

Wodel: A Domain-Specific Language for Model Mutation

P. Gómez-Abajo , E. Guerra, J. de Lara
{Pablo.GomezA, Esther.Guerra, Juan.deLara}@uam.es



<http://www.miso.es>

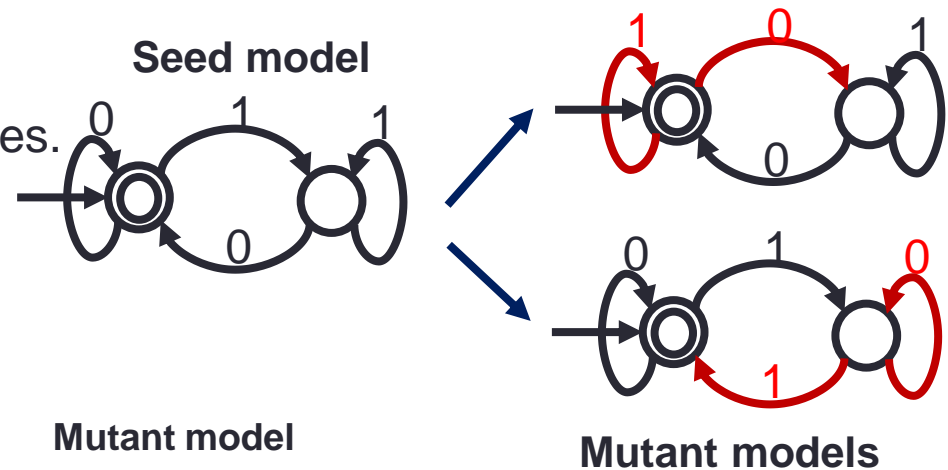
modelling & software engineering
research group

Universidad Autónoma de Madrid (Spain)

SAC'2016, Pisa (Italy)

What is a 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 testing.
 - Software product lines testing.
 - Automated generation of exercises.
 - Evolutionary algorithms.
 - ...



Seed model

```
...
if (a == true) then
...
```

Wodel

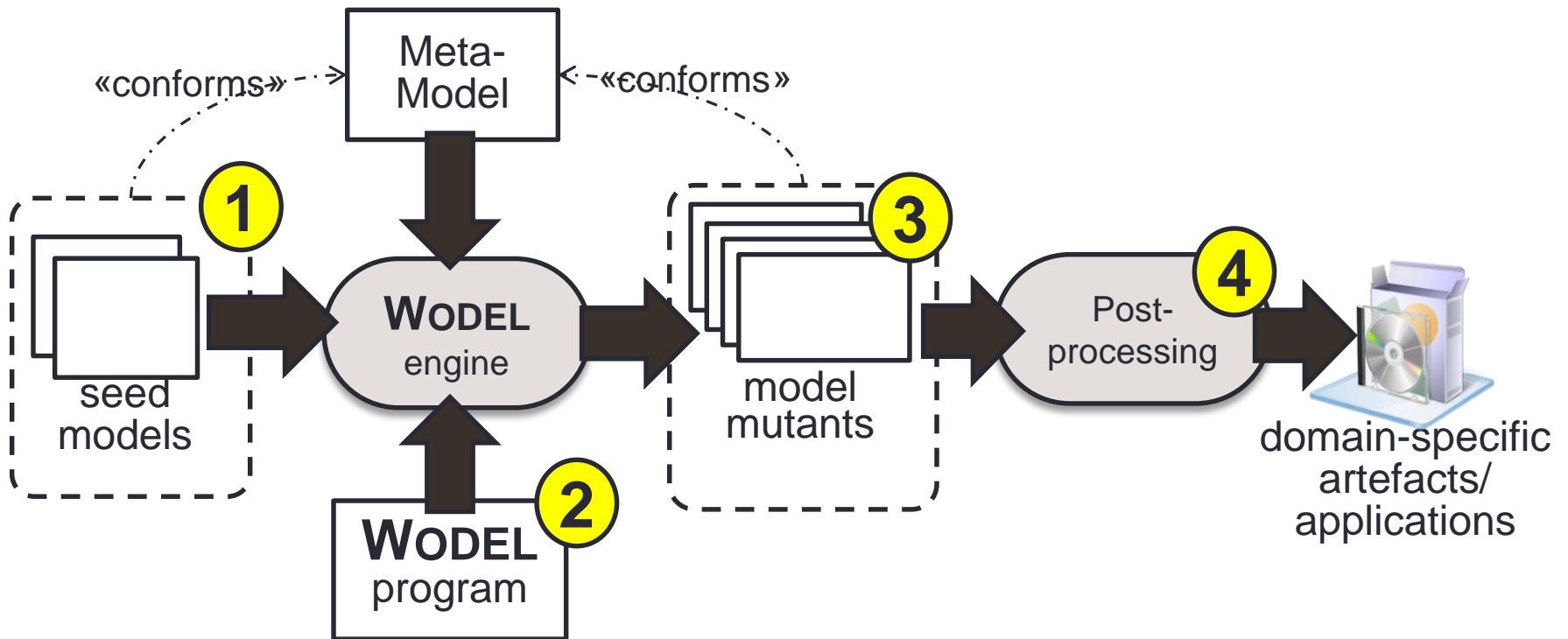
Mutant model

```
...
if (a != true) then
...
```

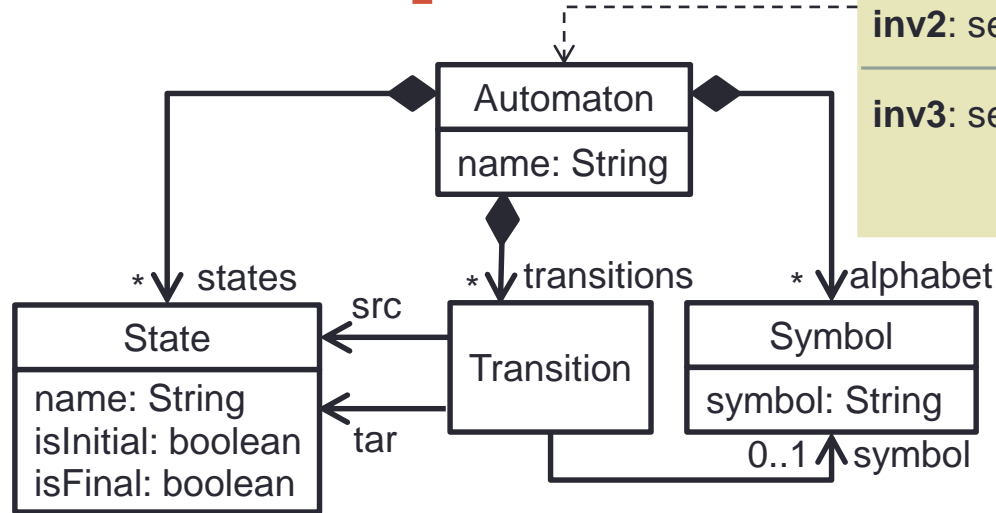
Motivation

- Existing frameworks for model mutation:
 - are specific for a language (e.g., logic formula).
 - or specific for a domain (e.g., testing).
 - mutation operators are manually encoded.
- We propose the DSL **Wodel** for model mutation:
 - high-level mutation primitives.
 - independence from target language and domain.
 - compiled into Java code.
 - extensible through post-processors.

Overview



Wodel: example



inv1: self.states->one(s | s.isInitial)

inv2: self.states->exists(s | s.isFinal)

inv3: self.alphabet->forAll (a1, a2 |
a1.symbol = a2.symbol
implies a1 = a2)

```
generate 3 mutants in "out/" from "evenBinary.fa"
```

```
metamodel "http://fa.com"
```

```
with commands {
```

```
  s0 = modify one State where {isFinal = true} with {reverse(isFinal)}
```

```
  s1 = create State with {isFinal = true}
```

```
  t0 = create Transition with {src = s0, tar = s1, symbol = one Symbol}
```

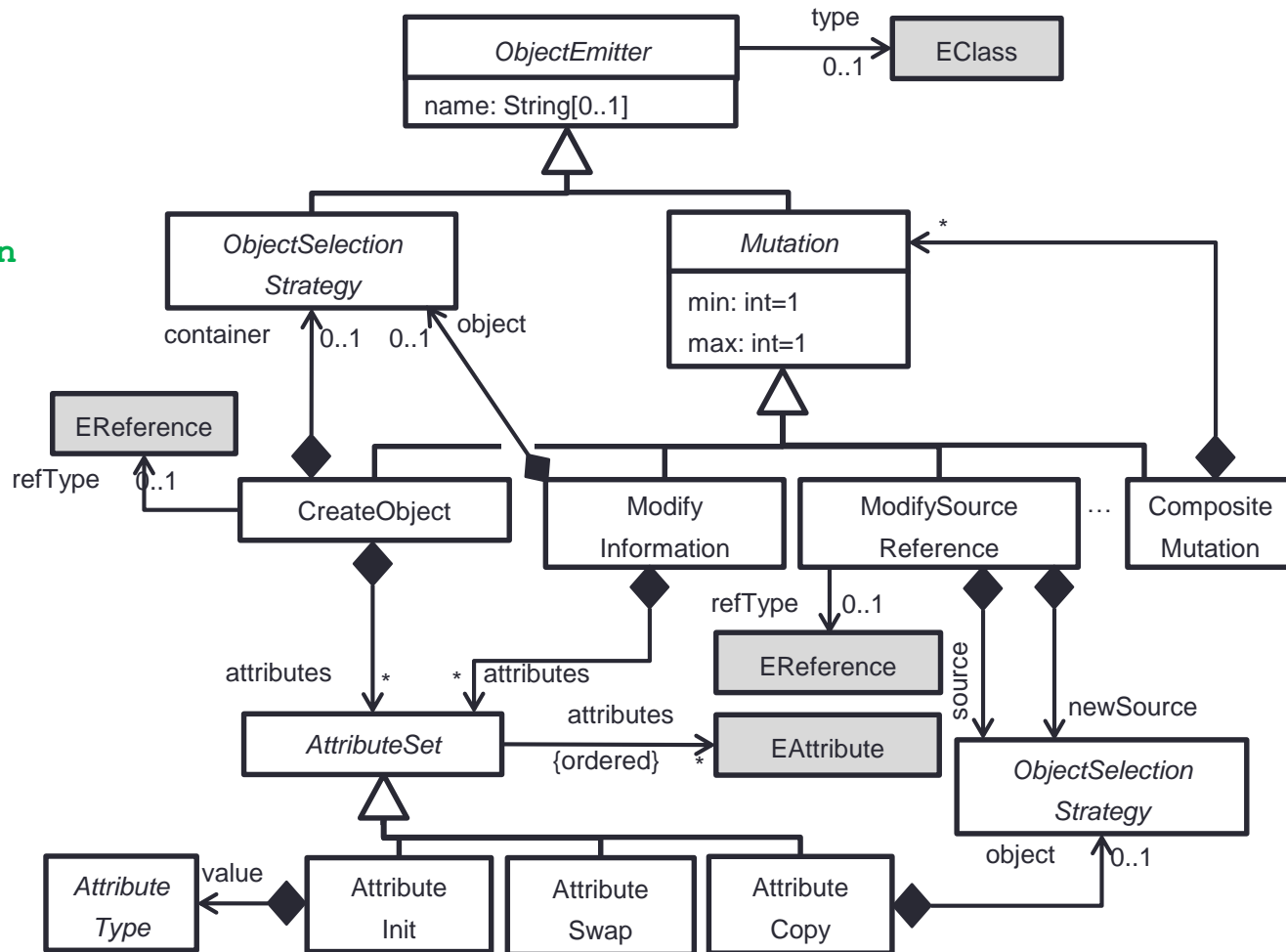
```
}
```

Wodel: mutation operators

```

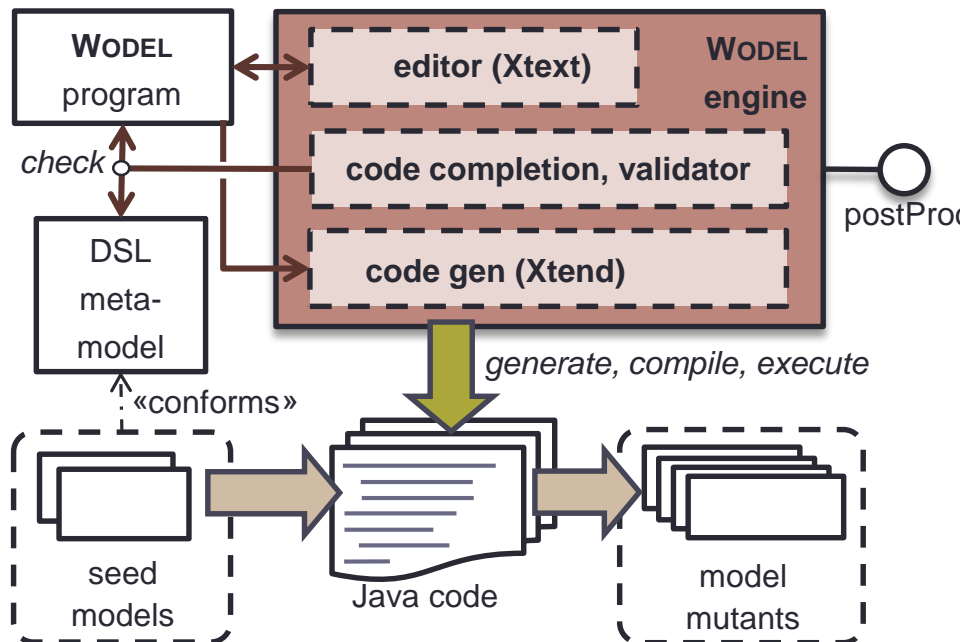
// object/ref creation
create State
create reference tar in
  one Transition
// object/ref modification
modify one State
  with {isFinal = true}
modify source src from
  one Transition
modify target tar from
  one Transition
// object/ref deletion
remove one State
remove reference tar in
  one Transition
// composite mutation
[
  s0 = create State
  modify s0 with {name = 's0'}
]

```

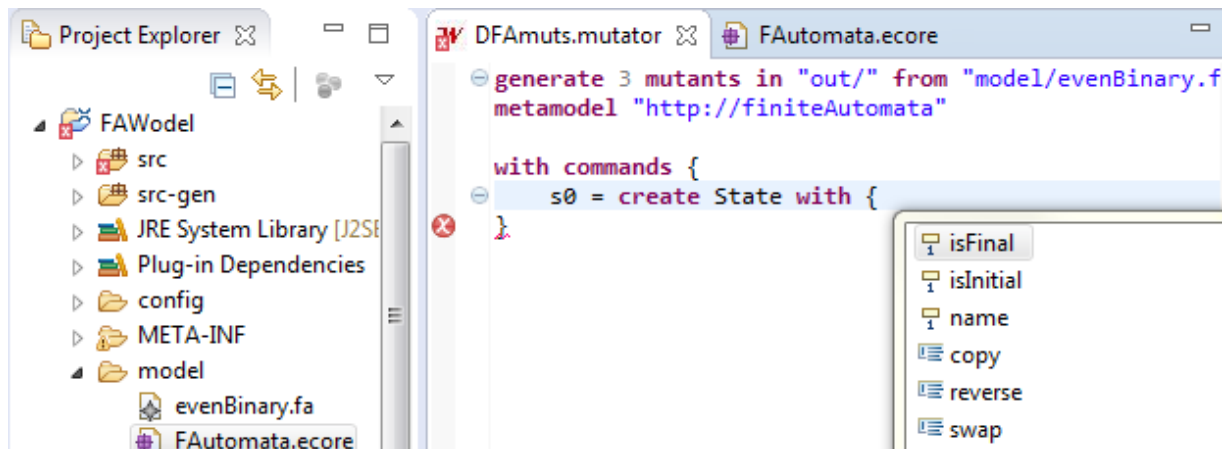


Tool support

Architecture of the environment



Screenshot of the Wodel IDE



Java code generation

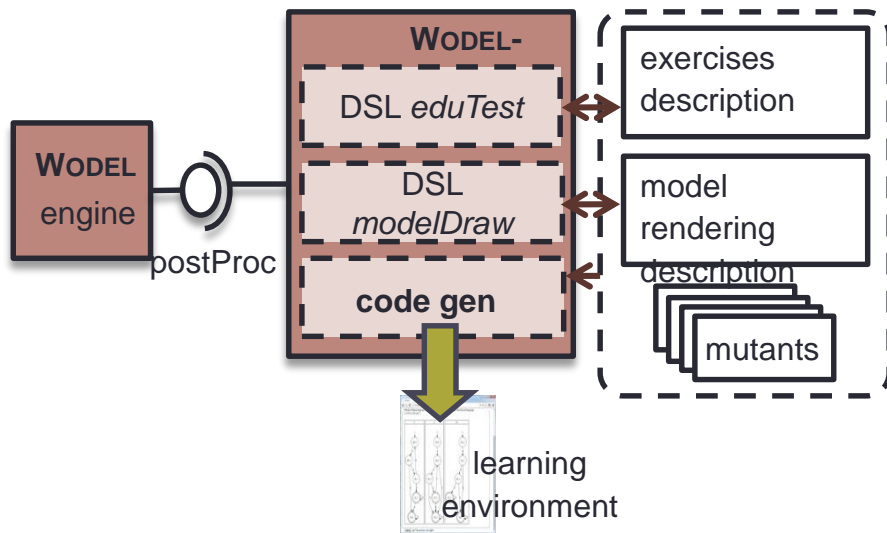
create Transition with {symbol = one Symbol}



```
1.....
2.// create transition
3.EClass transitionClass = (EClass)epackage.getEClassifier("Transition");
4.EObject transition = EcoreUtil.create(transitionClass);
5.
6.// search object automaton in model
7.EObject automaton = null;
8.for (TreeIterator<EObject> it = seed.getAllContents(); it.hasNext();) {
9.    automaton = it.next();
10.   if (automaton.eClass().getName().equals("Automaton")) {
11.       // add transition to automaton
12.       EStructuralFeature feature =
13.           automaton.eClass().getEStructuralFeature("transitions");
14.       ((List<EObject>)automaton.eGet(feature)).add(transition);
15.       // set random state as source of the transition
16.       feature = automaton.eClass().getEStructuralFeature("states");
17.       List<EObject> states = (List<EObject>)automaton.eGet(feature);
18.       EObject randomState = states.get(rand.nextInt(states.size()));
19.       feature = transitionClass.getEStructuralFeature("src");
20.       transition.eSet(feature, randomState);
21.....
```


Wodel-Edu: Model Mutation for the Generation of Exercises

Architecture of the Wodel-Edu plug-in



Generated application

Conclusions

- Wodel is a DSL for model mutation:
 - high-level mutation primitives.
 - domain-independent.
 - support for composite mutations.
 - non-repeated mutants, conformance checking.
 - compiled into Java.
 - extensible for different applications.
- Wodel-Edu: automated generation of exercises via model mutation

Future Work

- Extend Wodel with new mutation primitives and a rule policy language
- Develop new plugins for Wodel (e.g., for model-based testing, evolutionary computation...)
- Extend Wodel-Edu to support the generation of more complex learning environments

You can get the source code of this project on
GitHub:

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

Short video demo:

<https://youtu.be/GBfXH0Rf-fl>

Thank you!!

Pablo.GomezA@uam.es