>188/300</td>
<td>233/300</td>
<td>196/300</td>
<td>256/300</td>
</tr>
</tbody>
</table>

**RQ1.1**

**RQ1.2**

**RQ2**
# Empirical Study

## RQ3

<table>
<thead>
<tr>
<th>Donor</th>
<th>Host</th>
<th>Average</th>
<th>Std. Dev.</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idct</td>
<td>Pidgin</td>
<td>5</td>
<td>7</td>
<td>97</td>
</tr>
<tr>
<td>Mytar</td>
<td>Pidgin</td>
<td>3</td>
<td>1</td>
<td>65</td>
</tr>
<tr>
<td>Web</td>
<td>Pidgin</td>
<td>8</td>
<td>5</td>
<td>160</td>
</tr>
<tr>
<td>Cflow</td>
<td>Pidgin</td>
<td>58</td>
<td>16</td>
<td>1151</td>
</tr>
<tr>
<td>Tux</td>
<td>Pidgin</td>
<td>29</td>
<td>10</td>
<td>574</td>
</tr>
<tr>
<td>Idct</td>
<td>Cflow</td>
<td>3</td>
<td>5</td>
<td>59</td>
</tr>
<tr>
<td>Mytar</td>
<td>Cflow</td>
<td>3</td>
<td>1</td>
<td>53</td>
</tr>
<tr>
<td>Web</td>
<td>Cflow</td>
<td>5</td>
<td>2</td>
<td>102</td>
</tr>
<tr>
<td>Cflow</td>
<td>Cflow</td>
<td>44</td>
<td>9</td>
<td>872</td>
</tr>
<tr>
<td>Tux</td>
<td>Cflow</td>
<td>31</td>
<td>11</td>
<td>623</td>
</tr>
<tr>
<td>Idct</td>
<td>SoX</td>
<td>12</td>
<td>17</td>
<td>233</td>
</tr>
<tr>
<td>Mytar</td>
<td>SoX</td>
<td>3</td>
<td>1</td>
<td>60</td>
</tr>
<tr>
<td>Web</td>
<td>SoX</td>
<td>7</td>
<td>3</td>
<td>132</td>
</tr>
<tr>
<td>Cflow</td>
<td>SoX</td>
<td>89</td>
<td>53</td>
<td>74</td>
</tr>
<tr>
<td>Tux</td>
<td>SoX</td>
<td>34</td>
<td>13</td>
<td>94</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>334 (min)</strong></td>
<td><strong>10 (Average)</strong></td>
<td><strong>72 (hours)</strong></td>
</tr>
</tbody>
</table>
# Case Study

## VLC

### Transplant Time & Test Suites

<table>
<thead>
<tr>
<th>Time (hours)</th>
<th>Regression</th>
<th>Regression++</th>
<th>Acceptance</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.264</td>
<td>26</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
## Case Study - Kate

<table>
<thead>
<tr>
<th>Donor</th>
<th>Host</th>
<th>All Passed</th>
<th>Organ Test Suite</th>
<th>Regression</th>
<th>Regression++</th>
<th>Acceptance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cflow</td>
<td>Kate</td>
<td>16</td>
<td>18</td>
<td>20</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>Indent</td>
<td>Kate</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>34/40</td>
<td>37/40</td>
<td>40/40</td>
<td>35/40</td>
<td>37/40</td>
</tr>
</tbody>
</table>

All Passed — RQ1.1 RQ1.2 RQ2

<table>
<thead>
<tr>
<th>Execution Time (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Donor</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Cflow</td>
</tr>
<tr>
<td>Indent</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>