Quantitative Evaluation of Mutation Operators for WS-BPEL Compositions

A. Estero F. Palomo I. Medina

SPI&FM Research Group
University of Cádiz

Mutation 2010
April 6th 2010

Contents

Introduction

Research line

Enhance the quality of test-suites automatically for WS-BPEL compositions

Techniques

- Mutation analysis
- Genetic algorithms

Previous works

- A set of mutation operators for the WS-BPEL 2.0 language
- A system for mutant generation based in genetic algorithms: GAMERA

Objective

Measuring the quality of the mutation operators defined to decide whether some of them can be discarded or must be redefined

Motivation

- The economic impact of Web Services and WS-BPEL compositions is increasing. $\Longrightarrow$ It is necessary to pay special attention to the test of this type of software.
- Only a few papers have been published in this field. Most of them are related to test case generation.
- The majority of these papers do not study the quality of test cases generated. $\Longrightarrow$ It is important to advance in the study of quality of test suites for WS compositions in WS-BPEL.

Mutation Analysis

- It allows to measure the quality of test suites.
- It has been successfully applied to different languages.

The WS-BPEL language

What is WS-BPEL?

WS-BPEL is an XML-based language which allows to specify the behavior of a business process based in its interactions with other Web Services.

WS-BPEL process structure

- Declaration of relationships with external partners (client invoking the business process and WS invoked by the process).
- Declaration of variables used by the process.
- Declaration of handlers that the process can use (event handlers, fault handlers, . . .).
- Description of the business process behavior.

The WS-BPEL language

Example

```
<flow>
 <links>
  <link name="checkFlight-To-BookFlight"
 </links>
 <invoke name="checkFlight" ... />
  <sources>
   <source linkName="checkFlight-To-BookFlight"
  </sources>
 </invoke>
 <invoke name="checkHotel"  ... />
 <invoke name="checkRentalCar"  ... />
 <invoke name="bookFlight"  ... />
  <targets>
   <target linkName="checkFlight-To-BookFlight" />
  </targets>
 </invoke>
</flow>
```

The WS-BPEL language

Example

```
<flow>
 <links>
  <link name="checkFlight-To-BookFlight"
 </links>
 <invoke name="checkFlight" ... /> ← Basic activity
  <sources>
   <source linkName="checkFlight-To-BookFlight"
  </sources>
 </invoke>
 <invoke name="checkHotel"  ... />
 <invoke name="checkRentalCar"  ... />
 <invoke name="bookFlight"  ... />
  <targets>
   <target linkName="checkFlight-To-BookFlight" />
  </targets>
 </invoke>
</flow>
```

The WS-BPEL language

Example

```
<flow>
 <links>
  <link name="checkFlight-To-BookFlight"
 </links>
 <invoke name="checkFlight" ... />
  <sources>
   <source linkName="checkFlight-To-BookFlight"
  </sources>
 </invoke>
 <invoke name="checkHotel"  ... />
 <invoke name="checkRentalCar"  ... />
 <invoke name="bookFlight"  ... />
  <targets>
   <target linkName="checkFlight-To-BookFlight" />
  </targets>
 </invoke>
</flow>
```

The WS-BPEL language

Example

```
<flow> ← Structured activity
 <links>
  <link name="checkFlight-To-BookFlight"
 </links>
 <invoke name="checkFlight" ... />
  <sources>
   <source linkName="checkFlight-To-BookFlight"
  </sources>
 </invoke>
 <invoke name="checkHotel"  ... />
 <invoke name="checkRentalCar"  ... />
 <invoke name="bookFlight"  ... />
  <targets>
   <target linkName="checkFlight-To-BookFlight" />
  </targets>
 </invoke>
</flow>
```

The WS-BPEL language

Example

```
<flow>
 <links> ← Container
  <link name="checkFlight-To-BookFlight"
 </links>
 <invoke name="checkFlight" ... />
  <sources> ← Container
   <source linkName="checkFlight-To-BookFlight"
  </sources>
 </invoke>
 <invoke name="checkHotel" ← Attribute ... />
 <invoke name="checkRentalCar" ← Attribute ... />
 <invoke name="bookFlight" ← Attribute ... />
  <targets>  ← Container
   <target linkName="checkFlight-To-BookFlight" />
  </targets>
 </invoke>
</flow>
```

The WS-BPEL language

Example

```
<flow>
 <links>
  <link name="checkFlight-To-BookFlight" ← Attribute/> ← Element
 </links>
 <invoke name="checkFlight" ... />
  <sources>
   <source linkName="checkFlight-To-BookFlight" ← Attribute/> ← Element
  </sources>
 </invoke>
 <invoke name="checkHotel"  ... />
 <invoke name="checkRentalCar"  ... />
 <invoke name="bookFlight"  ... />
  <targets>
   <target linkName="checkFlight-To-BookFlight" ← Attribute/> ← Element
  </targets>
 </invoke>
</flow>
```

The WS-BPEL language

Example

```
<flow>
 <links>
  <link name="checkFlight-To-BookFlight"
 </links>
 <invoke name="checkFlight" ... />
  <sources>
   <source linkName="checkFlight-To-BookFlight"
  </sources>
 </invoke>
 <invoke name="checkHotel"  ... />
 <invoke name="checkRentalCar"  ... />
 <invoke name="bookFlight"  ... />
  <targets>
   <target linkName="checkFlight-To-BookFlight" />
  </targets>
 </invoke>
</flow>
```

The WS-BPEL language

Example

```
<flow>
 <links>
  <link name="checkFlight-To-BookFlight"
 </links>
 <invoke name="checkFlight" ... />
  <sources>
   <source linkName="checkFlight-To-BookFlight"
  </sources>
 </invoke>
 <invoke name="checkHotel"  ... />
 <invoke name="checkRentalCar"  ... />
 <invoke name="bookFlight"  ... />
  <targets>
   <target linkName="checkFlight-To-BookFlight" />
  </targets>
 </invoke>
</flow>
```

The WS-BPEL language

Example

```
<flow>
 <links>
  <link name="checkFlight-To-BookFlight"
 </links>
 <invoke name="checkFlight" ... />
  <sources>
   <source linkName="checkFlight-To-BookFlight"
  </sources>
 </invoke>
 <invoke name="checkHotel"  ... />
 <invoke name="checkRentalCar"  ... />
 <invoke name="bookFlight"  ... />
  <targets>
   <target linkName="checkFlight-To-BookFlight" />
  </targets>
 </invoke>
</flow>
```

The WS-BPEL language

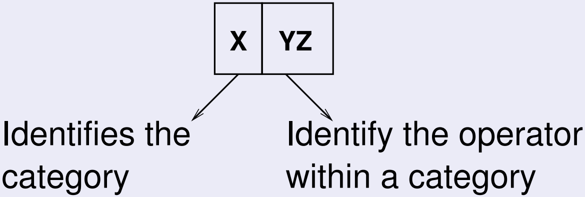
Example

```
<flow>
 <links>
  <link name="checkFlight-To-BookFlight"
 </links>
 <invoke name="checkFlight" ... />
  <sources>
   <source linkName="checkFlight-To-BookFlight"
  </sources>
 </invoke>
 <invoke name="checkHotel"  ... />
 <invoke name="checkRentalCar"  ... />
 <invoke name="bookFlight"  ... />
  <targets>
   <target linkName="checkFlight-To-BookFlight" />
  </targets>
 </invoke>
</flow>
```

Mutation operators for WS-BPEL 2.0

- A set of 26 operators has been defined. These operators have been classified in four categories:
    1. Identifier replacement operators (I)
    2. Expression operators (E)
    3. Activity operators (A)
    4. Exception and event operators (X)

Naming operators



Mutation operators for WS-BPEL 2.0

OPERATOR		DESCRIPTION
		IDENTIFIER MUTATION
ISV		Replaces a variable identifier by another of the same type
		EXPRESSION MUTATION
EAA		Replaces an arithmetic operator by another of the same kind
EEU		Removes the unary minus operator from an expression
ERR		Replaces a relational operator by another of the same kind
ELL		Replaces a logical operator by another of the same kind
ECN		Modifies a numerical constant
ECC	☆	Replaces a path operator by another of the same kind
EMD	☆	Modifies a duration expression
EMF	☆	Modifies a deadline expression

Mutation operators for WS-BPEL 2.0

OPERATOR		DESCRIPTION
		IDENTIFIER MUTATION
ISV		Replaces a variable identifier by another of the same type
		EXPRESSION MUTATION
EAA		Replaces an arithmetic operator by another of the same kind
EEU		Removes the unary minus operator from an expression
ERR		Replaces a relational operator by another of the same kind
ELL		Replaces a logical operator by another of the same kind
ECN		Modifies a numerical constant
ECC	☆	Replaces a path operator by another of the same kind
EMD	☆	Modifies a duration expression
EMF	☆	Modifies a deadline expression

Mutation operators for WS-BPEL 2.0

OPERATOR		DESCRIPTION
		IDENTIFIER MUTATION
ISV		Replaces a variable identifier by another of the same type
		EXPRESSION MUTATION
EAA		Replaces an arithmetic operator by another of the same kind
EEU		Removes the unary minus operator from an expression
ERR		Replaces a relational operator by another of the same kind
ELL		Replaces a logical operator by another of the same kind
ECN		Modifies a numerical constant
ECC	☆	Replaces a path operator by another of the same kind
EMD	☆	Modifies a duration expression
EMF	☆	Modifies a deadline expression

Mutation operators for WS-BPEL 2.0

OPERATOR		DESCRIPTION
		IDENTIFIER MUTATION
ISV		Replaces a variable identifier by another of the same type
		EXPRESSION MUTATION
EAA		Replaces an arithmetic operator by another of the same kind
EEU		Removes the unary minus operator from an expression
ERR		Replaces a relational operator by another of the same kind
ELL		Replaces a logical operator by another of the same kind
ECN		Modifies a numerical constant
ECC	☆	Replaces a path operator by another of the same kind
EMD	☆	Modifies a duration expression
EMF	☆	Modifies a deadline expression

Mutation operators for WS-BPEL 2.0

Operator		Description
		Concurrent Activity Mutation
ACI	☆	Changes the `createInstance` attribute from an inbound message activity to *no*
AFP	☆	Replaces a sequential `forEach` activity by a parallel one
AIS	☆	Changes the `isolated` attribute of a `scope` to *no*
ASF	☆	Replaces a `sequence` activity by a `flow` activity
		Non-concurrent Activity Mutation
AEL		Deletes an activity
AIE		Deletes an `elseif` element or the `else` element from an `if` activity
AWR		Replaces a `while` activity by `repeatUntil` and vice versa
AJC	☆	Removes the `joinCondition` attribute from an activity
ASI	☆	Exchanges the order of two `sequence` child activities
APM	☆	Removes an `onMessage` element from a `pick` activity
APA	☆	Removes the `onAlarm` element from a `pick` activity or from an event handler

Mutation operators for WS-BPEL 2.0

Operator		Description
		Concurrent Activity Mutation
ACI	☆	Changes the `createInstance` attribute from an inbound message activity to *no*
AFP	☆	Replaces a sequential `forEach` activity by a parallel one
AIS	☆	Changes the `isolated` attribute of a `scope` to *no*
ASF	☆	Replaces a `sequence` activity by a `flow` activity
		Non-concurrent Activity Mutation
AEL		Deletes an activity
AIE		Deletes an `elseif` element or the `else` element from an `if` activity
AWR		Replaces a `while` activity by `repeatUntil` and vice versa
AJC	☆	Removes the `joinCondition` attribute from an activity
ASI	☆	Exchanges the order of two `sequence` child activities
APM	☆	Removes an `onMessage` element from a `pick` activity
APA	☆	Removes the `onAlarm` element from a `pick` activity or from an event handler

Mutation operators for WS-BPEL 2.0

OPERATOR		DESCRIPTION
		CONCURRENT ACTIVITY MUTATION
ACI	☆	Changes the `createInstance` attribute from an inbound message activity to *no*
AFP	☆	Replaces a sequential `forEach` activity by a parallel one
AIS	☆	Changes the `isolated` attribute of a `scope` to *no*
ASF	☆	Replaces a `sequence` activity by a `flow` activity
		NON-CONCURRENT ACTIVITY MUTATION
AEL		Deletes an activity
AIE		Deletes an `elseif` element or the `else` element from an `if` activity
AWR		Replaces a `while` activity by `repeatUntil` and vice versa
AJC	☆	Removes the `joinCondition` attribute from an activity
ASI	☆	Exchanges the order of two `sequence` child activities
APM	☆	Removes an `onMessage` element from a `pick` activity
APA	☆	Removes the `onAlarm` element from a `pick` activity or from an event handler

Mutation operators for WS-BPEL 2.0

OPERATOR		DESCRIPTION
		EXCEPTION AND EVENT MUTATION
XMF		Removes a `catch` element or the `catchAll` element from a fault handler
XMC	☆	Removes the definition of a compensation handler
XMT	☆	Removes the definition of a termination handler
XTF	☆	Replaces the fault thrown by a `throw` activity
XER	☆	Removes a `rethrow` activity
XEE	☆	Removes an `onEvent` element from an event handler

Measuring the quality of mutation operators

Questions

- Does an operator generate many invalid mutants?
- Does an operator generate many equivalent mutants?
- How many weak and resistant mutants are produced?
  - A weak mutant is killed by every test case in the test-suite.
  - A resistant mutant is killed by a single test case in the test-suite.
- Is an operator selective at qualifying test cases?
  - Effectiveness

Effectiveness

Derezińska's Effectiveness

$$\mathcal{E} = \frac{\sum_{m \in M} |K_m|}{(|M| - |E|) \cdot |T|}$$

K_m	Test cases killing mutant m
M	Mutants
E	Equivalent mutants
T	Test-suite

- If $\mathcal{E}$ is high for an operator $\Longrightarrow$
  Mutants generated are killed by many test cases.
- If $\mathcal{E}$ is low for an operator $\Longrightarrow$
  Mutants generated are killed by few test cases.

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \frac{\sum_{m \in M} |K_m|}{(|M| - |E|) \cdot |T|}$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \frac{|D|}{|M| - |E|} \cdot \frac{\sum_{m \in M} |K_m|}{|D| \cdot |T|}$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \frac{|D|}{|M| - |E|} \cdot \frac{\sum_{m \in M} |K_m|}{|D| \cdot |T|} = \mathcal{S} \cdot \frac{\sum_{m \in M} |K_m|}{|D| \cdot |T|}$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \mathcal{S} \cdot \frac{\sum_{m \in M} |K_m|}{|D| \cdot |T|}$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \mathcal{S} \cdot \frac{\sum_{m \in M} |K_m|}{|D| \cdot |T|}$$

Simplifying with K_m

$$\sum_{m \in M} |K_m|$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \mathcal{S} \cdot \frac{\sum_{m \in M} |K_m|}{|D| \cdot |T|}$$

Simplifying with K_m

$$\sum_{m \in M} |K_m| = \sum_{m \in D \cup P} |K_m|$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \mathcal{S} \cdot \frac{\sum_{m \in M} |K_m|}{|D| \cdot |T|}$$

Simplifying with K_m

$$\sum_{m \in M} |K_m| = \sum_{m \in D \cup P} |K_m| = \sum_{m \in D} |K_m| + \sum_{m \in P} |K_m| - \sum_{m \in D \cap P} |K_m|$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \mathcal{S} \cdot \frac{\sum_{m \in M} |K_m|}{|D| \cdot |T|}$$

Simplifying with K_m

$$\sum_{m \in M} |K_m| = \sum_{m \in D} |K_m| + \sum_{m \in P} |K_m| - \sum_{m \in D \cap P} |K_m|$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \mathcal{S} \cdot \frac{\sum_{m \in M} |K_m|}{|D| \cdot |T|}$$

Simplifying with K_m

$$\sum_{m \in M} |K_m| = \sum_{m \in D} |K_m| + 0 - \sum_{m \in D \cap P} |K_m|$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \mathcal{S} \cdot \frac{\sum_{m \in M} |K_m|}{|D| \cdot |T|}$$

Simplifying with K_m

$$\sum_{m \in M} |K_m| = \sum_{m \in D} |K_m| - \sum_{m \in D \cap P} |K_m|$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \mathcal{S} \cdot \frac{\sum_{m \in M} |K_m|}{|D| \cdot |T|}$$

Simplifying with K_m

$$\sum_{m \in M} |K_m| = \sum_{m \in D} |K_m| - \sum_{m \in \varnothing} |K_m|$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \mathcal{S} \cdot \frac{\sum_{m \in M} |K_m|}{|D| \cdot |T|}$$

Simplifying with K_m

$$\sum_{m \in M} |K_m| = \sum_{m \in D} |K_m| - 0$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \mathcal{S} \cdot \frac{\sum_{m \in D} |K_m|}{|D| \cdot |T|}$$

Simplifying with K_m

$$\sum_{m \in M} |K_m| = \sum_{m \in D} |K_m|$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \mathcal{S} \cdot \frac{\sum_{m \in D} |K_m|}{|D| \cdot |T|}$$

Simplifying with K_m

$$\sum_{m \in M} |K_m| = \sum_{m \in D} |K_m|$$

Average number of test cases killing *dead* mutants

$$\overline{K}_D = \frac{\sum_{m \in D} |K_m|}{|D|}$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \mathcal{S} \cdot \frac{\sum_{m \in D} |K_m|}{|D| \cdot |T|}$$

Simplifying with K_m

$$\sum_{m \in M} |K_m| = \sum_{m \in D} |K_m|$$

Average number of test cases killing *dead* mutants

$$\overline{K}_D = \frac{\sum_{m \in D} |K_m|}{|D|}$$

Effectiveness and mutation score

A new look at $\mathcal{E}$

$$\mathcal{E} = \mathcal{S} \cdot \frac{\overline{K}_D}{|T|}$$

Simplifying with K_m

$$\sum_{m \in M} |K_m| = \sum_{m \in D} |K_m|$$

Average number of test cases killing *dead* mutants

$$\overline{K}_D = \frac{\sum_{m \in D} |K_m|}{|D|}$$

Efectiveness and test-suite conditions

Properties of our test-suites

- Adequate
- Non-redundant
- Minimal

Remarks

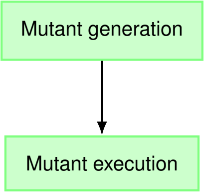
- When a test-suite is adequate, $\mathcal{S} = 1$, therefore: $\mathcal{E} = \frac{\overline{K}_D}{|T|}$.
- The minimum value of $\mathcal{E}$ is $\frac{1}{|T|}$, it is achieved when all the mutants generated are resistant.
- The maximum value of $\mathcal{E}$ is 1, it is achieved when all the mutants generated are weak.
- An operator is more interesting when its $\mathcal{E}$ is near to its minimum.

Experimental method

Mutant generation

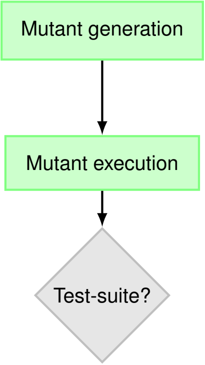
1. Generation of mutants with GAMERA

Experimental method



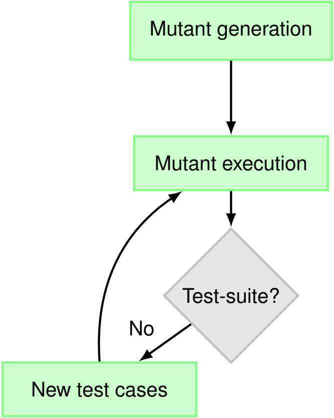
1. Generation of mutants with GAMERA
2. Execution of the mutants against test-suite: invalid and surviving mutants

Experimental method



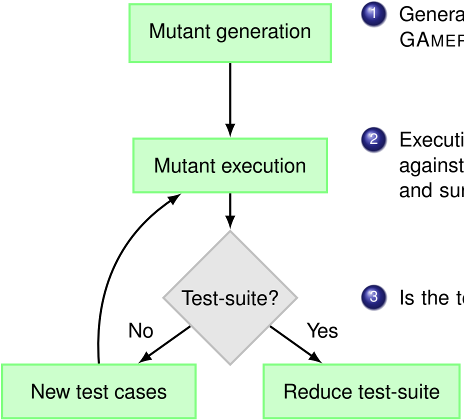
1. Generation of mutants with GAMERA
2. Execution of the mutants against test-suite: invalid and surviving mutants
3. Is the test-suite adequate?

Experimental method



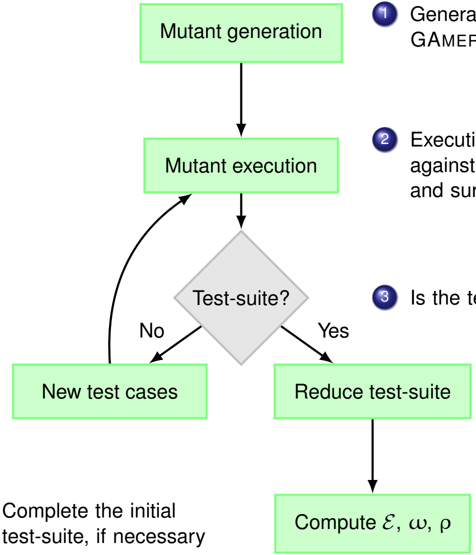
1. Generation of mutants with GAMERA
2. Execution of the mutants against test-suite: invalid and surviving mutants
3. Is the test-suite adequate?
4. Complete the initial test-suite, if necessary

Experimental method



1. Generation of mutants with GAMERA
2. Execution of the mutants against test-suite: invalid and surviving mutants
3. Is the test-suite adequate?
4. Complete the initial test-suite, if necessary
5. Reduce the adequate test-suite to minimum non-redundant test-suite

Experimental method



1. Generation of mutants with GAMERA
2. Execution of the mutants against test-suite: invalid and surviving mutants
3. Is the test-suite adequate?
4. Complete the initial test-suite, if necessary
5. Reduce the adequate test-suite to minimum non-redundant test-suite
6. Compute $\mathcal{E}$ & proportions of weak and resistant mutants

Results

	LoanApproval	MarketPlace	MetaSearch	Total
ISV	0	0	140	140
EAA	0	0	44	44
EEU	0	0	2	2
ERR	10	5	65	80
ELL	0	3	2	5
ECC	13	0	41	54
ECN	4	0	100	104
EMD	0	2	0	2
AFP	0	0	2	2
ASF	5	2	15	22
AEL	17	16	73	106
AIE	2	1	11	14
AJC	0	1	0	1
ASI	9	2	22	33
APM	0	4	0	4
APA	0	1	0	1
XMF	0	2	0	2
XMC	0	2	0	2
Total	60	41	517	618

Results

LoanApproval composition ($|T| = 4$, LOC = 110) $\mathcal{E}_{min} = 0.25$

	Total	Invalid	Equivalent	$\mathcal{E}$	ω (%)	ρ (%)
ERR	10	0	0	0.37	0.0	60.0
ECC	13	0	7	0.37	0.0	50.0
ECN	4	0	0	0.31	0.0	75.0
ASF	5	1	1	0.31	0.0	25.0
AEL	17	1	3	0.45	14.3	35.7
AIE	2	0	0	0.37	0.0	50.0
ASI	9	1	3	0.62	33.3	16.7

Results

LoanApproval composition ($|T| = 4$, LOC = 110) $\mathcal{E}_{min} = 0.25$

	Total	Invalid	Equivalent	$\mathcal{E}$	ω (%)	ρ (%)
ERR	10	0	0	0.37	0.0	60.0
ECC	13	0	7	0.37	0.0	50.0
ECN	4	0	0	0.31	0.0	75.0
ASF	5	1	1	0.31	0.0	25.0
AEL	17	1	3	0.45	14.3	35.7
AIE	2	0	0	0.37	0.0	50.0
ASI	9	1	3	0.62	33.3	16.7

Results

LoanApproval composition ($|T| = 4$, LOC = 110) $\mathcal{E}_{min} = 0.25$

	Total	Invalid	Equivalent	$\mathcal{E}$	ω (%)	ρ (%)
ERR	10	0	0	0.37	0.0	60.0
ECC	13	0	7	0.37	0.0	50.0
ECN	4	0	0	0.31	0.0	75.0
ASF	5	1	1	0.31	0.0	25.0
AEL	17	1	3	0.45	14.3	35.7
AIE	2	0	0	0.37	0.0	50.0
ASI	9	1	3	0.62	33.3	16.7

Results

LoanApproval composition ($|T| = 4$, LOC = 110) $\mathcal{E}_{min} = 0.25$

	Total	Invalid	Equivalent	$\mathcal{E}$	ω (%)	ρ (%)
ERR	10	0	0	0.37	0.0	60.0
ECC	13	0	7	0.37	0.0	50.0
ECN	4	0	0	0.31	0.0	75.0
ASF	5	1	1	0.31	0.0	25.0
AEL	17	1	3	0.45	14.3	35.7
AIE	2	0	0	0.37	0.0	50.0
ASI	9	1	3	0.62	33.3	16.7

Results

MarketPlace composition ($|T| = 4$, LOC = 202) $\mathcal{E}_{min} = 0.25$

	Total	Invalid	Equivalent	$\mathcal{E}$	ω (%)	ρ (%)
ERR	5	0	0	0.45	0.0	40.0
ELL	3	0	0	0.42	0.0	66.7
EMD	2	0	0	0.50	0.0	0.0
ASF	2	0	2	–	–	–
AEL	16	7	0	0.28	0.0	50.0
AIE	1	0	0	0.25	0.0	100.0
AJC	1	0	0	0.25	0.0	100.0
ASI	2	0	0	0.25	0.0	100.0
APM	4	4	0	–	–	–
APA	1	0	0	0.25	0.0	100.0
XMF	2	0	1	0.25	0.0	100.0
XMC	2	0	0	0.25	0.0	100.0

Results

MarketPlace composition ($|T| = 4$, LOC = 202) $\mathcal{E}_{min} = 0.25$

	Total	Invalid	Equivalent	$\mathcal{E}$	ω (%)	ρ (%)
ERR	5	0	0	0.45	0.0	40.0
ELL	3	0	0	0.42	0.0	66.7
EMD	2	0	0	0.50	0.0	0.0
ASF	2	0	2	–	–	–
AEL	16	7	0	0.28	0.0	50.0
AIE	1	0	0	0.25	0.0	100.0
AJC	1	0	0	0.25	0.0	100.0
ASI	2	0	0	0.25	0.0	100.0
APM	4	4	0	–	–	–
APA	1	0	0	0.25	0.0	100.0
XMF	2	0	1	0.25	0.0	100.0
XMC	2	0	0	0.25	0.0	100.0

Results

MarketPlace composition ($|T| = 4$, LOC = 202) $\mathcal{E}_{min} = 0.25$

	Total	Invalid	Equivalent	$\mathcal{E}$	ω (%)	ρ (%)
ERR	5	0	0	0.45	0.0	40.0
ELL	3	0	0	0.42	0.0	66.7
EMD	2	0	0	0.50	0.0	0.0
ASF	2	0	2	–	–	–
AEL	16	7	0	0.28	0.0	50.0
AIE	1	0	0	0.25	0.0	100.0
AJC	1	0	0	0.25	0.0	100.0
ASI	2	0	0	0.25	0.0	100.0
APM	4	4	0	–	–	–
APA	1	0	0	0.25	0.0	100.0
XMF	2	0	1	0.25	0.0	100.0
XMC	2	0	0	0.25	0.0	100.0

Results

MarketPlace composition ($|T| = 4$, LOC = 202) $\mathcal{E}_{min} = 0.25$

	Total	Invalid	Equivalent	$\mathcal{E}$	ω (%)	ρ (%)
ERR	5	0	0	0.45	0.0	40.0
ELL	3	0	0	0.42	0.0	66.7
EMD	2	0	0	0.50	0.0	0.0
ASF	2	0	2	–	–	–
AEL	16	7	0	0.28	0.0	50.0
AIE	1	0	0	0.25	0.0	100.0
AJC	1	0	0	0.25	0.0	100.0
ASI	2	0	0	0.25	0.0	100.0
APM	4	4	0	–	–	–
APA	1	0	0	0.25	0.0	100.0
XMF	2	0	1	0.25	0.0	100.0
XMC	2	0	0	0.25	0.0	100.0

Results

MetaSearch composition ($|T| = 7$, LOC = 633) $\mathcal{E}_{min} = 0.14$

	Total	Invalid	Equivalent	$\mathcal{E}$	ω (%)	ρ (%)
ISV	140	0	8	0.48	0.0	5.3
EAA	44	0	0	0.54	0.0	11.4
EEU	2	0	2	–	–	–
ERR	65	0	20	0.47	0.0	15.6
ELL	2	0	1	0.57	0.0	0.0
ECC	41	0	40	0.71	0.0	0.0
ECN	100	0	35	0.47	0.0	27.7
AFP	2	0	1	0.86	0.0	0.0
ASF	15	1	10	0.61	0.0	25.0
AEL	73	2	6	0.69	16.9	6.1
AIE	11	0	3	0.43	0.0	12.5
ASI	22	1	5	0.66	12.5	12.5

Results

MetaSearch composition ($|T| = 7$, LOC = 633) $\mathcal{E}_{min} = 0.14$

	Total	Invalid	Equivalent	$\mathcal{E}$	ω (%)	ρ (%)
ISV	140	0	8	0.48	0.0	5.3
EAA	44	0	0	0.54	0.0	11.4
EEU	2	0	2	–	–	–
ERR	65	0	20	0.47	0.0	15.6
ELL	2	0	1	0.57	0.0	0.0
ECC	41	0	40	0.71	0.0	0.0
ECN	100	0	35	0.47	0.0	27.7
AFP	2	0	1	0.86	0.0	0.0
ASF	15	1	10	0.61	0.0	25.0
AEL	73	2	6	0.69	16.9	6.1
AIE	11	0	3	0.43	0.0	12.5
ASI	22	1	5	0.66	12.5	12.5

Results

MetaSearch composition ($|T| = 7$, LOC = 633) $\mathcal{E}_{min} = 0.14$

	Total	Invalid	Equivalent	$\mathcal{E}$	ω (%)	ρ (%)
ISV	140	0	8	0.48	0.0	5.3
EAA	44	0	0	0.54	0.0	11.4
EEU	2	0	2	–	–	–
ERR	65	0	20	0.47	0.0	15.6
ELL	2	0	1	0.57	0.0	0.0
ECC	41	0	40	0.71	0.0	0.0
ECN	100	0	35	0.47	0.0	27.7
AFP	2	0	1	0.86	0.0	0.0
ASF	15	1	10	0.61	0.0	25.0
AEL	73	2	6	0.69	16.9	6.1
AIE	11	0	3	0.43	0.0	12.5
ASI	22	1	5	0.66	12.5	12.5

Results

MetaSearch composition ($|T| = 7$, LOC = 633) $\mathcal{E}_{min} = 0.14$

	Total	Invalid	Equivalent	$\mathcal{E}$	ω (%)	ρ (%)
ISV	140	0	8	0.48	0.0	5.3
EAA	44	0	0	0.54	0.0	11.4
EEU	2	0	2	–	–	–
ERR	65	0	20	0.47	0.0	15.6
ELL	2	0	1	0.57	0.0	0.0
ECC	41	0	40	0.71	0.0	0.0
ECN	100	0	35	0.47	0.0	27.7
AFP	2	0	1	0.86	0.0	0.0
ASF	15	1	10	0.61	0.0	25.0
AEL	73	2	6	0.69	16.9	6.1
AIE	11	0	3	0.43	0.0	12.5
ASI	22	1	5	0.66	12.5	12.5

Results

MetaSearch composition ($|T| = 7$, LOC = 633) $\mathcal{E}_{min} = 0.14$

	Total	Invalid	Equivalent	$\mathcal{E}$	ω (%)	ρ (%)
ISV	140	0	8	0.48	0.0	5.3
EAA	44	0	0	0.54	0.0	11.4
EEU	2	0	2	–	–	–
ERR	65	0	20	0.47	0.0	15.6
ELL	2	0	1	0.57	0.0	0.0
ECC	41	0	40	0.71	0.0	0.0
ECN	100	0	35	0.47	0.0	27.7
AFP	2	0	1	0.86	0.0	0.0
ASF	15	1	10	0.61	0.0	25.0
AEL	73	2	6	0.69	16.9	6.1
AIE	11	0	3	0.43	0.0	12.5
ASI	22	1	5	0.66	12.5	12.5

Results

Summary

	% Mutants	% Invalid	% Equiv.	ω (%)	ρ (%)
ISV	22.6	0.0	5.7	0.0	5.3
EAA	7.1	0.0	0.0	0.0	11.4
EEU	0.3	0.0	100.0	–	–
ERR	12.9	0.0	25.0	0.0	25.0
ELL	0.8	0.0	20.0	0.0	50.0
ECC	8.7	0.0	87.0	0.0	42.8
ECN	16.8	0.0	33.6	0.0	30.4
EMD	0.3	0.0	0.0	0.0	0.0
AFP	0.3	0.0	50.0	0.0	0.0
ASF	3.5	9.1	50.0	0.0	25.0
AEL	17.1	9.4	8.5	13.7	17.9
AIE	2.3	0.0	21.4	0.0	27.3
AJC	0.2	0.0	0.0	0.0	100.0
ASI	5.3	6.1	24.2	16.7	15.1
APM	0.6	100.0	0.0	–	–
APA	0.2	0.0	0.0	0.0	100.0
XMF	0.3	0.0	50.0	0.0	100.0
XMC	0.3	0.0	0.0	0.0	100.0

Results

Summary

	% Mutants	% Invalid	% Equiv.	ω (%)	ρ (%)
ISV	22.6	0.0	5.7	0.0	5.3
EAA	7.1	0.0	0.0	0.0	11.4
EEU	0.3	0.0	100.0	–	–
ERR	12.9	0.0	25.0	0.0	25.0
ELL	0.8	0.0	20.0	0.0	50.0
ECC	8.7	0.0	87.0	0.0	42.8
ECN	16.8	0.0	33.6	0.0	30.4
EMD	0.3	0.0	0.0	0.0	0.0
AFP	0.3	0.0	50.0	0.0	0.0
ASF	3.5	9.1	50.0	0.0	25.0
AEL	17.1	9.4	8.5	13.7	17.9
AIE	2.3	0.0	21.4	0.0	27.3
AJC	0.2	0.0	0.0	0.0	100.0
ASI	5.3	6.1	24.2	16.7	15.1
APM	0.6	100.0	0.0	–	–
APA	0.2	0.0	0.0	0.0	100.0
XMF	0.3	0.0	50.0	0.0	100.0
XMC	0.3	0.0	0.0	0.0	100.0

Results

Summary

	% Mutants	% Invalid	% Equiv.	ω (%)	ρ (%)
ISV	22.6	0.0	5.7	0.0	5.3
EAA	7.1	0.0	0.0	0.0	11.4
EEU	0.3	0.0	100.0	–	–
ERR	12.9	0.0	25.0	0.0	25.0
ELL	0.8	0.0	20.0	0.0	50.0
ECC	8.7	0.0	87.0	0.0	42.8
ECN	16.8	0.0	33.6	0.0	30.4
EMD	0.3	0.0	0.0	0.0	0.0
AFP	0.3	0.0	50.0	0.0	0.0
ASF	3.5	9.1	50.0	0.0	25.0
AEL	17.1	9.4	8.5	13.7	17.9
AIE	2.3	0.0	21.4	0.0	27.3
AJC	0.2	0.0	0.0	0.0	100.0
ASI	5.3	6.1	24.2	16.7	15.1
APM	0.6	100.0	0.0	–	–
APA	0.2	0.0	0.0	0.0	100.0
XMF	0.3	0.0	50.0	0.0	100.0
XMC	0.3	0.0	0.0	0.0	100.0

Results

Summary

	% Mutants	% Invalid	% Equiv.	ω (%)	ρ (%)
ISV	22.6	0.0	5.7	0.0	5.3
EAA	7.1	0.0	0.0	0.0	11.4
EEU	0.3	0.0	100.0	–	–
ERR	12.9	0.0	25.0	0.0	25.0
ELL	0.8	0.0	20.0	0.0	50.0
ECC	8.7	0.0	87.0	0.0	42.8
ECN	16.8	0.0	33.6	0.0	30.4
EMD	0.3	0.0	0.0	0.0	0.0
AFP	0.3	0.0	50.0	0.0	0.0
ASF	3.5	9.1	50.0	0.0	25.0
AEL	17.1	9.4	8.5	13.7	17.9
AIE	2.3	0.0	21.4	0.0	27.3
AJC	0.2	0.0	0.0	0.0	100.0
ASI	5.3	6.1	24.2	16.7	15.1
APM	0.6	100.0	0.0	–	–
APA	0.2	0.0	0.0	0.0	100.0
XMF	0.3	0.0	50.0	0.0	100.0
XMC	0.3	0.0	0.0	0.0	100.0

Results

Summary

	% Mutants	% Invalid	% Equiv.	ω (%)	ρ (%)
ISV	22.6	0.0	5.7	0.0	5.3
EAA	7.1	0.0	0.0	0.0	11.4
EEU	0.3	0.0	100.0	–	–
ERR	12.9	0.0	25.0	0.0	25.0
ELL	0.8	0.0	20.0	0.0	50.0
ECC	8.7	0.0	87.0	0.0	42.8
ECN	16.8	0.0	33.6	0.0	30.4
EMD	0.3	0.0	0.0	0.0	0.0
AFP	0.3	0.0	50.0	0.0	0.0
ASF	3.5	9.1	50.0	0.0	25.0
AEL	17.1	9.4	8.5	13.7	17.9
AIE	2.3	0.0	21.4	0.0	27.3
AJC	0.2	0.0	0.0	0.0	100.0
ASI	5.3	6.1	24.2	16.7	15.1
APM	0.6	100.0	0.0	–	–
APA	0.2	0.0	0.0	0.0	100.0
XMF	0.3	0.0	50.0	0.0	100.0
XMC	0.3	0.0	0.0	0.0	100.0

Results

Summary

	% Mutants	% Invalid	% Equiv.	ω (%)	ρ (%)
ISV	22.6	0.0	5.7	0.0	5.3
EAA	7.1	0.0	0.0	0.0	11.4
EEU	0.3	0.0	100.0	–	–
ERR	12.9	0.0	25.0	0.0	25.0
ELL	0.8	0.0	20.0	0.0	50.0
ECC	8.7	0.0	87.0	0.0	42.8
ECN	16.8	0.0	33.6	0.0	30.4
EMD	0.3	0.0	0.0	0.0	0.0
AFP	0.3	0.0	50.0	0.0	0.0
ASF	3.5	9.1	50.0	0.0	25.0
AEL	17.1	9.4	8.5	13.7	17.9
AIE	2.3	0.0	21.4	0.0	27.3
AJC	0.2	0.0	0.0	0.0	100.0
ASI	5.3	6.1	24.2	16.7	15.1
APM	0.6	100.0	0.0	–	–
APA	0.2	0.0	0.0	0.0	100.0
XMF	0.3	0.0	50.0	0.0	100.0
XMC	0.3	0.0	0.0	0.0	100.0

Results

Summary

	% Mutants	% Invalid	% Equiv.	ω (%)	ρ (%)
ISV	22.6	0.0	5.7	0.0	5.3
EAA	7.1	0.0	0.0	0.0	11.4
EEU	0.3	0.0	100.0	–	–
ERR	12.9	0.0	25.0	0.0	25.0
ELL	0.8	0.0	20.0	0.0	50.0
ECC	8.7	0.0	87.0	0.0	42.8
ECN	16.8	0.0	33.6	0.0	30.4
EMD	0.3	0.0	0.0	0.0	0.0
AFP	0.3	0.0	50.0	0.0	0.0
ASF	3.5	9.1	50.0	0.0	25.0
AEL	17.1	9.4	8.5	13.7	17.9
AIE	2.3	0.0	21.4	0.0	27.3
AJC	0.2	0.0	0.0	0.0	100.0
ASI	5.3	6.1	24.2	16.7	15.1
APM	0.6	100.0	0.0	–	–
APA	0.2	0.0	0.0	0.0	100.0
XMF	0.3	0.0	50.0	0.0	100.0
XMC	0.3	0.0	0.0	0.0	100.0

Results

Summary

	% Mutants	% Invalid	% Equiv.	ω (%)	ρ (%)
ISV	22.6	0.0	5.7	0.0	5.3
EAA	7.1	0.0	0.0	0.0	11.4
EEU	0.3	0.0	100.0	–	–
ERR	12.9	0.0	25.0	0.0	25.0
ELL	0.8	0.0	20.0	0.0	50.0
ECC	8.7	0.0	87.0	0.0	42.8
ECN	16.8	0.0	33.6	0.0	30.4
EMD	0.3	0.0	0.0	0.0	0.0
AFP	0.3	0.0	50.0	0.0	0.0
ASF	3.5	9.1	50.0	0.0	25.0
AEL	17.1	9.4	8.5	13.7	17.9
AIE	2.3	0.0	21.4	0.0	27.3
AJC	0.2	0.0	0.0	0.0	100.0
ASI	5.3	6.1	24.2	16.7	15.1
APM	0.6	100.0	0.0	–	–
APA	0.2	0.0	0.0	0.0	100.0
XMF	0.3	0.0	50.0	0.0	100.0
XMC	0.3	0.0	0.0	0.0	100.0

Conclusions

1. This work evaluates the quality of a set of mutation operators
   - The target language is WS-BPEL 2.0
   - A set of 26 mutation operators has been used
2. We have conducted several experiments to determine:
   - The number of invalid and equivalent mutants
   - The effectiveness of the mutation operators
   - The proportions of weak and resistant mutants
3. Test-suites used accomplish several conditions
   - Adequate
   - Non-redundant
   - Minimum size
4. Therefore, results are not distorted by unproductive test cases.

Limitations and future work

Limitations

1. EMF, ACI, AIS, AWR, XMT, XTF, XER, and XEE are not used.
2. We use mockups instead of real services.
   - Mockups are a poor substitute for real services.
   - Their simpler logic can increase the number of equivalent mutants.

Future work

1. Extending the study to a wider number of compositions
   - Exercise all the mutation operators defined.
   - Validate the results with bigger compositions.
2. Making mockups smarter
3. Implementing approximation and exact algorithms for reduction
   - Currently, we use a greedy approximation algorithm.
   - Small size instances can be reduced to hand-manageable sizes.
   - Assuring minimum size for big instances needs exact algorithms.

Thanks for your attention!

SPI&FM Research Group
University of Cádiz
SOAQSim Project: TIN2007-67843-C06-04

antonia.estero@uca.es