A Domain-Specific Language for Model Mutation and its Application to the Automated Generation of Exercises

Pablo Gómez-Abajo, Esther Guerra, Juan de Lara

Universidad Autónoma de Madrid, 28049 Madrid, Spain,
{Pablo.GomezA,Esther.Guerra,Juan.deLara}@uam.es
http://www.miso.es

Abstract. Model-Driven Engineering (MDE) is a Software Engineering paradigm that uses models as main assets in all development phases. While many Domain-Specific Languages (DSLs) exist to specify model transformation [3], model simulation, or code generation, there is a lack of frameworks to specify and apply model mutations.
A *model mutation* is a kind of model manipulation that creates a set of variants (or *mutants*) of a seed model by the application of one or more *mutation operators*. Model mutation has many applications, for instance, in the areas of model transformation testing [1], model-based testing, education [2,4] or software testing verification.
In this poster, we present a domain-specific language called Wodel for the specification and generation of model mutants [2]. Wodel is domain-independent, as it can be used to generate mutants of models conformant to arbitrary meta-models. Its development environment is extensible, permitting the incorporation of post-processors for different applications. In particular, we describe Wodel-Edu, a post-processing extension directed to the automated generation of exercises for particular domains and their automated correction. We show the application of Wodel-Edu to the generation of exercises for deterministic automata, and report on an evaluation of the quality of the generated exercises, obtaining overall good results.

Keywords: Domain-Specific Languages, Model-Driven Engineering, Model Mutation, Education, Automated Generation of Exercises

References

1. Aranega, V., Mottu, J., Etien, A., Degueule, T., Baudry, B., Dekeyser, J.: Towards an automation of the mutation analysis dedicated to model transformation. STVR 25(5-7), 653–683 (2015)
2. Gómez-Abajo, P., Guerra, E., de Lara, J.: Wodel: a domain-specific language for model mutation. In: SAC. pp. 1968–1973. ACM (2016)
3. Mens, T., Gorp, P.V.: A taxonomy of model transformation. Electr. Notes Theor. Comput. Sci. 152, 125–142 (2006)
4. Sadigh, D., Seshia, S.A., Gupta, M.: Automating exercise generation: A step towards meeting the MOOC challenge for embedded systems. In: WESE. ACM (2013)

UNIVERSIDAD AUTÓNOMA DE MADRID
http://www.miso.es
modelling & software engineering research group
Escuela Politécnica Superior

Wodel: A DSL for Model Mutation; and Wodel-Edu: its Application to the Automated Generation of Exercises

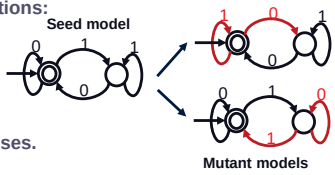
Pablo Gómez-Abajo, Esther Guerra, Juan de Lara
Universidad Autónoma de Madrid (Spain)

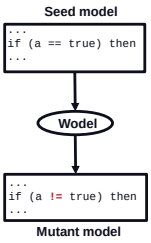
What is a Model Mutation?

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.





Motivation

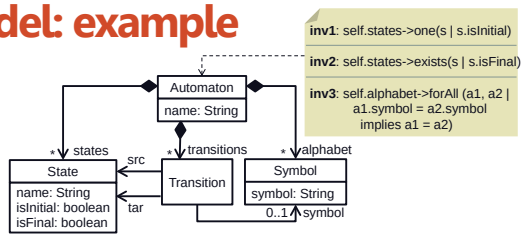
Existing frameworks for model mutation:

- 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.

Wodel: example

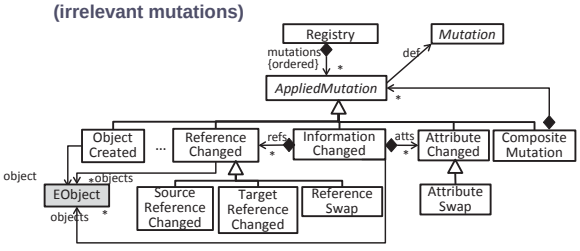



```
generate 3 mutants in "out/" from "evenBinary.fa"
metamodel "http://fa.com"

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}
}
```

Mutations Registry

- Optional, it is activated through the preferences page
- Useful when generating text options in test exercises
- References to seed models and mutant models
- Optionally, the framework can reduce the registry (irrelevant mutations)



Blocks and OCL Constraints

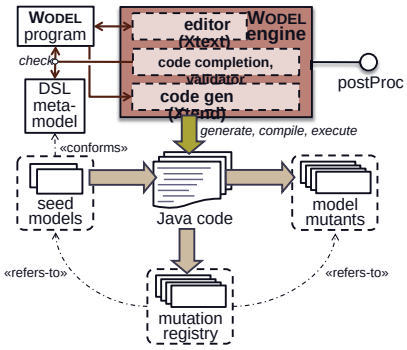
Wodel supports mutation blocks:

- Mutants generation by stages
- A block can take as seed models the mutants generated in previous blocks
- Folders hierarchy for mutants identification
- Duplicated mutants control with directive repeat=no

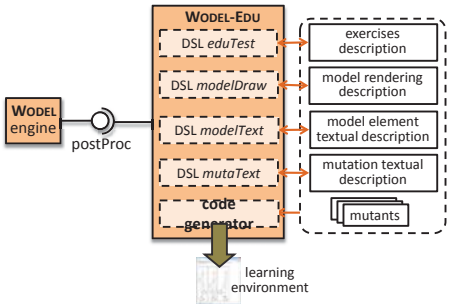
OCL Constraints in Wodel Code:

- Applied over the generated mutants, although they are not in the domain meta-model

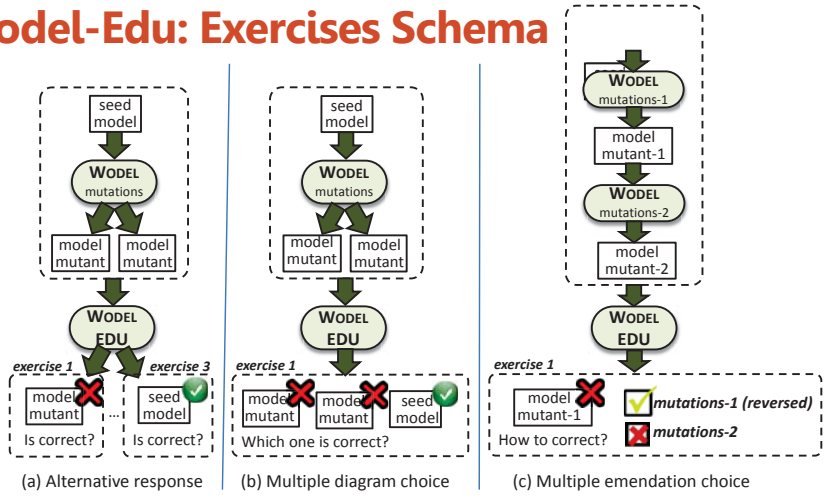
Tool Architecture



Wodel-Edu: Architecture



Wodel-Edu: Exercises Schema



(a) Alternative response (b) Multiple diagram choice (c) Multiple emendation choice

DSM-TP 2016, Genève, Switzerland, August 22-26 2016