Estimating the Accuracy of Dynamic Change-impact Analysis using Sensitivity Analysis

Haipeng Cai
Raul Santelices
Tianyu Xu

*University of Notre Dame, USA
* Fudan University, China

Supported by ONR Award N000141410037
Software keeps changing

Program P
- Change 1
- .......

P'
- Change 2
- .......

P''
- Change 3
- .......
Software keeps changing

Program P
- Change 1
- .......

What will be impacted

P'
- Change 2
- .......

Broken

P''
- Change 3
- .......

What will be impacted

Broken
Change impacts need be analyzed

What will be impacted

Program P
- Change 1
- .......

P'
- Change 2
- .......

P''
- Change 3
- .......

After change

Dynamic Change-impact Analysis (CIA)
Change impacts need be analyzed

Program P
• Change 1
• ……

P'
• Change 2
• ……

P’’
• Change 3
• ……

Before change

CIA

Predictive Dynamic Change-impact Analysis (CIA)
How accurate is it?

Predictive Dynamic Change-impact Analysis (CIA)

- Predicted Impacts
  - How Accurate (relative to real impacts)
    - How Safe
    - How Precise

Programbase

Candidate Change Locations

All entities

E1, E2, E3, E4, E5

E2

E1, E2, E3, E5

E2

E4

Real impacts
Why study CIA accuracy?

Predictive Dynamic Change-Impact Analysis (CIA)

Program base

Candidate Change Locations

Predicted Impacts

- Not Accurate
  - Not Safe: threaten quality
  - Not Precise: waste time

(If not accurate) Motivate developing better techniques
Which techniques to study?

- Our target
  - Method level
  - The most cost-effective

Predictive Dynamic Change-impact Analysis (CIA)

- PathImpact (PI)  
  [Law & Rothermel ICSE’03]

- CoverageImpact (CI)  
  [Orso & Apiwattanapong & Harrold FSE’03]

- Execute-After-Sequences (EAS)  
  [Apiwattanapong & Orso & Harrold ICSE’05]

- InfluenceDynamic (InfDyn)  
  [Breech & Tegtmeyer & Pollock ICSM’06]

PI/EAS

Much less precise

Little more precise

Much more expensive
Outline

- Background – PI/EAS
- Methodology
  - Sensitivity Analysis
  - Execution Differencing
- Results
- Contributions
How PI/EAS works

Methods called from M2 directly/transitively

Methods returned into after M2 returns

Methods called after M2 returns

Potentially Impacted

Candidate change location

Potentially Impacted

Call stack

After M2 returned
How PI/EAS works

- **Example code**

```java
public class A {
    static int M1(int f, int z) {
        M2(f+z);
        return new B().M3(f,1);
    }
    void M2(int m) {
        if (m > 0)
            C.M5();
    }
}
```

```java
public class B {
    public static int t=0;
    int M3(int a, int b) {
        int n = b*b - a;
        return n;
    }
    public static boolean M5() {
        return B.t > 10;
    }
}
```

```java
public class C {
    public static void M0() {
        if (A.M1(4,-3) > 0)
            B.M4();
    }
}
```

- **Example execution**

```
enter M0  enter M1  enter M2  enter M5  return into M2  return into M1  enter M3  return into M1  return into M0
```

- **Suppose changes will be in method M2**
  - M5, M3 potentially impacted: entered after M2 entered
  - M1, M0 potentially impacted: returned into after M2 returned
  - Impact set = {M0,M1,M2,M3,M5}
Outline

- Background: PI/EAS
- Methodology
  - Sensitivity Analysis
  - Execution Differencing
- Results
- Contributions
Study as many changes as possible

Predictive Dynamic Change-impact Analysis (CIA)
Bug fix is a common type of change

**Predictive Dynamic Change-impact Analysis (CIA)**

- Program_{buggy}
- Bug fix
- Compute real impacts
- Predicted Impact set
- Real impacts
- Compare

Program_{buggy} and Program_{changed}
Efficient Sensitivity Analysis

Static

Program

‘changed’ version

Program

Instrumenter

Instrumented Program

Runtime

Program₁

Program₂

……

Programₙ

base versions

Change Locations (methods)

Change Randomly

(State) Changer

(State) Changes
Execution Differencing

Base version

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>False</td>
</tr>
<tr>
<td>6</td>
<td>True</td>
</tr>
<tr>
<td>11</td>
<td>-3</td>
</tr>
<tr>
<td>12</td>
<td>-3</td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>False</td>
</tr>
<tr>
<td>4</td>
<td>-3</td>
</tr>
</tbody>
</table>

Execution History

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>False</td>
</tr>
<tr>
<td>6</td>
<td>True</td>
</tr>
<tr>
<td>11</td>
<td>-3</td>
</tr>
<tr>
<td>12</td>
<td>-3</td>
</tr>
<tr>
<td>4</td>
<td>-3</td>
</tr>
</tbody>
</table>

Results

(statements):

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
</tr>
</tbody>
</table>

Changed version

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>False</td>
</tr>
<tr>
<td>6</td>
<td>False</td>
</tr>
<tr>
<td>11</td>
<td>-3</td>
</tr>
<tr>
<td>12</td>
<td>-3</td>
</tr>
<tr>
<td>4</td>
<td>-3</td>
</tr>
</tbody>
</table>

16 public class C {
    public static boolean M5() {
        return B.t > 10;
    }
    public static void M0() {
        if (A.M1(4,-3) > 0)
            B.M4();
    }
}

1 public class A {
    static int M1(int f, int z) {
        M2(f+z);
        return B().M3(f,1);
    }
    void M2(int m) {
        if (m < 0) // m < 0
            C.M5();
    }
    //  m < 0
}

8 public class B {
    public static int t=0;
    int M3(int a, int b) {
        int n = b*b - a;
        return n;
    }
}

15 public class C {
    public static boolean M5() {
        return B.t > 10;
    }
    public static void M0() {
        if (A.M1(4,-3) > 0)
            B.M4();
    }
}
Execution Differencing

```
public class A {
    static int M1(int f, int z) {
        M2(f+z);
        return new B().M3(f,1); }
    void M2(int m) {
        if (m > 0) // m < 0
            C.M5(); }
}
```

```
public class B {
    public static int t=0;
    public static boolean M5() {
        return B.t > 10; }
}
```

```
public class C {
    public static void M0() {
        if (A.M1(4,-3) > 0)
            B.M4(); }
}
```

### Results (statements):
- Line 6: Value True
- Line 7: Value False
- Line 17: Value False

### Results (methods):
- M2
- M5

Method-level Differential Execution Analysis (mDEA)
Accuracy estimation

Base version

Dynamic CIA (PI/EAS)

Test Suite

Changed version

mDEA

Potential change location (e.g., M2)

Predicted Impact Set

Actual change (e.g., \( m>0 \rightarrow m<0 \) at statement 6 of M2)

Compute Accuracy (of PI/EAS)

Real impacts

Tool available: SensA
Outline

- Background: PI/EAS
- Methodology
  - Sensitivity Analysis
  - Execution Differencing
- Results
- Contributions
## Subject programs and statistics

<table>
<thead>
<tr>
<th>Subject</th>
<th>Description</th>
<th>Lines of Code</th>
<th>Methods</th>
<th>Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule1</td>
<td>Priority Scheduler</td>
<td>290</td>
<td>24</td>
<td>2,650</td>
</tr>
<tr>
<td>NanoXML</td>
<td>XML parser</td>
<td>3,521</td>
<td>282</td>
<td>214</td>
</tr>
<tr>
<td>Ant</td>
<td>Java project build tool</td>
<td>18,830</td>
<td>1,863</td>
<td>112</td>
</tr>
<tr>
<td>XML-Security</td>
<td>Encryption library</td>
<td>22,361</td>
<td>1,928</td>
<td>92</td>
</tr>
<tr>
<td>JMeter</td>
<td>Performance monitor</td>
<td>35,547</td>
<td>3,054</td>
<td>79</td>
</tr>
<tr>
<td>ArgoUML</td>
<td>UML Modeling tool</td>
<td>102,400</td>
<td>8,856</td>
<td>211</td>
</tr>
</tbody>
</table>
Metrics

- Impact sets (predicted and real)
- Number of false positives (FP) & false negatives (FN)
- Accuracy (F1)
  - Precision
  - Recall
  - F1 = \(2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}\)
- Result classification
  - All: both S and N
    - Shortening (base execution is over 50% shorter)
    - Normal (otherwise)
average precision 52%
average recall 56%
average recall 56%

average accuracy (F1) 39%
Results for SIR changes (bug fixes)

<table>
<thead>
<tr>
<th>Schedule 1</th>
<th>NanoXML</th>
<th>XML-security</th>
<th>JMeter</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>Recall</td>
<td>Accuracy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100%</td>
<td>90%</td>
<td>80%</td>
<td>70%</td>
<td>60%</td>
</tr>
</tbody>
</table>

Legend:
- Blue: Precision
- Orange: Recall
- Green: Accuracy
New Results for real changes
Future work

- Enhance the experimentation framework to support extended study for
  - More subjects
  - Other dynamic impact analyses
- Develop more precise technique for dynamic impact prediction
Outline

- Background: PI/EAS
- Methodology
  - Sensitivity Analysis
  - Execution Differencing
- Results
- Contributions
Contributions

- A methodology for estimating the accuracy of dynamic impact analyses
- The first empirical study of the predictive accuracy of dynamic impact analysis
- Insights to the effectiveness of predictive dynamic impact analysis
  - Current dynamic impact analysis can be surprisingly imprecise
    - Precision 52% for random changes, 47% for SIR changes
  - Moreover, existing dynamic impact analysis can be also quite unsafe
    - Recall 56% for random changes, 87% for SIR changes