# An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

April 27, 2015

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivation

Componen

AFL Coverage

Mutatio

Implemen

Parsing Containers Intermediate Representation Final Representation

Experir

Case Application Mutant Insertion

`onclucion

#### Table of Contents

- 1. Motivation
- 2. External Components
  - 2.1 Automatic Fault Localization
  - 2.2 Per-Test Coverage
  - 2.3 Mutation
- 3. Implementation
  - 3.1 Parsing Containers
  - 3.2 Intermediate Representation
  - 3.3 Final Representation
  - 3.4 System Output
- 4. Experiment
  - 4.1 Case Application
  - 4.2 Mutant Insertion
  - 4.3 Results Analysis
- 5. Conclusion

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

#### Contents

Motivation

Components

AFL Coverage

Coverage Mutation

Implementatio

Intermediate
Representation
Final Representation

#### Experim

Case Application Mutant Insertion Results Analysis

Conclusio

4 D > 4 A > 4 B > 4 B > B 9 9 0

#### Motivation

▶ Debugging is complex and difficult

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivation

Component

Coverag

Mutation

mplementation

Parsing Containers Intermediate Representation Final Representation

Experim

Case Application
Mutant Insertion

C = = = |...=! = =



#### Motivation

- Debugging is complex and difficult
- ► Fault localization is the most expensive

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Motivation

#### Motivation

- Debugging is complex and difficult
- ► Fault localization is the most expensive
- Current techniques can be improved

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivation

Component

Coverag

Impleme

Parsing Containers
Intermediate

Representation Final Representati System Output

Experim

ase Application

onclusion

Conclusior



#### Automatic Fault Localization

Uses per-test coverage analysis

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motiva

C .....

Component

Coverag

.....

nplementation

Parsing Containers Intermediate Representation Final Representation

Experim

Case Application Mutant Insertion Results Analysis

C = = = |...=! = =

#### Automatic Fault Localization

- Uses per-test coverage analysis
- Ranks statements by suspiciousness

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

\_

Componen

Coverag

mplementation

Parsing Containers Intermediate Representation Final Representation

Experim

Case Application
Mutant Insertion

#### Automatic Fault Localization

- Uses per-test coverage analysis
- Ranks statements by suspiciousness
- Variety of risk evaluation functions Tarantula equation:

$$suspiciousness(e) = 1 - \frac{\frac{failed(e)}{totalfailed}}{\frac{passed(e)}{totalpassed} + \frac{failed(e)}{totalfailed}}$$
(1)

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivation

Components

AFL Coverag

Mutatio

Implemen

Parsing Containers Intermediate Representation Final Representation

Experimen

Case Application
Mutant Insertion

► Eclipse-compatible coverage analysis tool

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Components

ΔFI

Coverage

Implemen

nplementation

Intermediate
Representation
Final Representation
System Output

Experim

Case Application Mutant Insertion Results Analysis

Conclusion

- ► Eclipse-compatible coverage analysis tool
- Uses existing JUnit test suites

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Mativatio

Component

AFL Coverage

Impleme

Parsing Containers
Intermediate

Representation Final Representati System Output

Experim



- ► Eclipse-compatible coverage analysis tool
- Uses existing JUnit test suites
- ▶ Generates coverage information for each test method

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Components

AFL Coverage

Mutation

Implemer

Parsing Containers Intermediate Representation Final Representation

Experim

Case Application
Mutant Insertion
Results Analysis

onclusion



- Eclipse-compatible coverage analysis tool
- Uses existing JUnit test suites
- Generates coverage information for each test method
- Stores output in readable format (XML)

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Coverage



#### CodeCover

```
public static String listsToString(ArrayList<ArrayList<Object>> list)
           String ret = "";
           int listSize = list.size();
           for(int i = 0; i < listSize; i++)
                     i < listSize)
                      ret += "\n";
           return ret:
Problems @ Javadoc 🔂 Declaration 📮 Console 🔑 Test Sessions 🔀
st Session Container: Listswap Oct 27, 2014 5:34:19 PM
Jame
        edu.allegheny.test.ListSwapGeneratorTest:testMixed
        edu.allegheny.test.ListSwapGeneratorTest:testListEmpty
        edu.allegheny.test.ListSwapGeneratorTest:testDouble
```

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivation

Components

AFL Coverage

Mutation

Implementa

Intermediate
Representation
Final Representation
System Output

Experim

Mutant Insertion Results Analysis

Conclusion

#### **MAJOR Mutation**

▶ Introducing faults for experimental evaluation

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Component

Coverage

Mutation

mplementati

Parsing Containers Intermediate Representation

- System Out

Case Application
Mutant Insertion

-----

#### **MAJOR Mutation**

- ▶ Introducing faults for experimental evaluation
- MAJOR mutation system

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivati

Component

AFL Coverage

Mutation

Implemen

Parsing Containers Intermediate

Representation Final Representatio System Output

Experim

Case Application

onclusion

4 D > 4 B > 4 B > 4 B > 9 Q (?)

#### **MAJOR Mutation**

- ► Introducing faults for experimental evaluation
- MAJOR mutation system
- Representative of real-world faults

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Component

AFL Coverag

Mutation

Implemer

Parsing Containers

Intermediate
Representation
Final Representation
System Output

Experim

Case Application

onclusion

CodeCover stores information as XML

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Component

AFL

Coverag Mutatio

mplementation

Parsing Containers Intermediate

Representation Final Representation System Output

Experim

Case Application Mutant Insertion Results Analysis

Conclusion

- CodeCover stores information as XMI
- Parse using Java DOM

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Parsing Containers

- CodeCover stores information as XML
- Parse using Java DOM
- ▶ Contains several pieces of information:

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

. .

Component

AFI

Coverag Mutatio

nplementation

Parsing Containers

Representation
Final Representati

Experim

Case Application
Mutant Insertion

Conclusion

Conclusion



- CodeCover stores information as XML
- Parse using Java DOM
- Contains several pieces of information:
  - ► Complete source code
  - Statement definitions
  - ► List of statements covered by each test method for each file under test

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivation

Component

ΔFI

Coverage

mplementation

Parsing Containers Intermediate Representation Final Representation

System Outpu

Case Application
Mutant Insertion

onclusion

- CodeCover stores information as XML
- Parse using Java DOM
- Contains several pieces of information:
  - ► Complete source code
  - Statement definitions
  - List of statements covered by each test method for each file under test

<BasicStmnt CovItemId="S4" CovItemPrefix="adu.allegheny.listswap.ListSwapGenerator
<LocList>

<Loc EndOffset="777" SrcFileId="1" StartOffset="735"/>
</LocList>

</BasicStmnt>

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivation

Component

AFL

Coverage Mutation

nplementation

Parsing Containers Intermediate Representation Final Representation

Experiment

Case Application Mutant Insertion

### Coverage

```
<TestCase Comment="" Date="1414365770750"</pre>
    Name="edu.allegheny.test.ListSwapGeneratorTest:testString">
<CovList>
<CovPrefix CovItemPrefix="edu.allegheny.listswap.ListSwapGenerator.jaya">
<Cov CovItemId="B1" Value="5"/>
<Cov CovItemId="B2" Value="1"/>
<Cov CovItemId="I.1-2" Value="1"/>
<Cov CovItemId="L2-0" Value="1"/>
<Cov CovItemId="L2-1" Value="1"/>
<Cov CovItemId="L2-2" Value="2"/>
<Cov CovItemId="L3-2" Value="1"/>
<Cov CovItemId="S1" Value="1"/>
<Cov CovItemId="S10" Value="1"/>
<Cov CovItemId="S11" Value="1"/>
<Cov CovItemId="S12" Value="1"/>
<Cov CovItemId="S13" Value="6"/>
<Cov CovItemId="S14" Value="6"/>
<Cov CovItemId="S15" Value="5"/>
<Cov CovItemId="S16" Value="1"/>
<Cov CovItemId="S2" Value="1"/>
<Cov CovItemId="S3" Value="4"/>
```

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Motivation

Parsing Containers

#### Intermediate Representation

▶ DOM is overly complex for multi-pass analysis

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

\_

Component

Coverage

nplementati

Parsing Containers Intermediate Representation

ystem Output

Experim

Case Application Mutant Insertion Results Analysis

onclusion

#### Intermediate Representation

- ▶ DOM is overly complex for multi-pass analysis
- Store information in a more accessible form

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivation

Component

AFL Coverag

Coverag Mutatio

nplementation

Parsing Containers Intermediate Representation Final Representation

ystem Output

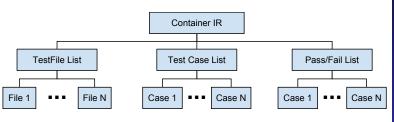
Experim

ase Application Insertion

onclusion

#### Intermediate Representation

- ▶ DOM is overly complex for multi-pass analysis
- Store information in a more accessible form



An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

#### Contents

Motivation

#### Component

AFL Coverage

#### Implementation

Parsing Containers Intermediate Representation Final Representation

#### Experir

Case Application Mutant Insertion Results Analysis

Conclusion

▶ IR is still not conducive to risk evaluation

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Component

AFL Coverag

Mutatio

mplement

Intermediate Representation Final Representation

Experim

Case Application Mutant Insertion Results Analysis

onclusion

- ▶ IR is still not conducive to risk evaluation
- ▶ Reformat information into a simpler representation

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivation

Component

AFL Coverage

Coverage Mutation

Impleme

Intermediate
Representation
Final Representation

Experim

Case Application Mutant Insertion Results Analysis

onclusion

Conclusion



- ▶ IR is still not conducive to risk evaluation
- Reformat information into a simpler representation
- Designed to allow very simple suspiciousness analysis

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivation

Component

AFL

Coverage Mutation

Impleme

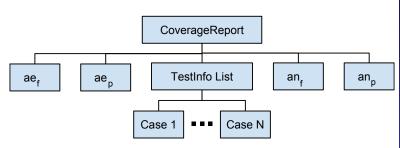
Intermediate
Representation
Final Representation

Experim

Case Application Mutant Insertion Results Analysis

Conclusion

- ▶ IR is still not conducive to risk evaluation
- Reformat information into a simpler representation
- Designed to allow very simple suspiciousness analysis



An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivation

Component

AFL Coverage

Coverage Mutation

Impleme

Parsing Containers Intermediate Representation Final Representation

#### Experin

Case Application Mutant Insertion Results Analysis

onclusion

# System Output

Output data in CSV format

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

System Output

# System Output

- Output data in CSV format
- Conform to Tidy Data standard

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motiva

Componen

Componen

Coverag Mutatio

mplementati

Parsing Containers
Intermediate
Representation

System Output

Experiment

Case Application Mutant Insertion Possilts Applysis

C = = = |...=! = =

Conclusior



# System Output

- Output data in CSV format
- Conform to Tidy Data standard

```
"Function", "StatementID", "Filename", "Suspiciousness", "Rank", "StatementCount", "CaseApplication", "IsFault"
"Jaggard", "S2", "net.sf.jniinghi.JniInchiAtom.jaya", "1", "1", "398", "jniinghi-119", "false"
"Jaccard", "S3", "net.sf.jniinchi.JniInchiAtom.java", "1", "2", "398", "jniinchi-119", "false"
"Jaccard", "$4", "net.sf.jniinchi.JniInchiAtom.java", "1", "3", "398", "jniinchi-119", "false"
"Jaccard", "S5", "net.sf.jniinchi.JniInchiAtom.java", "1", "4", "398", "jniinchi-119", "false"
"Jaccard", "S6", "net.sf.jniinchi.JniInchiAtom.java", "1", "5", "398", "jniinchi-119", "false"
"Jaccard", "S7", "net.sf.jniinchi.JniInchiAtom.jaya", "1", "6", "398", "jniinchi-119", "false"
"Jaccard", "S8", "net.sf.jniinghi.JniInchiAtom.java", "1", "7", "398", "jniinghi-119", "false"
"Jaccard", "S9", "net.sf.iniinchi.JniInchiAtom.java", "1", "8", "398", "jniinchi-119", "false"
"Jaccard", "$10", "net.sf.iniinchi.JniInchiAtom.java", "1", "9", "398", "jniinchi-119", "false"
"Jaccard", "S11", "net.sf.jniinchi.JniInchiAtom.java", "1", "10", "398", "jniinchi-119", "false"
"Jaccard". "S12". "net.sf.jniinghi.JniInchiAtom.jaya", "1", "11", "398", "jniinghi-119", "true"
```

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Motivation

System Output

# Case Application

 Obtained case applications from Sarojini Balasubramanian

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

# Case Application

- Obtained case applications from Sarojini Balasubramanian
- Several applications prepared for use with MAJOR

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

# Case Application

- Obtained case applications from Sarojini Balasubramanian
- Several applications prepared for use with MAJOR
- Only one (Jni-InChi) could be processed by CodeCover

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

#### Mutant Insertion

► Faults must be introduced for experimentation

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Component

Coverag

Implementation

Parsing Containers Intermediate Representation Final Representation

Experim

Case Application Mutant Insertion Results Analysis

C = = = |...=! = .

#### Mutant Insertion

- Faults must be introduced for experimentation
- MAJOR analysis information provided by Sarojini

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Component

AFI

Coverag Mutatio

Implementation

Parsing Containers
Intermediate
Representation
Final Representation

Experim

Case Application Mutant Insertion Results Analysis

- MAJOR analysis information provided by Sarojini
- Killed mutants of various types selected and inserted into Jni-InChi

```
119:ROR:==(java.lang.Object,java.lang.Object):
    FALSE(java.lang.Object,java.lang.Object):
    net.sf.jniinchi.JniInchiAtom@<init>:117:el == null |==> false
137:STD:<CALL>:<NO-OP>:
    net.sf.jniinchi.JniInchiStructure@addBond:99:
    bondList.add(bond) |==> <NO-OP>
149:LVR:POS:NEG:
    net.sf.jniinchi.JniInchiStereoOD@<init>:79:3 |==> -3
194:COR:||(boolean,boolean):LHS(boolean,boolean):
    net.sf.jniinchi.JniInchiWrapper@checkOptions:183:
    op.startsWith("-") || op.startsWith("/") |==> op.startsWith("-")
197:STD:<CALL>:<NO-OP>:
    net.sf.jniinchi.JniInchiWrapper@checkOptions:189:
    sbOptions.append(flagChar + option.name()) |==> <NO-OP>
```

#### An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motiva

Component

AFL Coverage

Implemen<sup>a</sup>

Parsing Containers Intermediate Representation Final Representation

#### Experim

Case Application Mutant Insertion Results Analysis

# Results Analysis

System executed on the CodeCover output for five mutants of Ini-Inchi

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

# Results Analysis

- System executed on the CodeCover output for five mutants of Jni-Inchi
- CSV Output imported into R

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Component

AFL Coverage

Implementation

Parsing Containers Intermediate Representation Final Representation

Experim

Case Application Mutant Insertion Results Analysis

Conclusion

4□ > 4□ > 4□ > 4□ > 4□ > 9

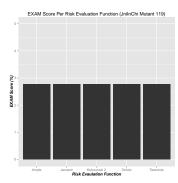
- System executed on the CodeCover output for five mutants of Ini-Inchi
- CSV Output imported into R
- Plots generated for individual mutants and average overall

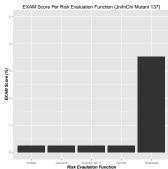
```
library (ggplot2)
attach( jniinchi_119_fault )
graph_119 <- qplot( Function, Exam, data=jniinchi_119_fault, geom="
   bar",
          vlab="EXAM Score (%)", xlab="Risk Evaulation Function",
        stat="identity", vlim=c( 0.5 ).
        main="EXAM Score Per Risk Evaluation Function (JniInChi
           Mutant 119)".
        sub="JniInChi Mutant 119" )
graph_119 <- graph_119 + theme( axis.title=element_text( face="bold.
   italic" ) )
detach( jniinchi_119_fault )
```

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

#### Results

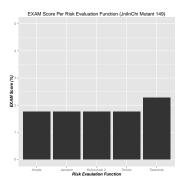


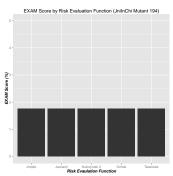


An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

#### Results





An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivat

C----

AFL Coverag

Coverag Mutatio

Implemen

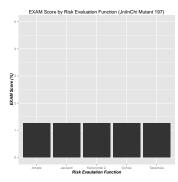
Parsing Containers Intermediate Representation Final Representation

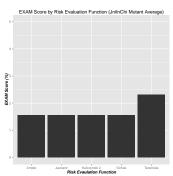
Experir

Case Application Mutant Insertion Results Analysis

^ = = =l...=! = .

#### Results





An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

▶ Of our two original goals, the first proved infeasible

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivat

Component

AFL

Coverage Mutation

mplementatio

Parsing Containers Intermediate Representation Final Representation

Experim

Case Application Mutant Insertion Results Analysis

- ▶ Of our two original goals, the first proved infeasible
- Implemented a system to apply suspiciousness evaluation to coverage data

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivation

Component

AFL Coverag

Coverag Mutatio

Implemen

Parsing Containers Intermediate Representation Final Representation

Experim

Case Application Mutant Insertion Results Analysis



- ▶ Of our two original goals, the first proved infeasible
- Implemented a system to apply suspiciousness evaluation to coverage data
- Completed a small study

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Contents

Motivation

Component

AFL Coverag

Mutatio

Impleme

Parsing Container

Intermediate Representation Final Representation System Output

Experim

Case Application Mutant Insertion Results Analysis

Conclusion



- Of our two original goals, the first proved infeasible
- Implemented a system to apply suspiciousness evaluation to coverage data
- Completed a small study
- ▶ No statistically significant results generated due to CodeCover limitations

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Motivation

- Of our two original goals, the first proved infeasible
- Implemented a system to apply suspiciousness evaluation to coverage data
- Completed a small study
- ▶ No statistically significant results generated due to CodeCover limitations
- Several possible areas for future work identified

An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

Motivation

