Real-world mutation testing:

Effectiveness, efficiency, and proper tool support

René Just

Research Associate

University of Washington, Seattle, USA

July 22, 2014

Research interests: Mutation testing

Correlation between mutants and real faults[1]

- Are mutants a valid substitute for real faults in testing and debugging research?
- Are commonly used mutation operators sufficient?

Efficient mutation analysis[2][3]

- Identify non-redundant, subsumed, and trivial mutants
- Test suite prioritization
- Monitor, propagate, and partition infected execution states

[1] R Just, D Jalali, L Inozemtseva, MD Ernst, R Holmes, G Fraser. *FSE'14*.
[2] R Just, GM Kapfhammer, F Schweiggert. *ISSRE'12*.
[3] R Just, MD Ernst, G Fraser. *ISSTA'14*.

Monitor, propagate, and partition infected execution states

```
public TriangleType classify
        (int a, int b, int c) {
    ...
    if ( a + b <= c ) {
        return Invalid;
    }
    ...
}
```

Original

$\neq$?

```
public TriangleType classify
        (int a, int b, int c) {
    ...
    if ( a * b <= c ) {
        return Invalid;
    }
    ...
}
```

Mutant 1

```
public TriangleType classify
        (int a, int b, int c) {
    ...
    if ( a / b <= c ) {
        return Invalid;
    }
    ...
}
```

Mutant 2

Optimizations:

- Infection

Monitor, propagate, and partition infected execution states

```
public TriangleType classify
      (int a, int b, int c) {
   ...
   if (a + b <= c) {
      return Invalid;
   }
   ...
}
```
Original

```
public TriangleType classify
      (int a, int b, int c) {
   ...
   if (a * b <= c) {
      return Invalid;
   }
   ...
}
```
Mutant 1

```
public TriangleType classify
      (int a, int b, int c) {
   ...
   if (a / b <= c) {
      return Invalid;
   }
   ...
}
```
Mutant 2

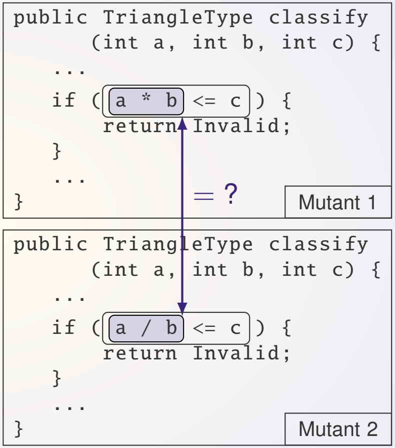
Optimizations:

- Infection
- Propagation

Monitor, propagate, and partition infected execution states

```
public TriangleType classify
        (int a, int b, int c) {
    ...
    if (a + b <= c) {
        return Invalid;
    }
    ...
}
```

Original



```
public TriangleType classify
        (int a, int b, int c) {
    ...
    if (a * b <= c) {
        return Invalid;
    }
    ...
}
```

Mutant 1

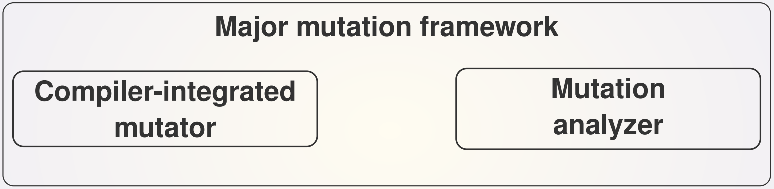
```
public TriangleType classify
        (int a, int b, int c) {
    ...
    if (a / b <= c) {
        return Invalid;
    }
    ...
}
```

Mutant 2

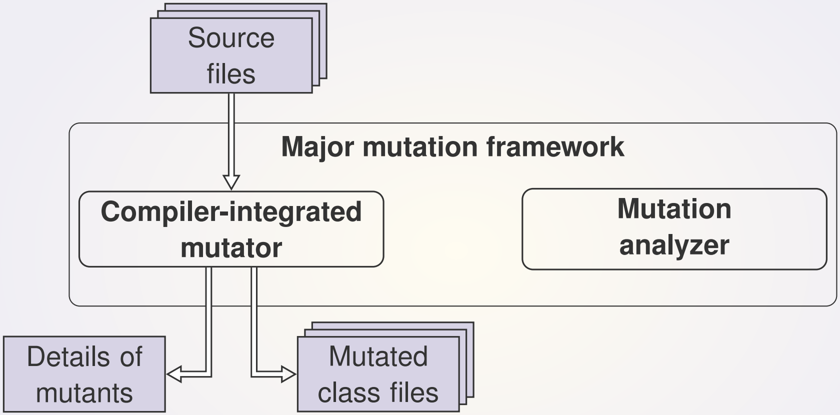
Optimizations:

- Infection
- Propagation
- Partitioning

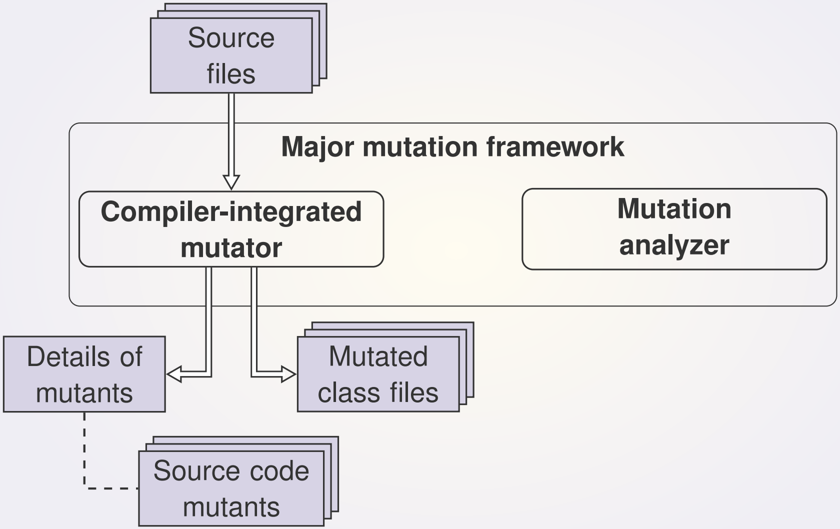
The Major mutation framework



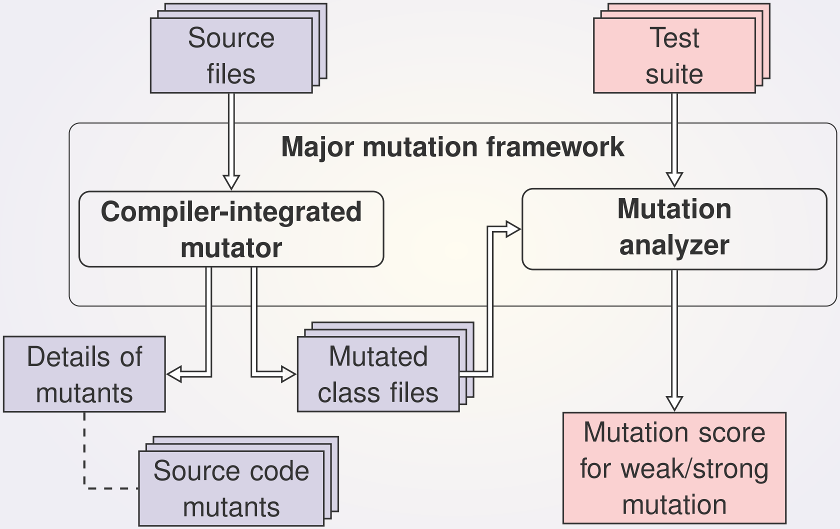
The Major mutation framework



The Major mutation framework



The Major mutation framework



Workshop goals

Acceptance of mutation testing

- What are the open challenges to achieve greater relevance?
- Do we need better tool support for practitioners?

New domains for mutation testing

- What are the next domains for mutation testing?
- How can we define mutation operators for new domains?