Adapting TDL to Provide Testing Support for Executable DSLs

Faezeh Khorram*, *Erwan Bousse, Jean-Marie Mottu, Gerson Sunyé

NaoMod group, LS2N lab, University of Nantes, IMT Atlantique, France
17th European Conference on Modelling Foundations and Applications (ECMFA2021)

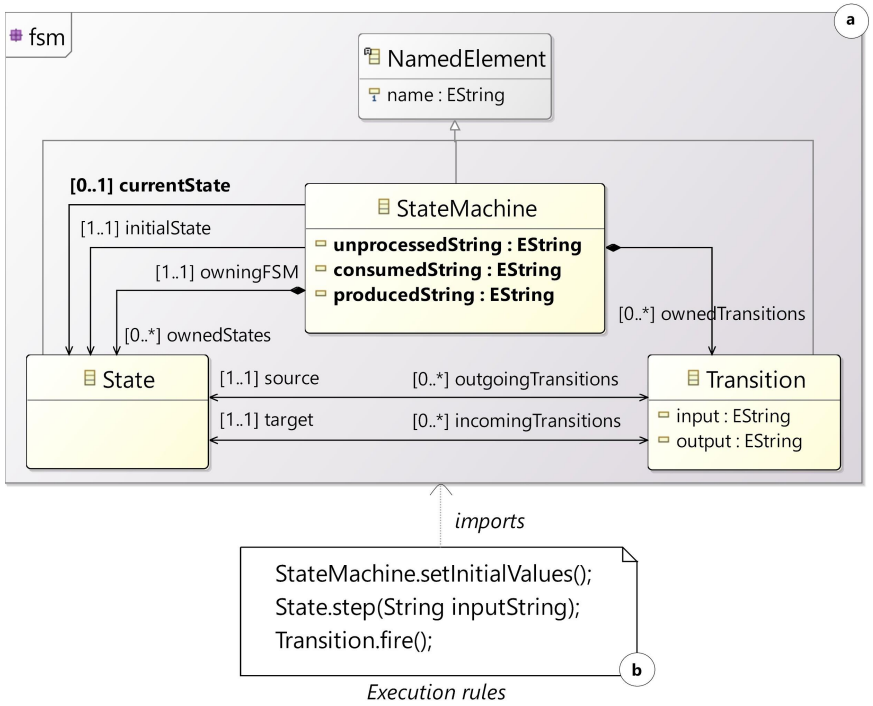
"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skołodowska-Curie grant agreement No 813884".

lowcomote

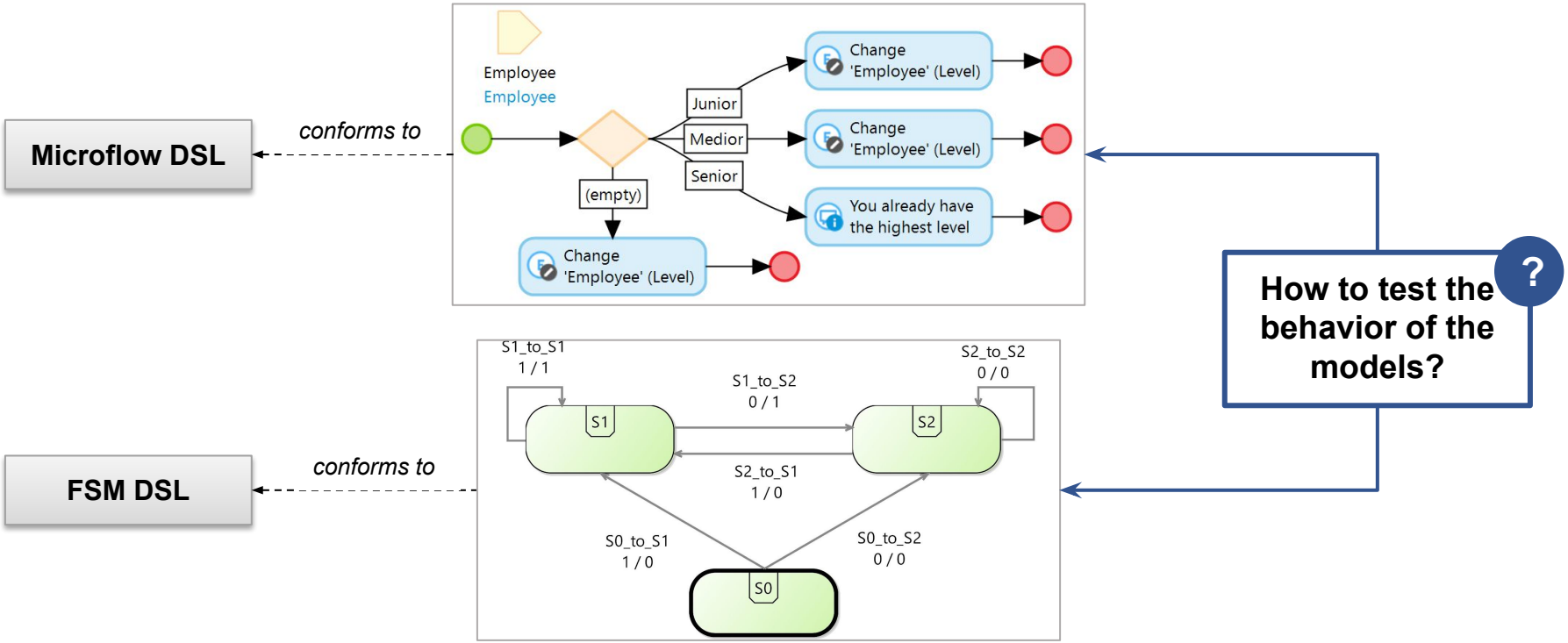
Context: Executable DSLs

Composed of two parts:

1. *Abstract Syntax*: Defining the domain concepts
2. *Operational Semantics* (Interpreter)
   a. the definition of the possible **runtime states** of a model under execution
   b. a set of **execution rules** that define how such a runtime state changes over time

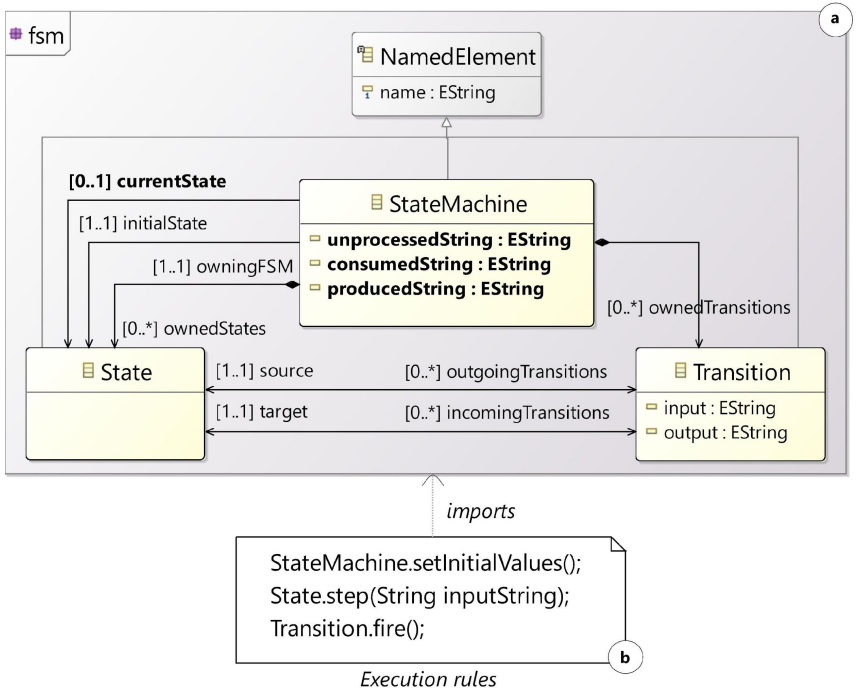


Heterogeneity of the DSLs

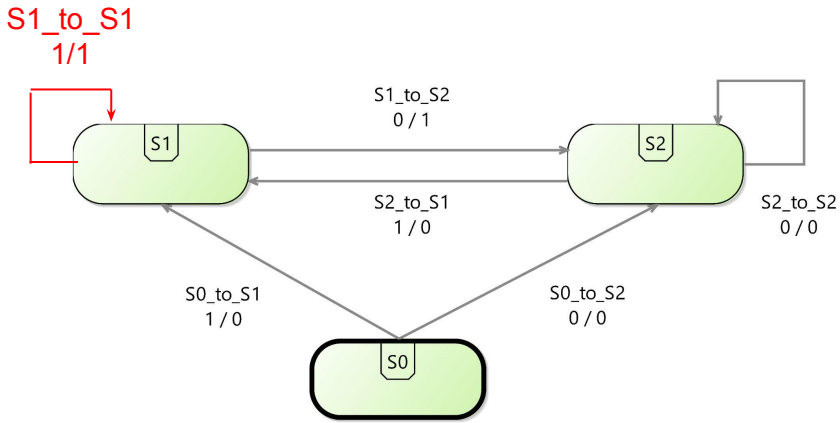


Running Example: DSL and Model

Executable FSM DSL (xFSM)

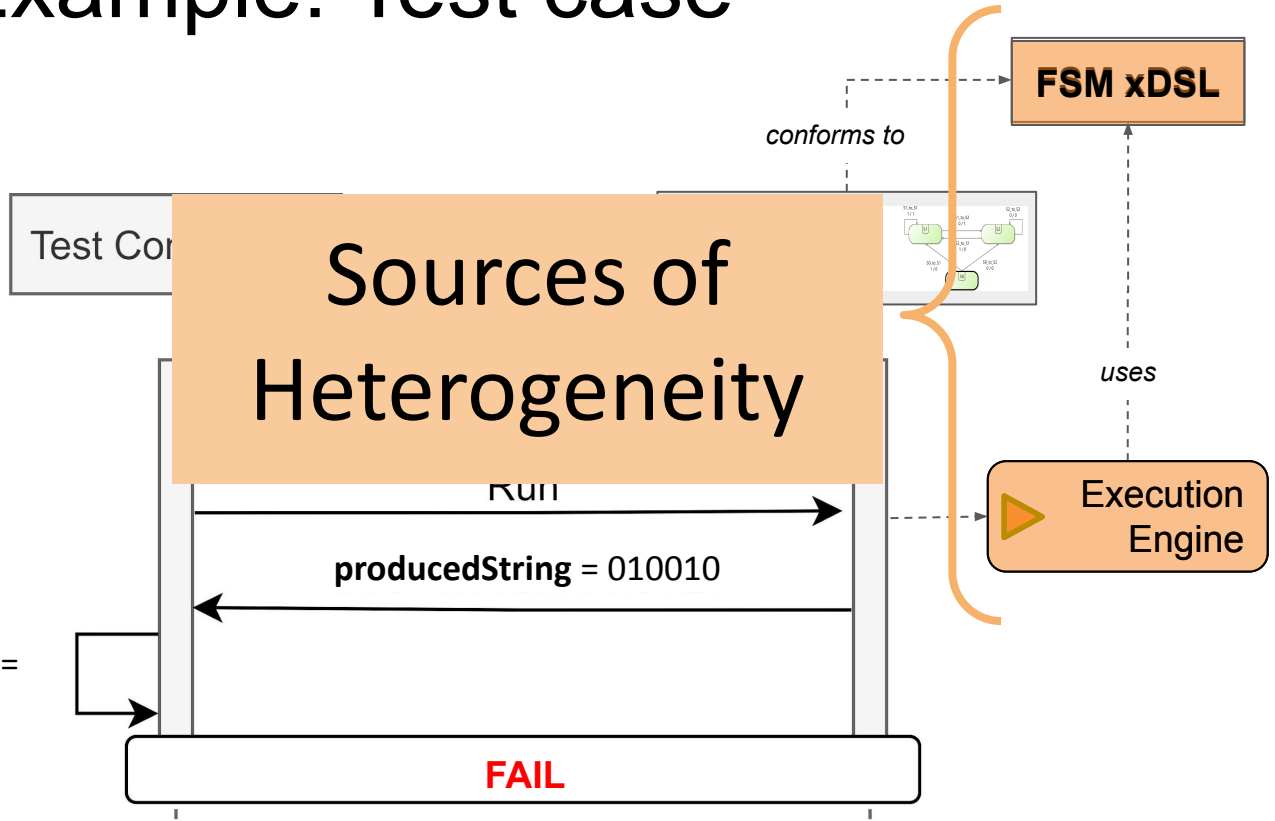


Executable FSM Model: Bit shifting FSM
The model under test



Input data: unprocessedString = 10010110
Expected output: producedString = 01001011

Running Example: Test case



Check against:
Expected **producedString** = 01001011

Main observation

Lack of a generic testing approach applicable to a wide range of xDSLs

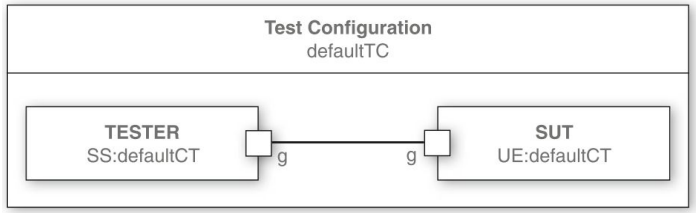
A generic testing approach must provide a *test language*

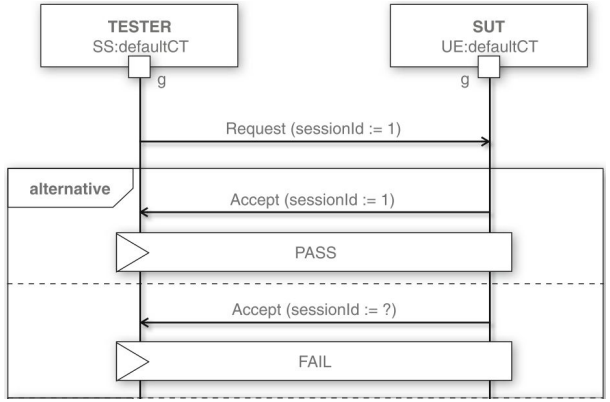
- Allowing the domain expert to write test cases using *domain* concepts
- Connected to the *semantics* of the xDSL to execute test cases on the models
- Providing facilities to analyze the runtime state of the tested model

Problem Statement: TDL Testing Language

Candidate: Test Description Language (TDL) [1]

- ✔ A standardized language for the specification of test descriptions
- ✔ Not specific to any specific GPL or DSL
- ✔ Designed as a simple language for testers lacking programming knowledge, so a good fit for domain experts





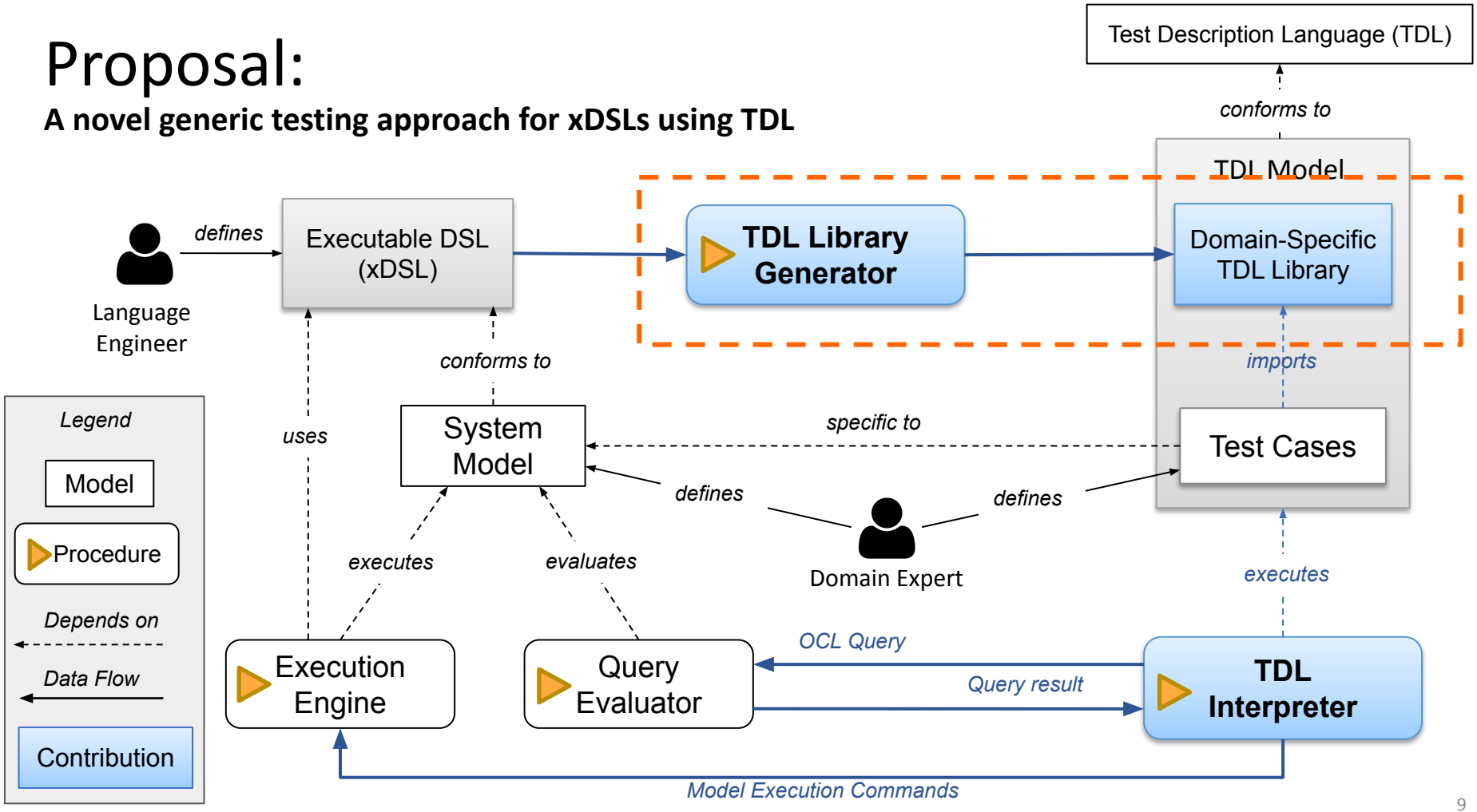
[1] Makedonski, P., Adamis, G., Käärik, M. et al. "Test descriptions with ETSI TDL". Software Quality Journal, vol. 27, no.2 , 885–917 (2019)

Problem Statement: TDL Limitations

- *Too generic*: the domain expert must first define the required domain-specific concepts, and then write test cases
- No clear way to enable TDL test cases to execute models conforming to an xDSL (i.e., the model under test)
- Relying on a simple representation of the expected behavior of the SUT

Proposal:

A novel generic testing approach for xDSLs using TDL



TDL Test Cases using Model-Execution Commands

Sending test input data: the bit shifting FSM in a runtime state

Request for model execution by sending **'runModel'** command, and get the current state of the model using **'getModelState'**, both provided by *common package*

Oracle with expected output data

```
1  Package bitShiftingFSM_TestSuite {
2    Import all from common;
3    Import all from fsmSpecificTypes;
4    Import all from testConfiguration;
5
6    StateMachine stateMachineNewState(
7      _name = "BitShifting");
8    State S2 (_name = "S2");
9
10   Test Description bitShiftingGenericTest uses
     configuration fsmConfiguration{
11     tester.genericGate sends
     stateMachineNewState
12       (unprocessedString = "10010110")
13       to fsm.genericGate;
14     tester.genericGate sends runModel
15       to fsm.genericGate;
16     tester.genericGate sends getModelState
17       to fsm.genericGate;
18     fsm.genericGate sends stateMachineNewState
19       (producedString = "01001011")
20       to tester.genericGate;
21   }
```

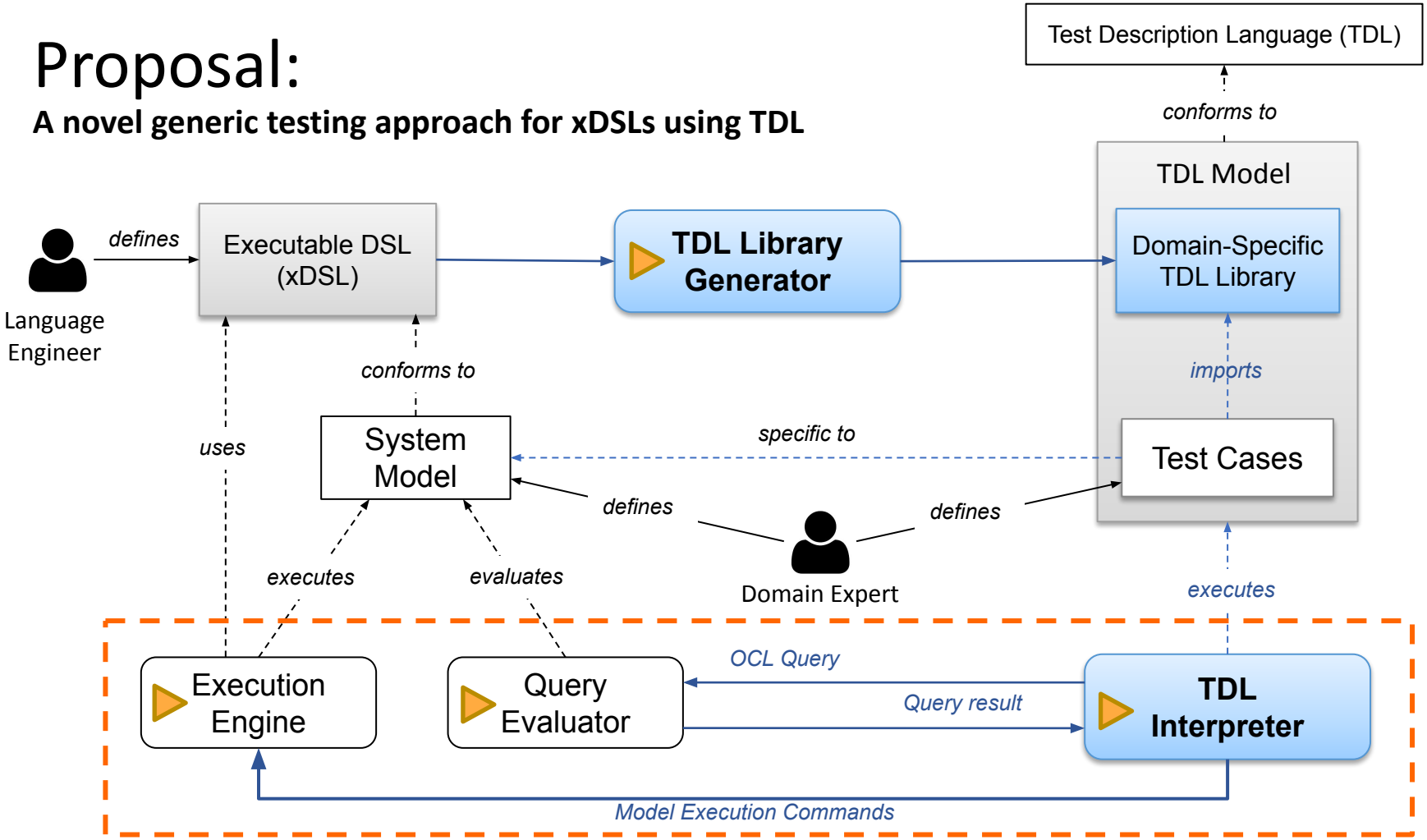
TDL Test Cases using OCL Queries

Using OCL queries to check the value of a specific dynamic feature after executing the model in a specific runtime state

```
22    Test Description bitShiftingOclTest uses
      configuration fsmConfiguration{
23      tester.genericGate sends
      stateMachineNewState
24        (unprocessedString = "000101010")
25        to fsm.genericGate;
26      tester.genericGate sends runModel
27        to fsm.genericGate;
28      tester.oclGate sends oclQuery
29        (query = "self.currentState")
30        to fsm.oclGate;
31      fsm.oclGate sends S2 to tester.oclGate;
32    }
33  }
```

Proposal:

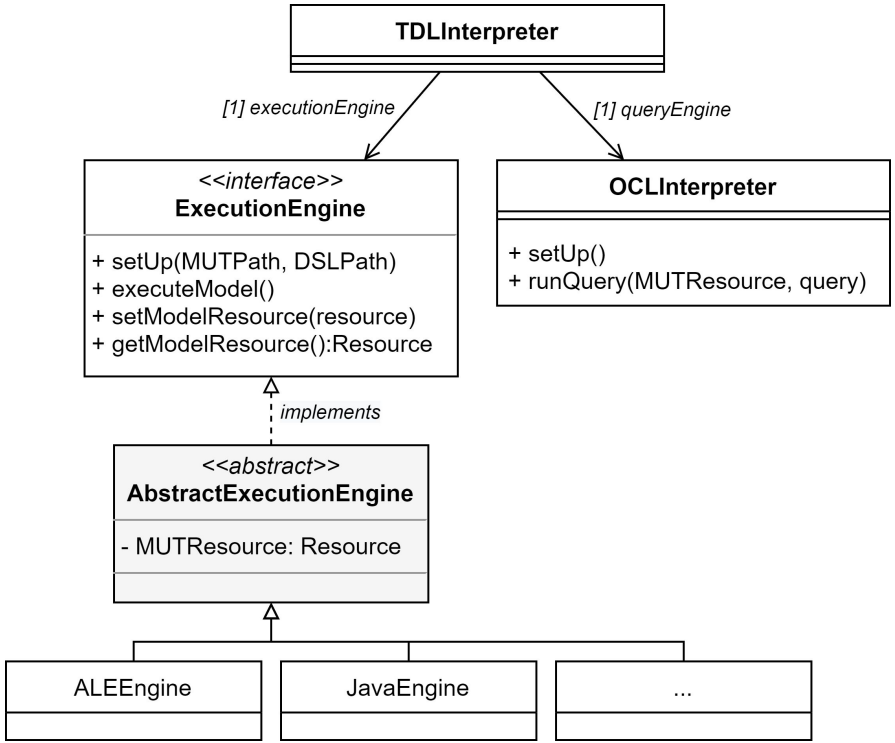
A novel generic testing approach for xDSLs using TDL



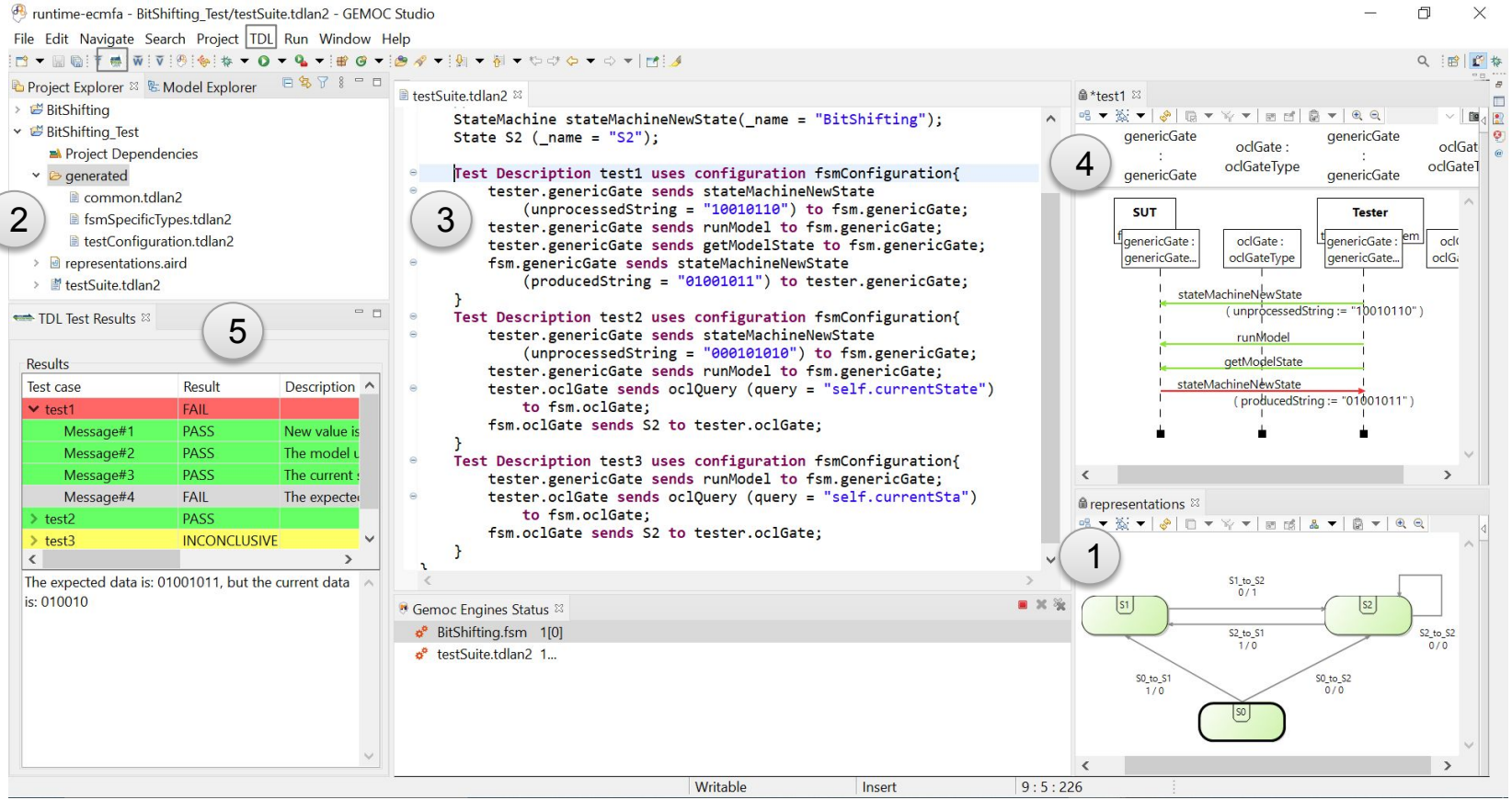
TDL Interpreter

- Connected to two external components
  - Execution Engine
  - OCL Query Evaluator (Using Eclipse OCL API)

- Definition of an interface to decouple the TDL Interpreter from various execution engines



Tool Support



Evaluation

xDSL	DSL size		Model	TDL Library Size (LoC generated)	Test suite size (LoC)
	Abstract syntax size (n. of EClasses)	Semantics size (LoC)			
xFSM	3	K3: 110 ALE: 90	TrafficLight	111	4
			BitShifting		4
			EdgeDetector		8
			ToLowerCase		5
			ToUpperCase		5
xArduino	59	K3: 667 ALE:421	Servo9g	259	4
			ActiveWaitIR		4
			TurnOnLED		2
			ServoIrButton		4
xBPMN	39	ALE: 318	VerifyUserAccess	202	2
			PromoteEmployee		4

Future Work

- Exploring the approach on more complex xDSLs
  - Testing models with more complex runtime states
- Extending the proposed approach
  - Supporting models conforming to several interconnected xDSLs
  - Providing testing support for Reactive DSLs

Adapting TDL to Provide Testing Support for Executable DSLs

Faezeh Khorram*, *Erwan Bousse, Jean-Marie Mottu, Gerson Sunyé

NaoMod group, LS2N lab, University of Nantes, IMT Atlantique, France
17th European Conference on Modelling Foundations and Applications (ECMFA2021)

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skołodowska-Curie grant agreement No 813884".

lowcomote