

TOWARDS A MDE SOLUTION FOR LANGUAGE INDEPENDENT MUTATION TESTING

P. Gómez-Abajo , E. Guerra, J. de Lara

Modelling&Software Engineering Research Group

<http://miso.es>

Universidad Autónoma de Madrid (Spain)

Mercedes G. Merayo

Design and Testing of Reliable Systems Research Group

<http://antares.sip.ucm.es/testing>

Universidad Complutense de Madrid (Spain)

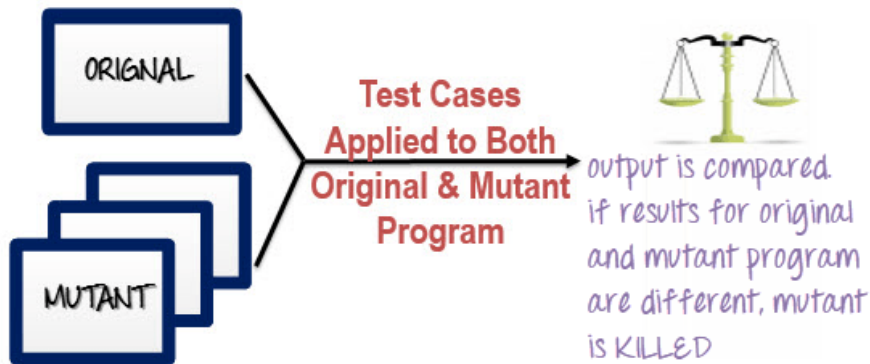


Universidad Autónoma
de Madrid

JISBD - 2018
Sevilla (Spain)



WHAT IS MUTATION TESTING?



- Approach of software testing to assess the quality of test suites
- Injection of syntax changes in a program by using mutation operators
- The mutations introduced simulate common programming faults
- Facilitates to improve the quality of the test-suites and the mutation operators set

PROBLEM

The existing frameworks for mutation testing are:

- Specific for a language (e.g., Java, C)
- Mutation operators are usually manually encoded and not modifiable

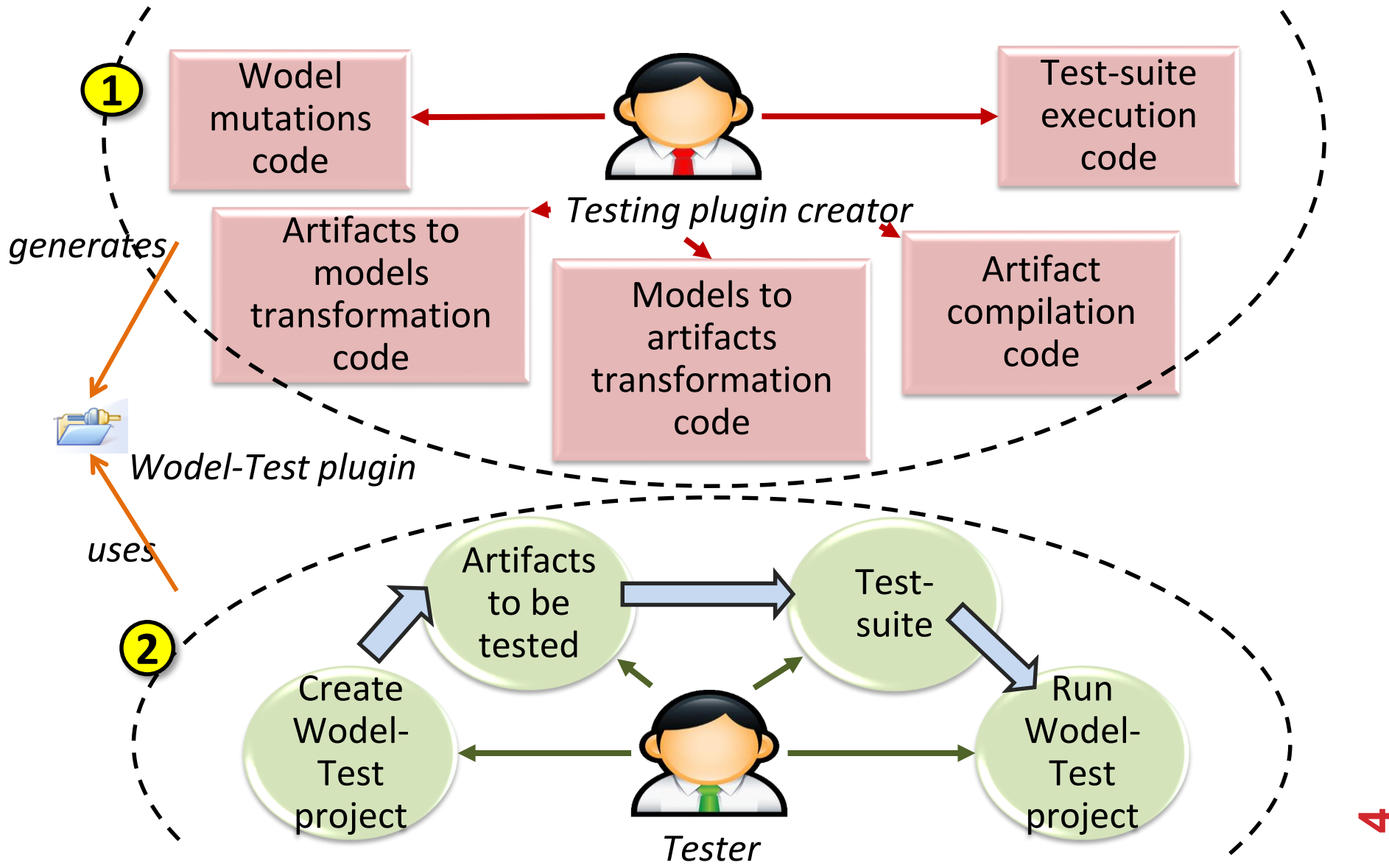
There is a lack of proposals facilitating:

- The definition of mutator operators
- Applicable to arbitrary languages
- Mutation-testing for multi-language projects

PROPOSED SOLUTION

We propose Wodel-Test, an extension to Wodel Framework to facilitate the specification, creation and execution of mutation testing plugins, which is language independent

WODEL-TEST PROCESS



WODEL-TEST: JAVA EXAMPLE

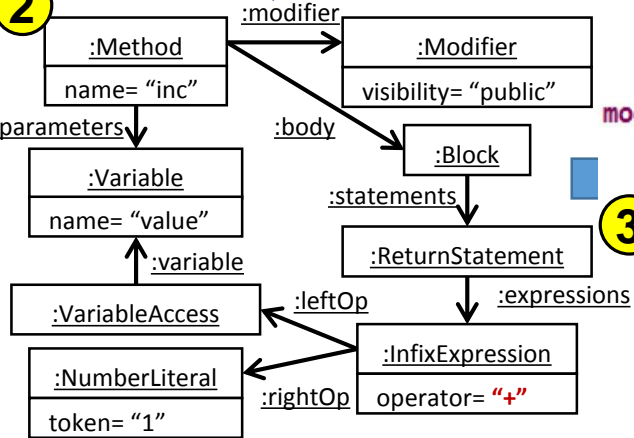
artifact to model

1

```
public int inc(int value) {  
    return value + 1;  
}
```



2



seed model

mutation

modify one InfixExpression
where {operator in ['+', '-']}
with {operator = '*'}



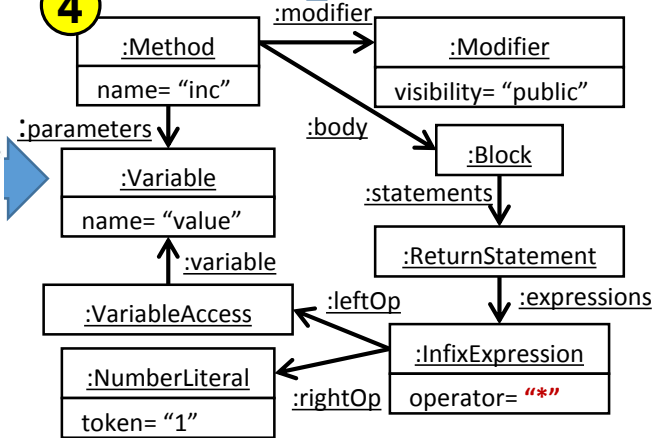
model to artifact

5

```
public int inc(int value) {  
    return value * 1;  
}
```



4



mutant

WODEL-TEST ENVIRONMENT

1 Project Explorer

2 CalculatorEngineTest.java

```
default:
    throw new AssertionError(operator.name())
}

void add(double number) {
    currentTotal += number % 1.0 == 0 ? (int) numl
}

void subtract(double number) {
    currentTotal -= number % 1.0 == 0 ? (int) numl
}

void multiply(double number) {
    currentTotal /= number % 1.0 == 0 ? (int) numl
    //modify information mutator: *= replaced by /=
}

void divide(double number) {
    currentTotal *= number % 1.0 == 0 ? (int) numl
}
```

3 CalculatorEngineTest.java

```
package calculator;
import org.junit.Before;

public class CalculatorEngi...

private CalculatorEngin...
//private CalculatorVie...

@Before
public void runBeforeEv...
    calcView.actionC...
    calcEngine.currentT...
}

@Test
public void testGetTot...
    calcEngine.currentT...
    assertEquals(50, ca...
}
```

4 Wodel-Test global graphical results

Running time (s): 48

Mutation operators applied: not applied (16:28) % Mutation operators applied 36,36% % Mutation operators not applied 63,64%

Mutants killed: equivalent: live (10:6:31) % Mutants killed 21,28% % Muts. equiv. 12,77% % Mutants live 65,96%

Mutants killed: live (10:3:1)

Mutation score 24,39%

Tests failed: passed (5:3) % Tests failed 62,5% % Tests passed 37,5%

5 Wodel-Test mutant results

Equivalent	Package/class/mutant	#Executed tests	#Failed tests	#Passed tests	Applied mutations/Failed test message
	/calculator/asrds/Output0/src/calculator/CalculatorEngine.java	8	1	7	modify information mutator: *= replaced by /=
	testSubtract	1		1	
	testDivide	1		1	
	testMultiplyByZero	1		1	
	testAdd	1		1	
	testMultiply	1	1		expected:<1.0> but was:<100.0>
	testGetTotalStringInt	1		1	
	testEqual	1		1	

6 Wodel-Test test results

Test suite/Test case	#Killed mutants/Message	Applied mutations
/calculator-test/src/calculator/CalculatorEngineTest.java	10	
testSubtract	2	
testDivide	5	
testMultiplyByZero	2	

7 Wodel-Test mutation operator results

Filter	Results	Mutation operator/description	Generated mutants/paths
		asrds/Division assignment	2
		asrds/Modulus assignment	2

CONCLUSIONS

- Wodel-Test is a framework for language independent mutation testing
- Tested with Java and ATL

FUTURE WORK

- Complete the ATL mutations library
- Improve Wodel-Test performance with large models

Thank you!!

Wodel project website:

<http://gomezabajo.github.io/Wodel/>

Pablo.GomezA@uam.es

[@GomezAbajo](#)