SEED MODEL SYNTHESIS FOR TESTING MODEL-BASED MUTATION OPERATORS

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)

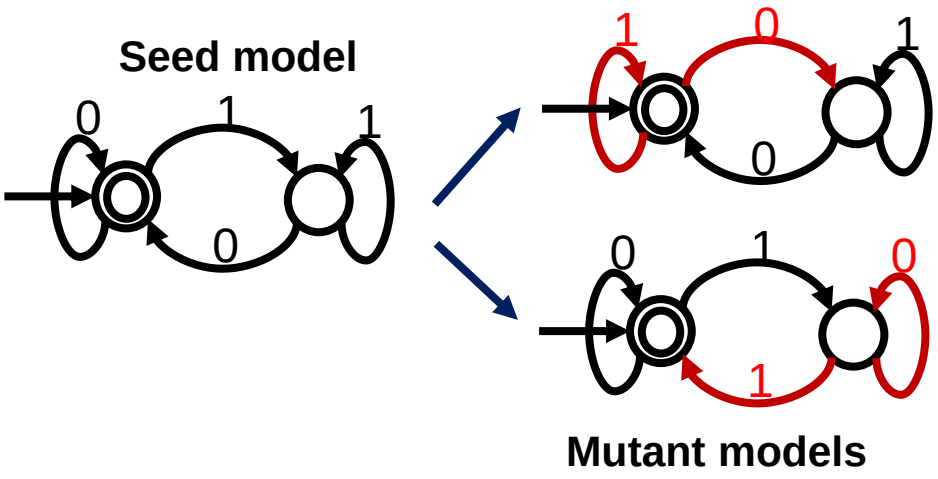
CAiSE FORUM 2020
June 8-12, Grenoble

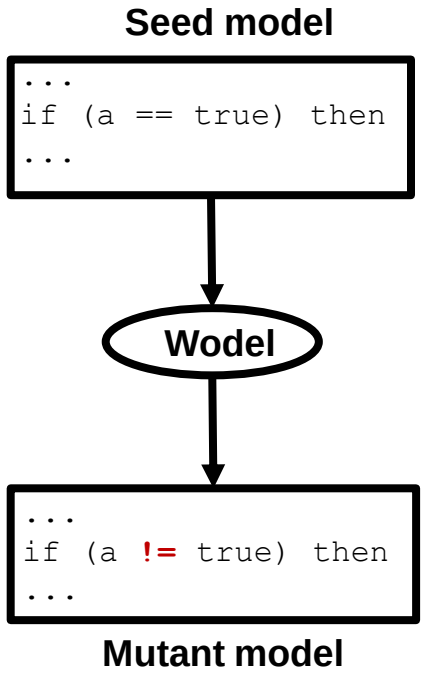
WHAT IS MODEL MUTATION?

A model mutation is a variation of a seed model by the application of one or more mutation operators

Model mutation has many applications:

- Model transformation testing
- Model-based software mutation testing
- Software product lines testing
- Automated generation of exercises
- Search-based engineering
- ...





PROBLEM

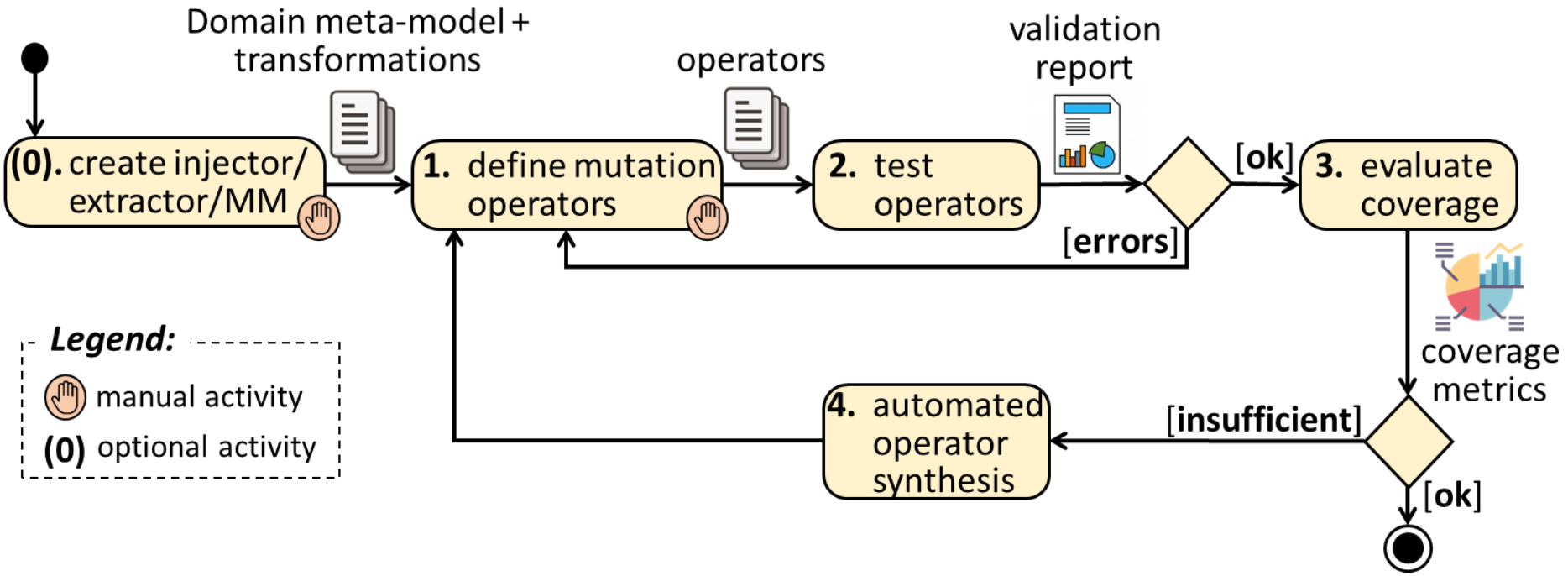
There is a need that mutation operators:

- Can be defined using a simple, compact notation
- Change artefacts in pertinent ways
- Easy to test

PROPOSED SOLUTION

- DSL Wodel for model mutation
- Model synthesis from the Wodel program
- Mutation operators can be tested over these models

PROCESS FOR THE ENGINEERING OF MUTATION OPERATORS



WODEL

DSL Wodel for model mutation with:

- High level mutation primitives
- Domain independent
- Compiled to Java code

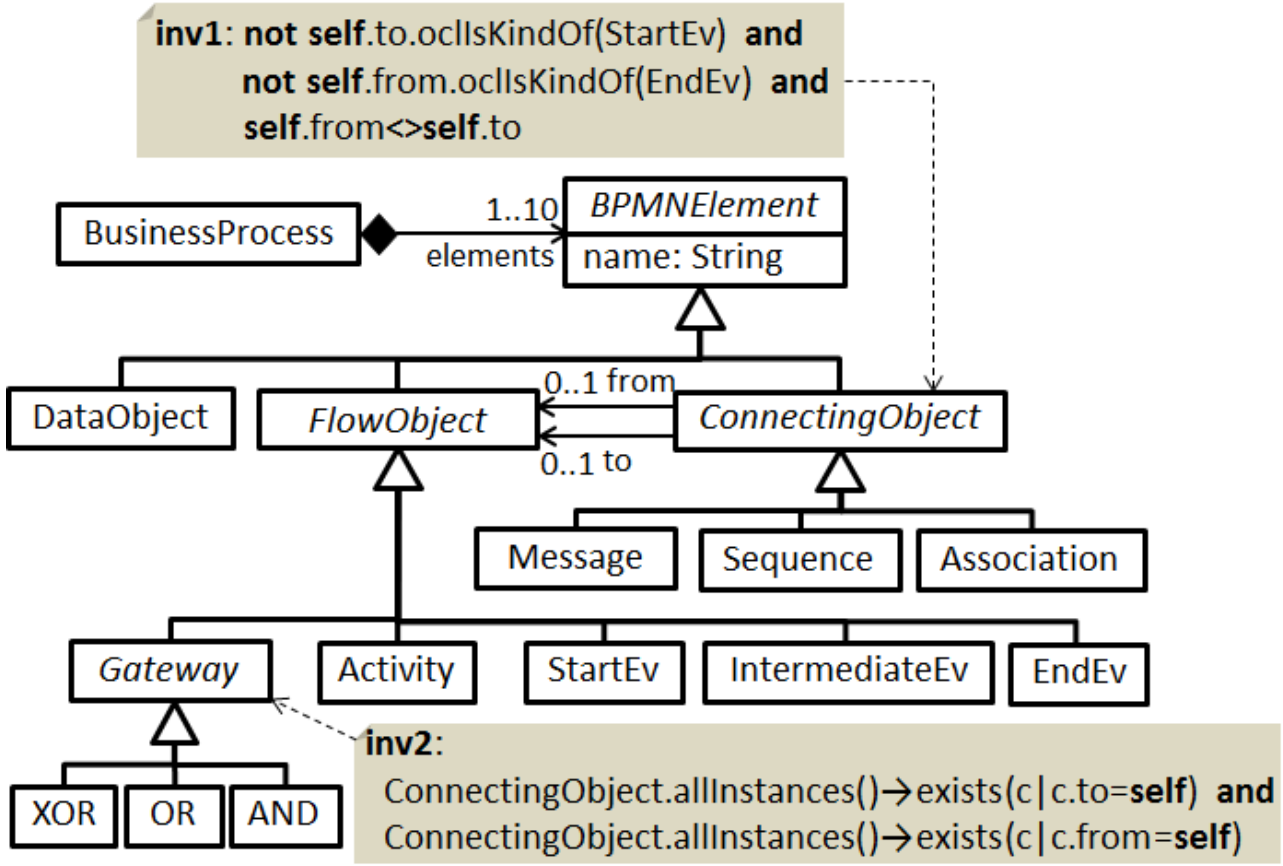
Extensible execution services

- Mutant validation
- Equivalent mutants detection
- Registry of the applied mutation operators
- Extensible for post-processing applications

Development services

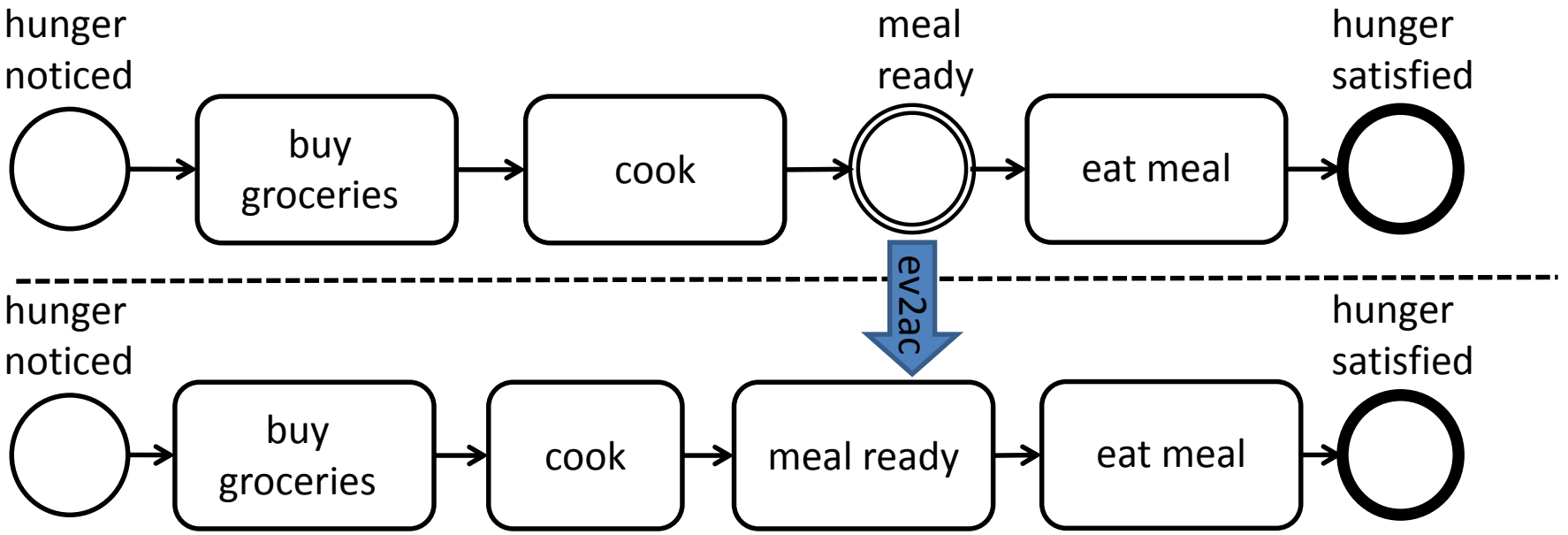
- Seed model synthesis
- Mutation footprints

WODEL EXAMPLE



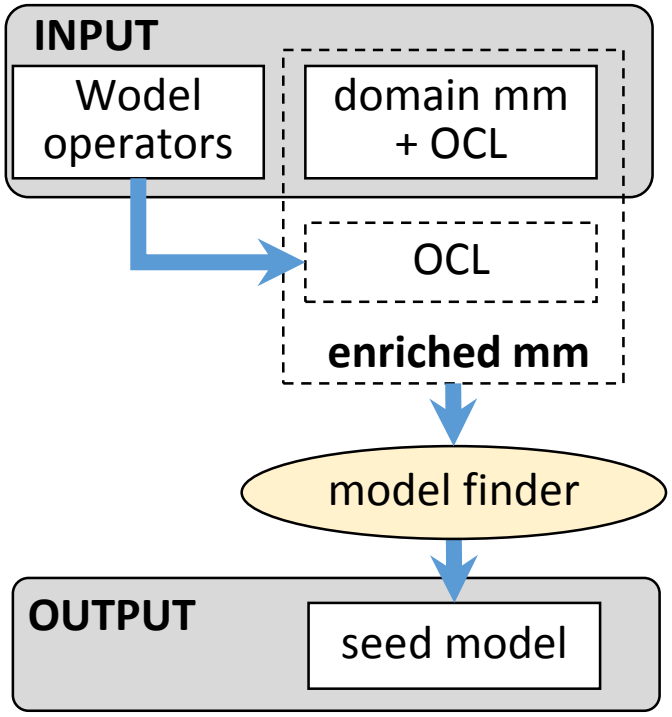

```
generate exhaustive mutants in "out/" from "models/"
metamodel "http://bpmn.com"

with blocks {
  ev2ac "Retypes an Event as an Activity"
  {
     retype one [StartEv, IntermediateEv, EndEv] as Activity
  }}
```

MUTATION EXAMPLE



MODEL SYNTHESIS

- Testing of mutation operators via model synthesis
- Models ensure the execution of all operators statements
- Based in model finding



OCL TEMPLATES

Conditions to check	OCL Template	Example

Object creation:
There is a container reference of the object's type with space to add more objects
	`<Container>.allInstances()`
	`->exists(o |`
	`o.<ref>->size() < <upB>)`
		Wodel: `a = create Activity`
		
		OCL:
		`BusinessProcess.allInstances()`
		`->exists(b | b.elements->size() < 10)`

Object deletion:
There is an object with the given type and feature values, and its deletion does not violate the lower bound of any reference of the object's type
	`<Class>.allInstances()`
	`->exists(o | <object-filter> and`
	`<Container>.allInstances()`
	`->forAll(c |`
	`c.<ref>->includes(o) implies`
	`c.<ref>->size() > <lowB>))`
		Wodel: `remove one Activity`
		
		OCL:
		`Activity.allInstances()`
		`->exists(a |`
		`BusinessProcess.allInstances()`
		`->forAll(b | b.elements->includes(a)`
		`implies b.elements->size() > 1))`

...

EXAMPLE

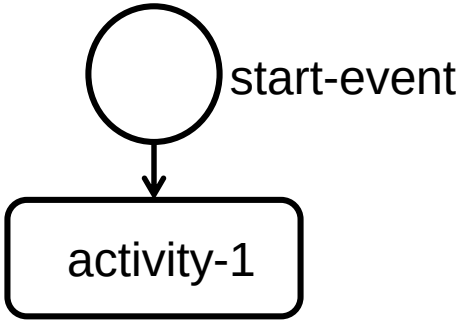
```
generate 2 mutants in "out/" from "models/"
metamodel http://bpmn.com

with blocks {
  ev2ac "Retypes an Event as an Activity"
  {
    retype one [StartEv, IntermediateEv, EndEv]
        as Activity
  }}
```

OCL

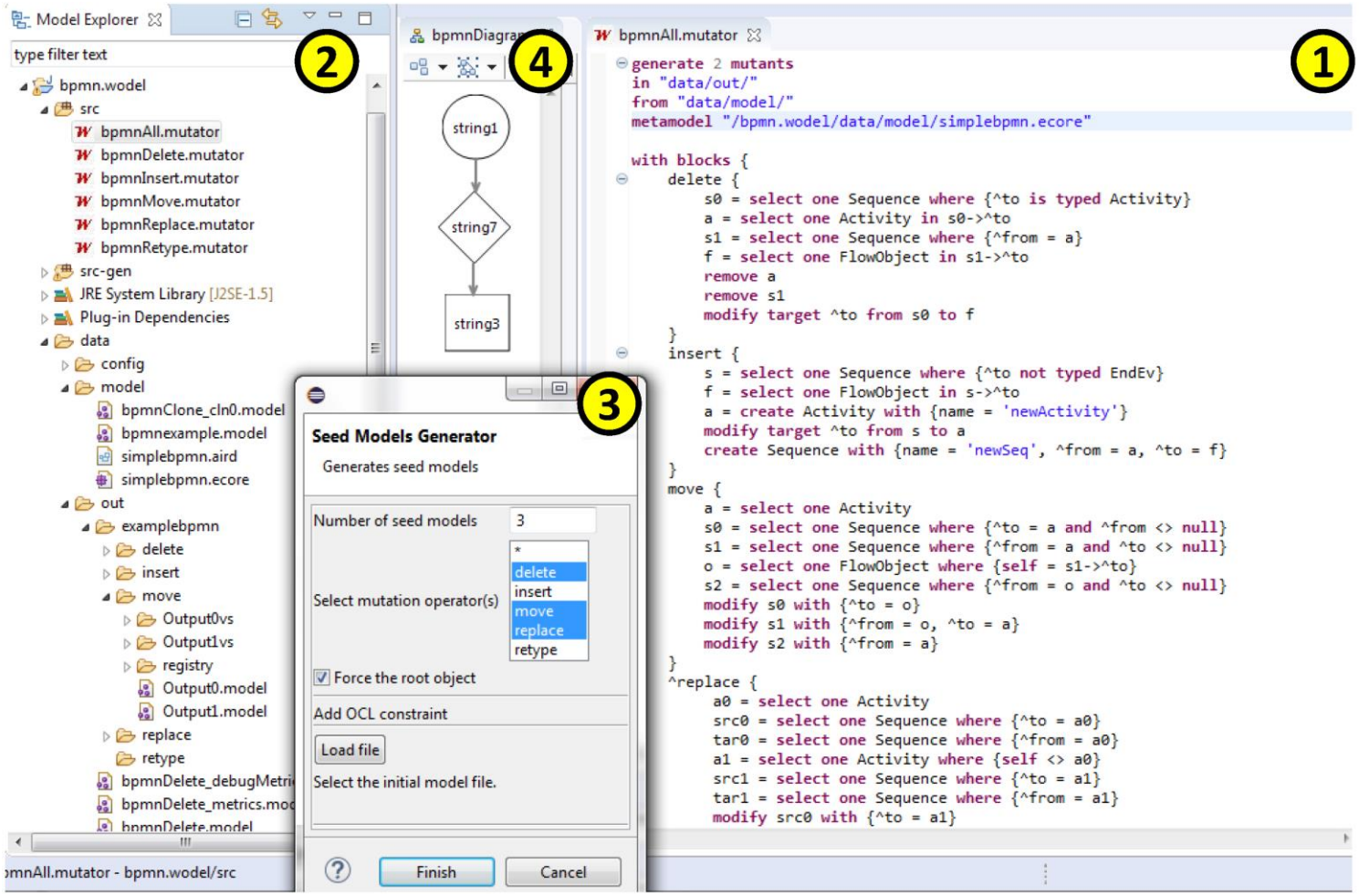
```
context Dummy
inv mut1 :  StartEv.allInstances()->exists(a | true) or
            IntermediateEv.allInstances()->exists(a | true) or
            EndEv.allInstances()->exists(a | true)
```

Model finder

Generated model

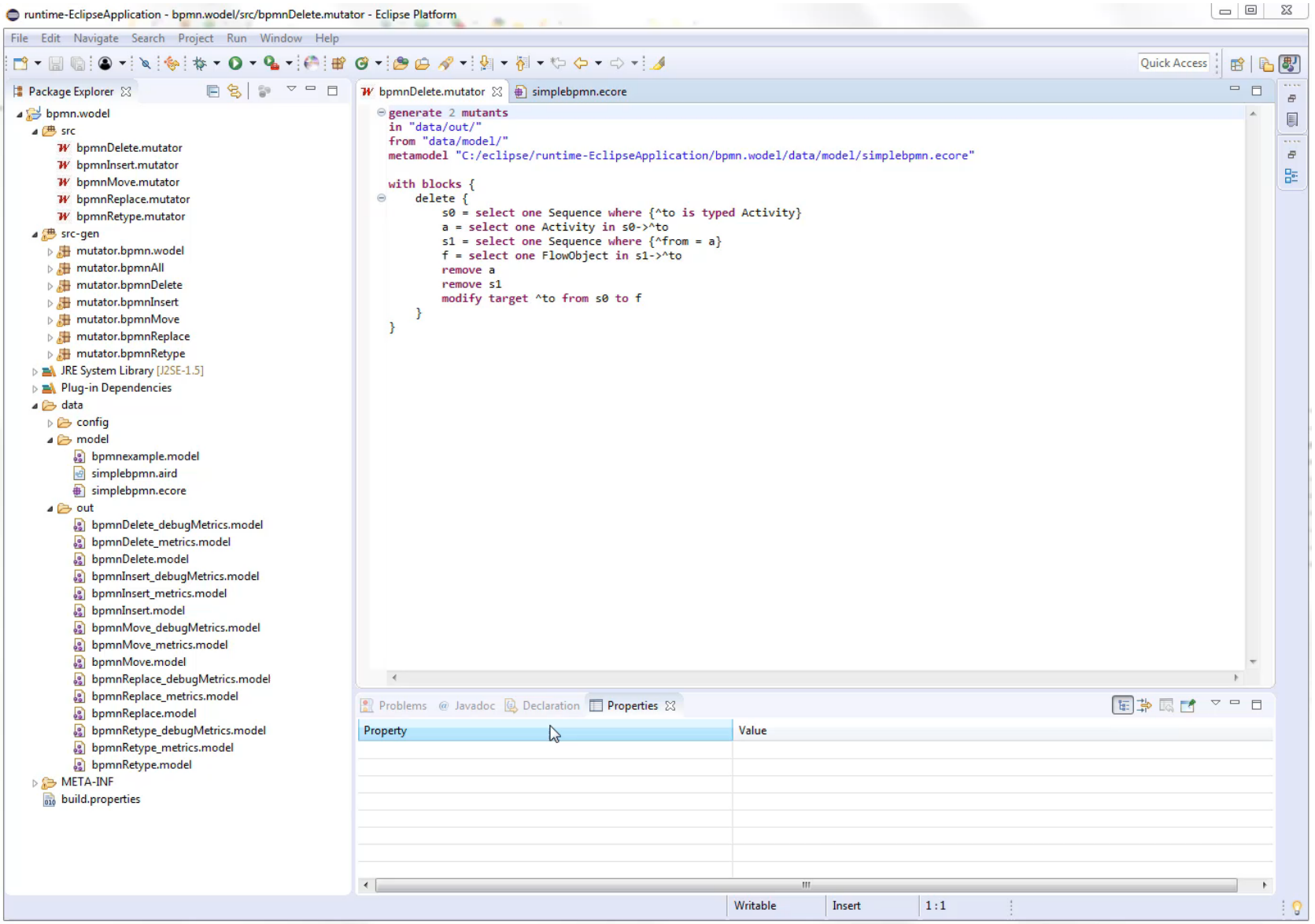


TOOL SUPPORT



Tool available at http://gomezabajo.github.io/Wodel/ along with videos, tutorials & examples

DEMO


runtime-EclipseApplication - bpmn.wodel/src/bpmnDelete.mutator - Eclipse Platform
File Edit Navigate Search Project Run Window Help
Quick Access
Package Explorer
bpmn.wodel
src
bpmnDelete.mutator
bpmnInsert.mutator
bpmnMove.mutator
bpmnReplace.mutator
bpmnRetype.mutator
src-gen
mutator.bpmn.wodel
mutator.bpmnAll
mutator.bpmnDelete
mutator.bpmnInsert
mutator.bpmnMove
mutator.bpmnReplace
mutator.bpmnRetype
JRE System Library [J2SE-1.5]
Plug-in Dependencies
data
config
model
bpmnexample.model
simplebpmn.aird
simplebpmn.ecore
out
bpmnDelete_debugMetrics.model
bpmnDelete_metrics.model
bpmnDelete.model
bpmnInsert_debugMetrics.model
bpmnInsert_metrics.model
bpmnInsert.model
bpmnMove_debugMetrics.model
bpmnMove_metrics.model
bpmnMove.model
bpmnReplace_debugMetrics.model
bpmnReplace_metrics.model
bpmnReplace.model
bpmnRetype_debugMetrics.model
bpmnRetype_metrics.model
bpmnRetype.model
META-INF
build.properties
bpmnDelete.mutator
simplebpmn.ecore
generate 2 mutants
in "data/out/"
from "data/model/"
metamodel "C:/eclipse/runtime-EclipseApplication/bpmn.wodel/data/model/simplebpmn.ecore"
with blocks {
delete {
s0 = select one Sequence where {^to is typed Activity}
a = select one Activity in s0->^to
s1 = select one Sequence where {^from = a}
f = select one FlowObject in s1->^to
remove a
remove s1
modify target ^to from s0 to f
}
}
Problems
Javadoc
Declaration
Properties
Property
Value
Writable
Insert
1 : 1

CONCLUSIONS

Model-based approach to facilitate mutation operators

- Definition → Wodel
- Testing & application → Model synthesis

FUTURE WORK

- Near misses generation
- Generate models to ensure the execution of the Wodel program leads to a correct model
- Methodology for the integral engineering of mutation operators

Pablo.GomezA@uam.es

@GomezAbajo