Partial Test Oracle in Model Transformation Testing

ICMT'13 – Budapest

Olivier Finot, Jean-Marie Mottu,

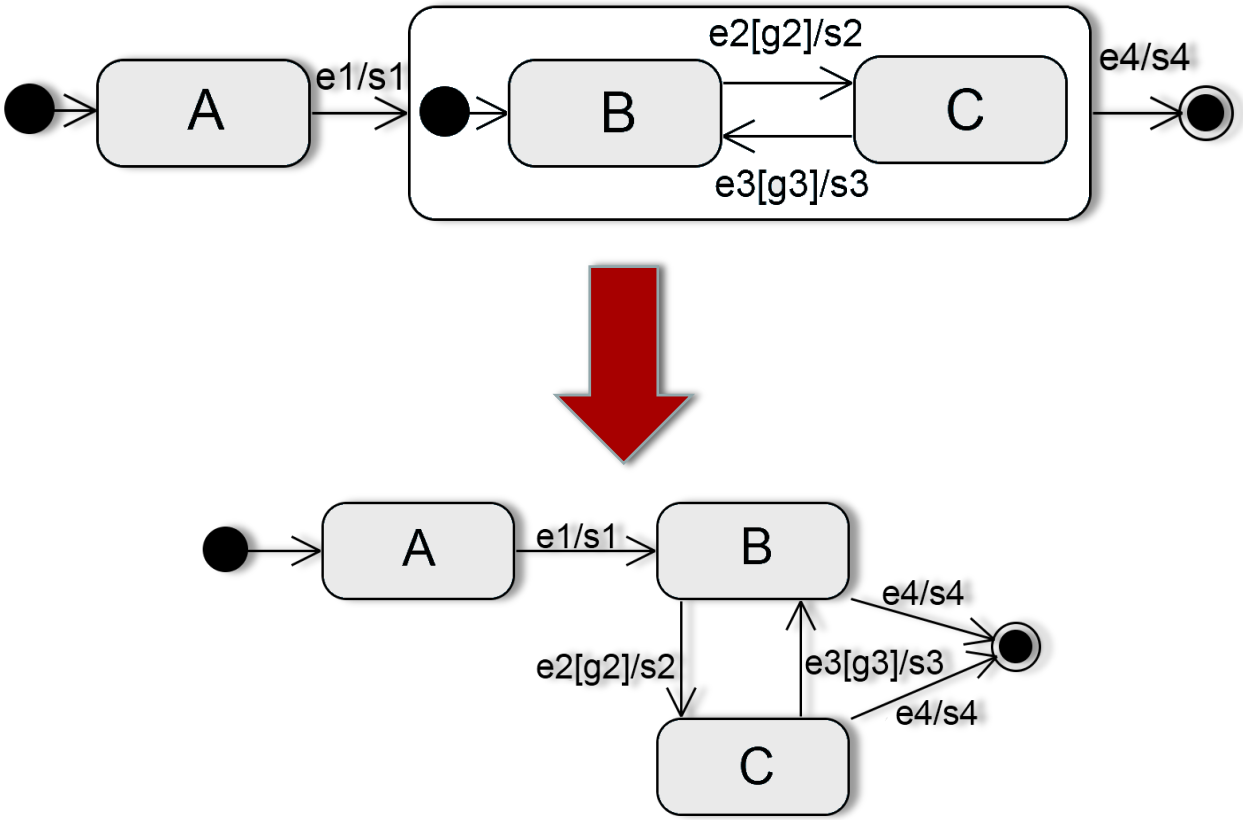
Gerson Sunyé, Christian Attiogbe

Outline

- Case Study : FSM2FFSM
- Introduction: Model Transformation Testing
- Problem: Tediousness of Creating and Analysing Models
- The Story So Far!
- Problem: Difficult To Predict Expected Models
- Filtered Model Comparison
- Tool & Experimentation
- Perspectives
- Conclusion

Case Study : FSM2FFSM

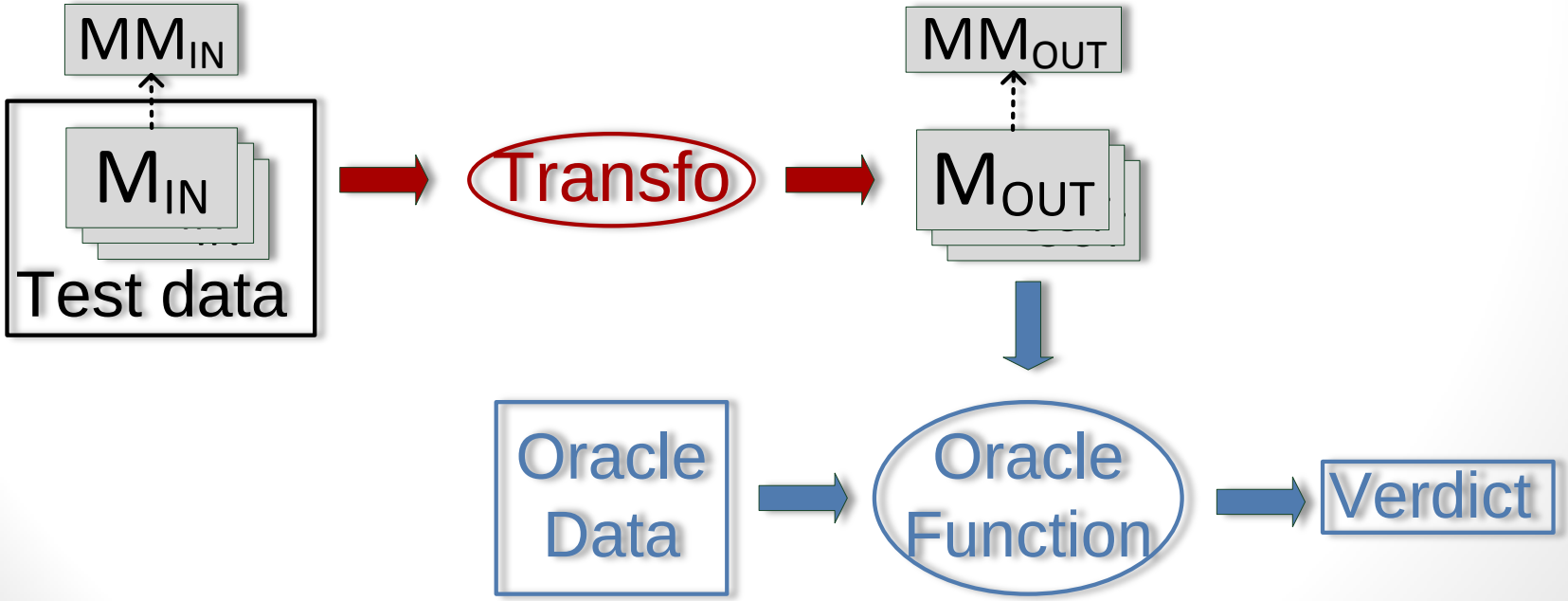
- Flattening of Hierarchical State Machines



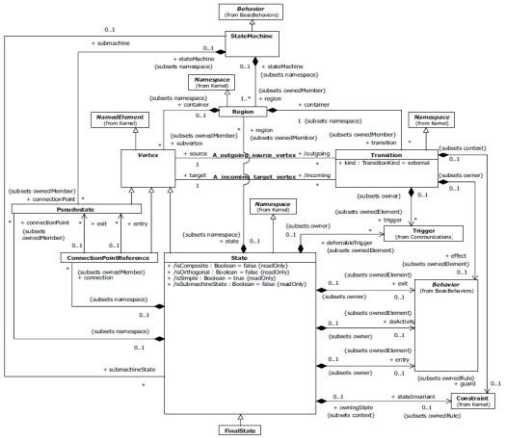
Model Transformation Testing

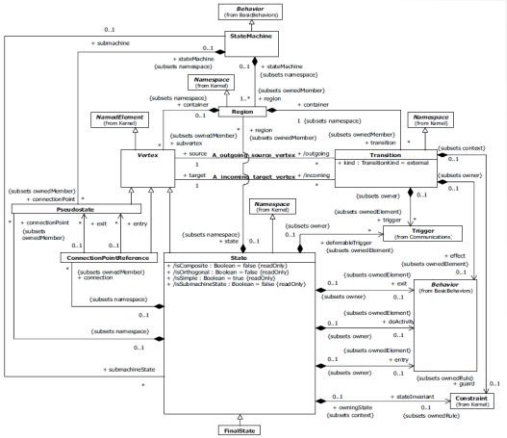
- Ensure Model Transformation Quality
- Prevent Fault Propagation

- Test Models are generated
- Test Oracles control that output models satisfy the specifications



Problem: Tediousness of Creating and Analysing Models



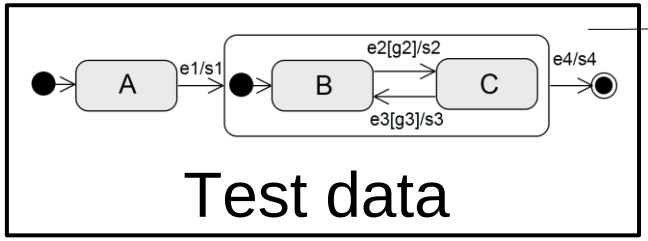


Specification

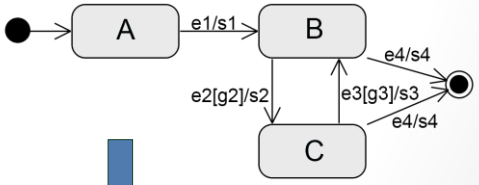
conforms

satisfies

conforms



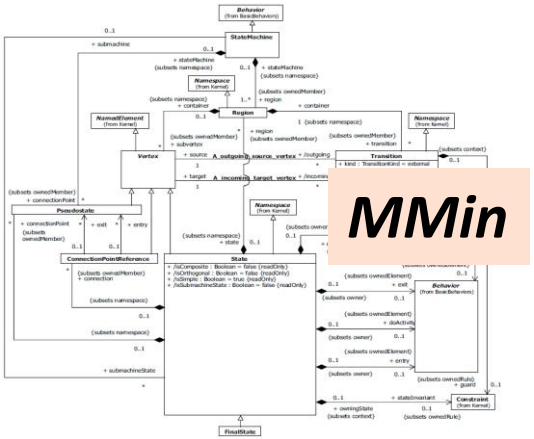
Transfo



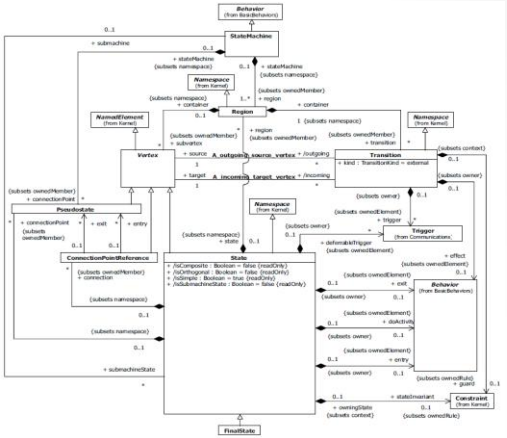
Oracle Data

Oracle Function

Problem: Tediousness of Creating and Analysing Models



MMin

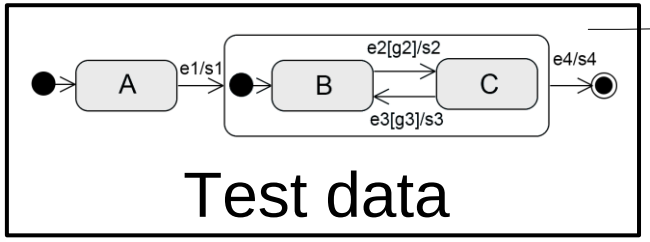


Specification

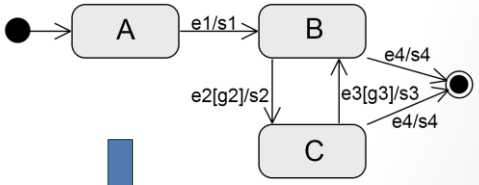
conforms

satisfies

conforms



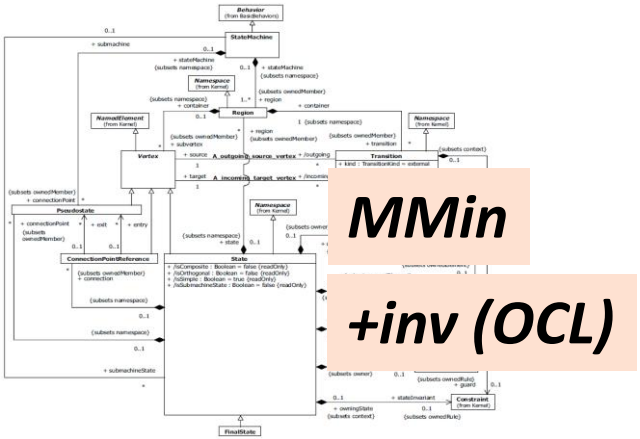
Transfo



Oracle Data

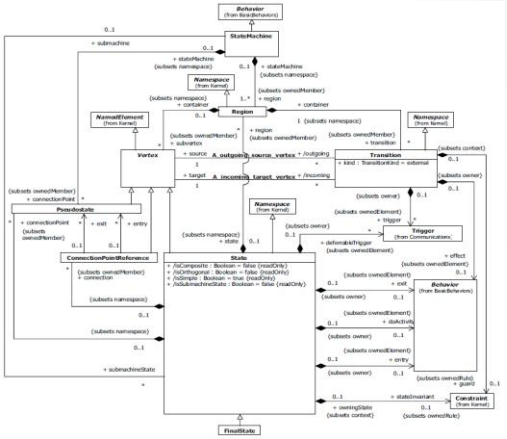
Oracle Function

Problem: Tediousness of Creating and Analysing Models



MMin

+inv (OCL)

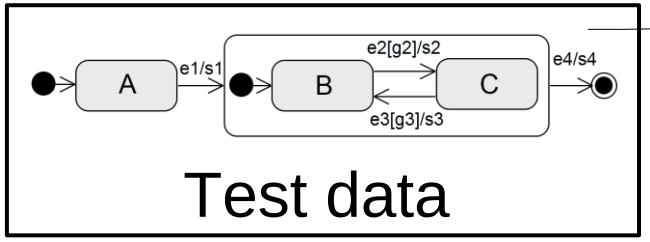


Specification

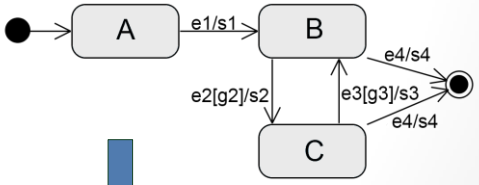
conforms

satisfies

conforms



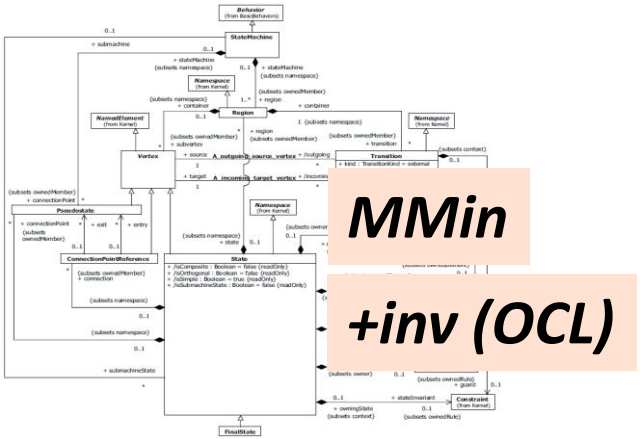
Transfo



Oracle Data

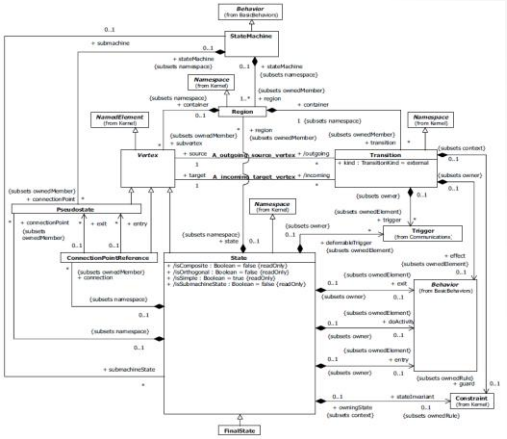
Oracle Function

Problem: Tediousness of Creating and Analysing Models



MMin

+inv (OCL)

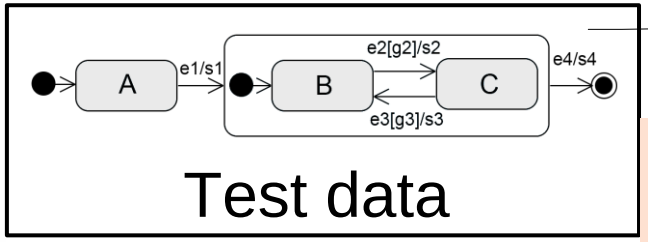


Specification

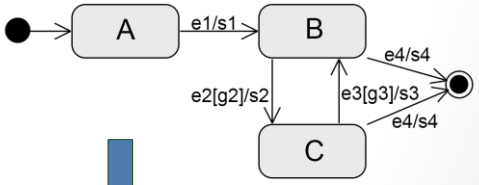
conforms

satisfies

conforms



Transfo

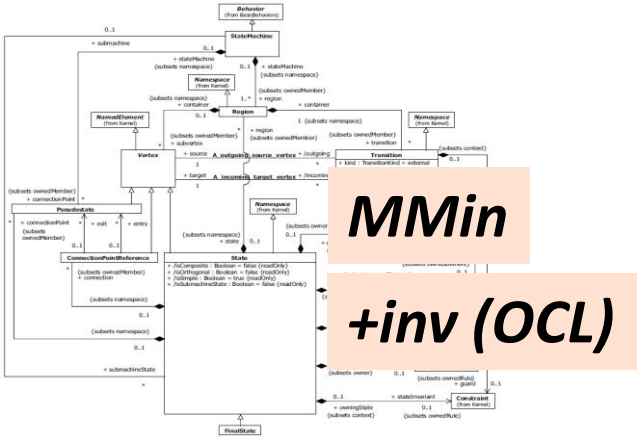


pre (OCL)

Oracle Data

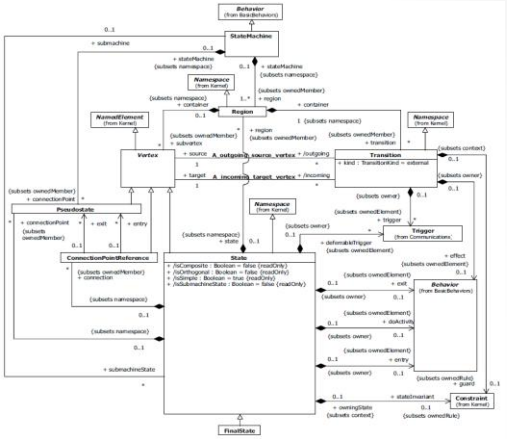
Oracle Function

Problem: Tediousness of Creating and Analysing Models



MMin

+inv (OCL)

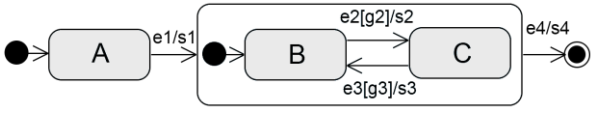


Specification

conforms

satisfies

conforms

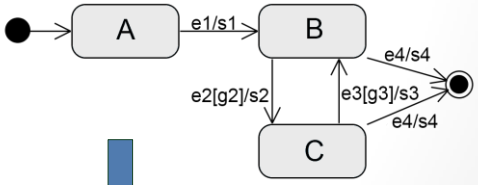


Test data

Test knowledge

Transfo

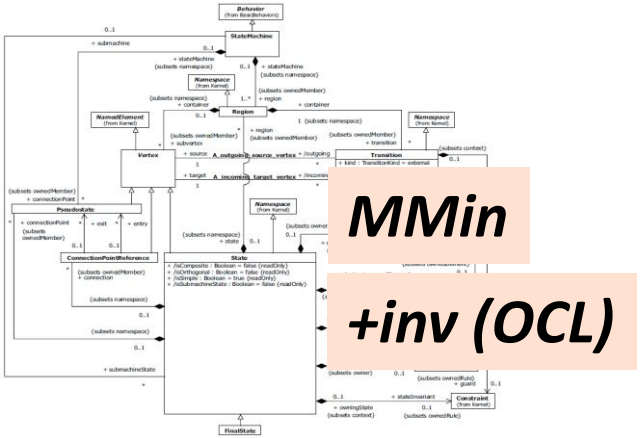
pre (OCL)



Oracle Data

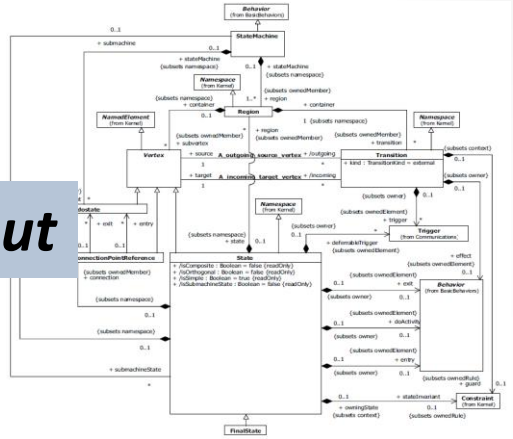
Oracle Function

Problem: Tediousness of Creating and Analysing Models



MMin

+inv (OCL)



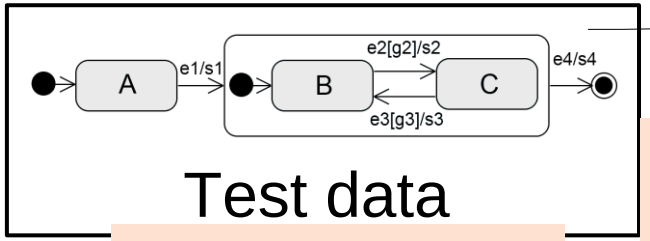
MMout

Specification

conforms

satisfies

conforms

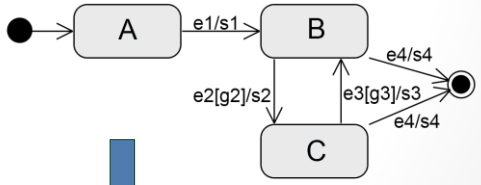


Test data

Test knowledge

Transfo

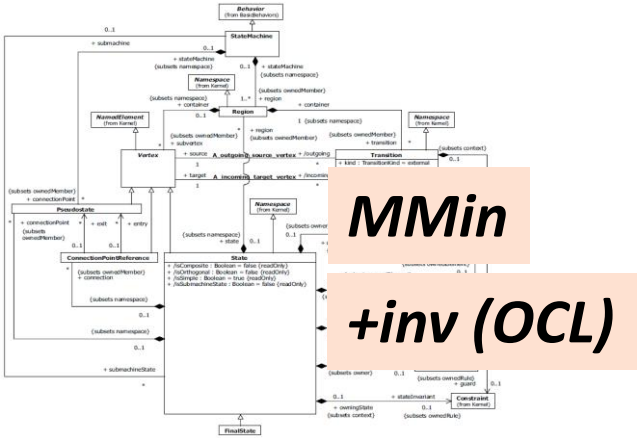
pre (OCL)



Oracle Data

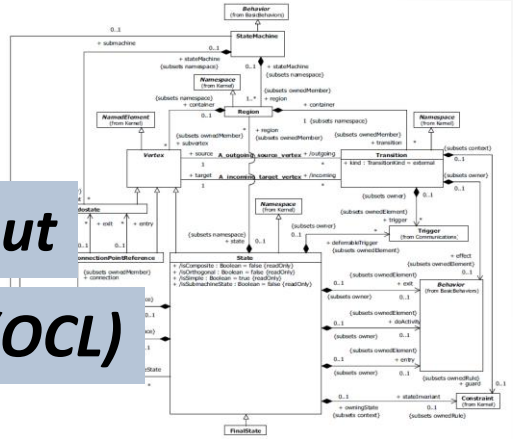
Oracle Function

Problem: Tediousness of Creating and Analysing Models



MMin

+inv (OCL)



MMout

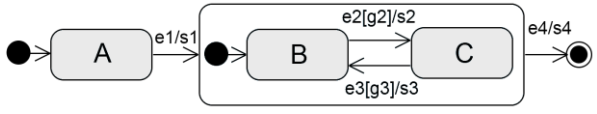
+inv (OCL)

Specification

conforms

satisfies

conforms

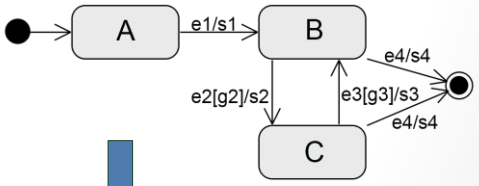


Test data

Test knowledge

Transfo

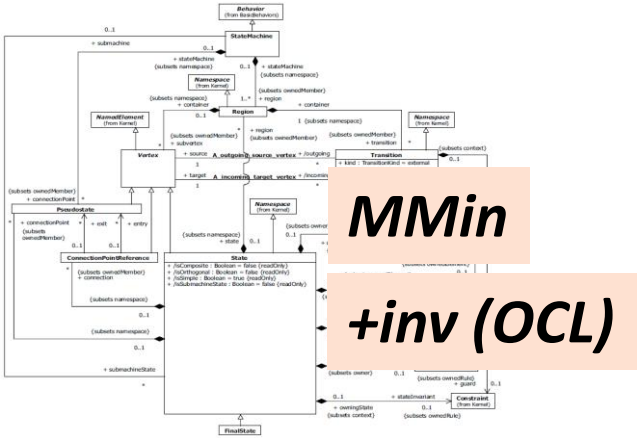
pre (OCL)



Oracle Data

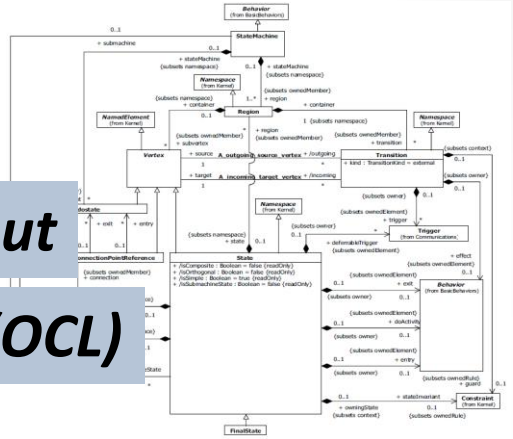
Oracle Function

Problem: Tediousness of Creating and Analysing Models



MMin

+inv (OCL)



MMout

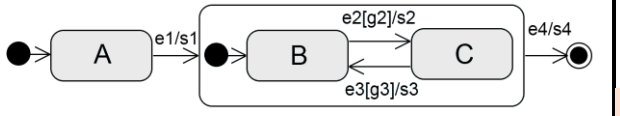
+inv (OCL)

Specification

conforms

satisfies

conforms



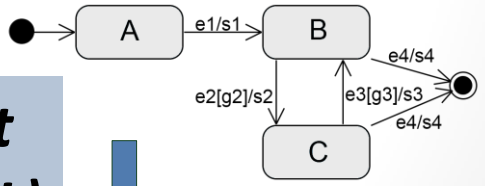
Test data

Test knowledge

Transfo

pre (OCL)

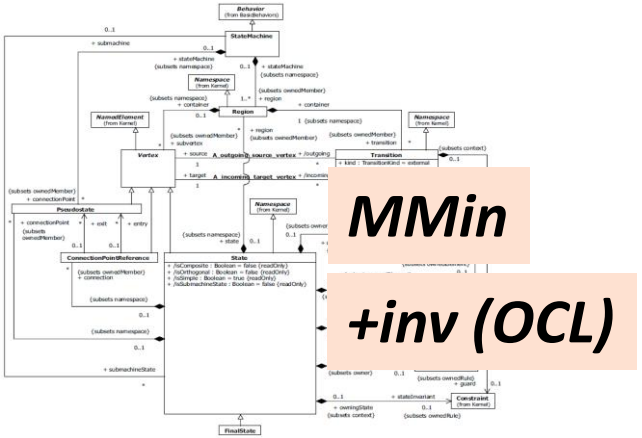
post (OCL)



Oracle Data

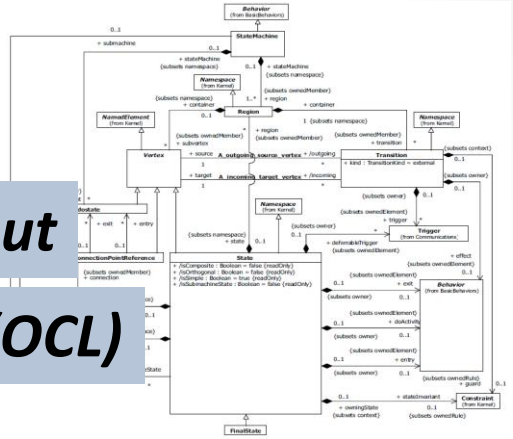
Oracle Function

Problem: Tediousness of Creating and Analysing Models



MMin

+inv (OCL)



MMout

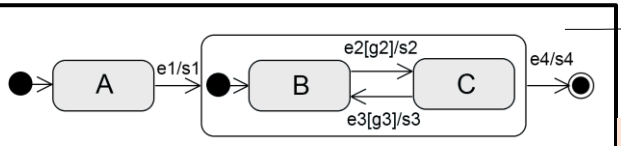
+inv (OCL)

Specification

satisfies

conforms

conforms



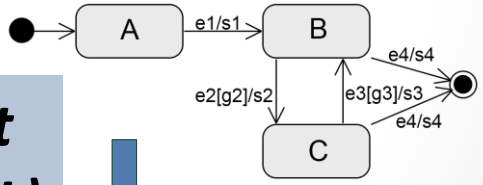
Test data

Test knowledge

Transfo

pre (OCL)

post (OCL)

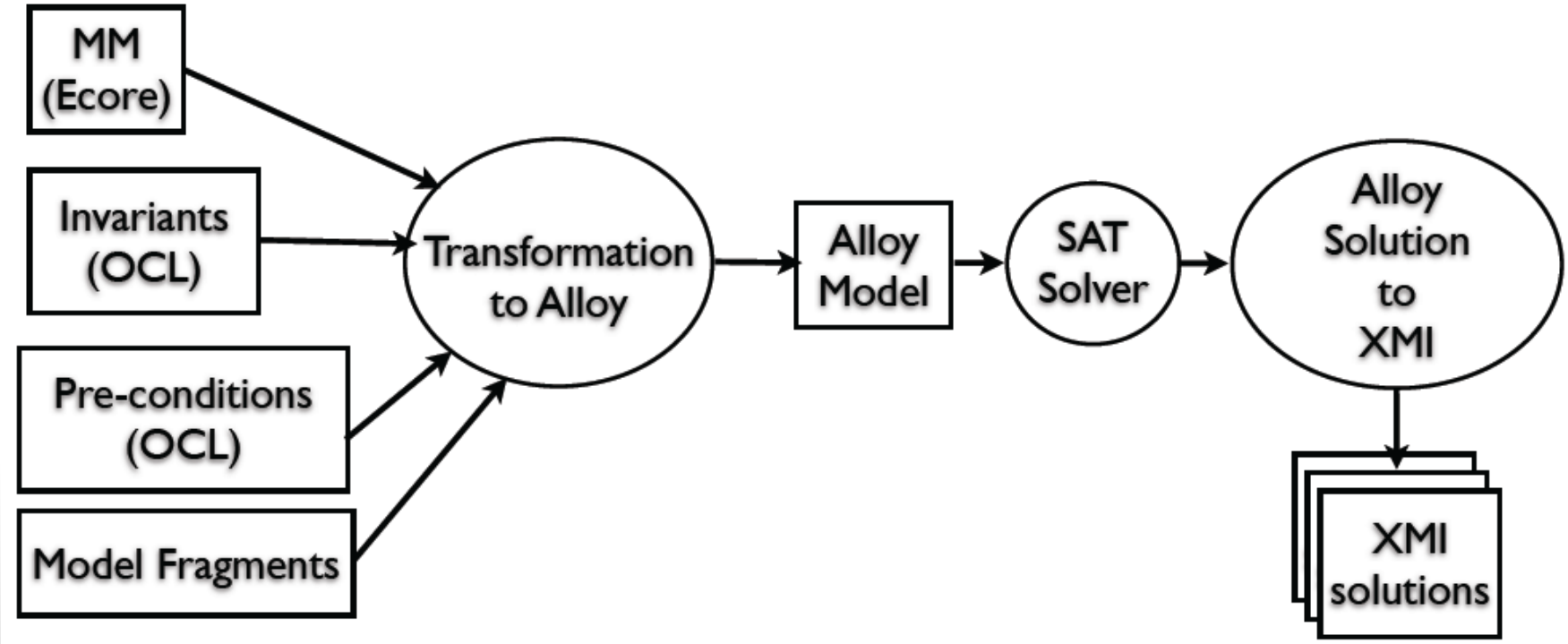


Oracle Data

Oracle Function

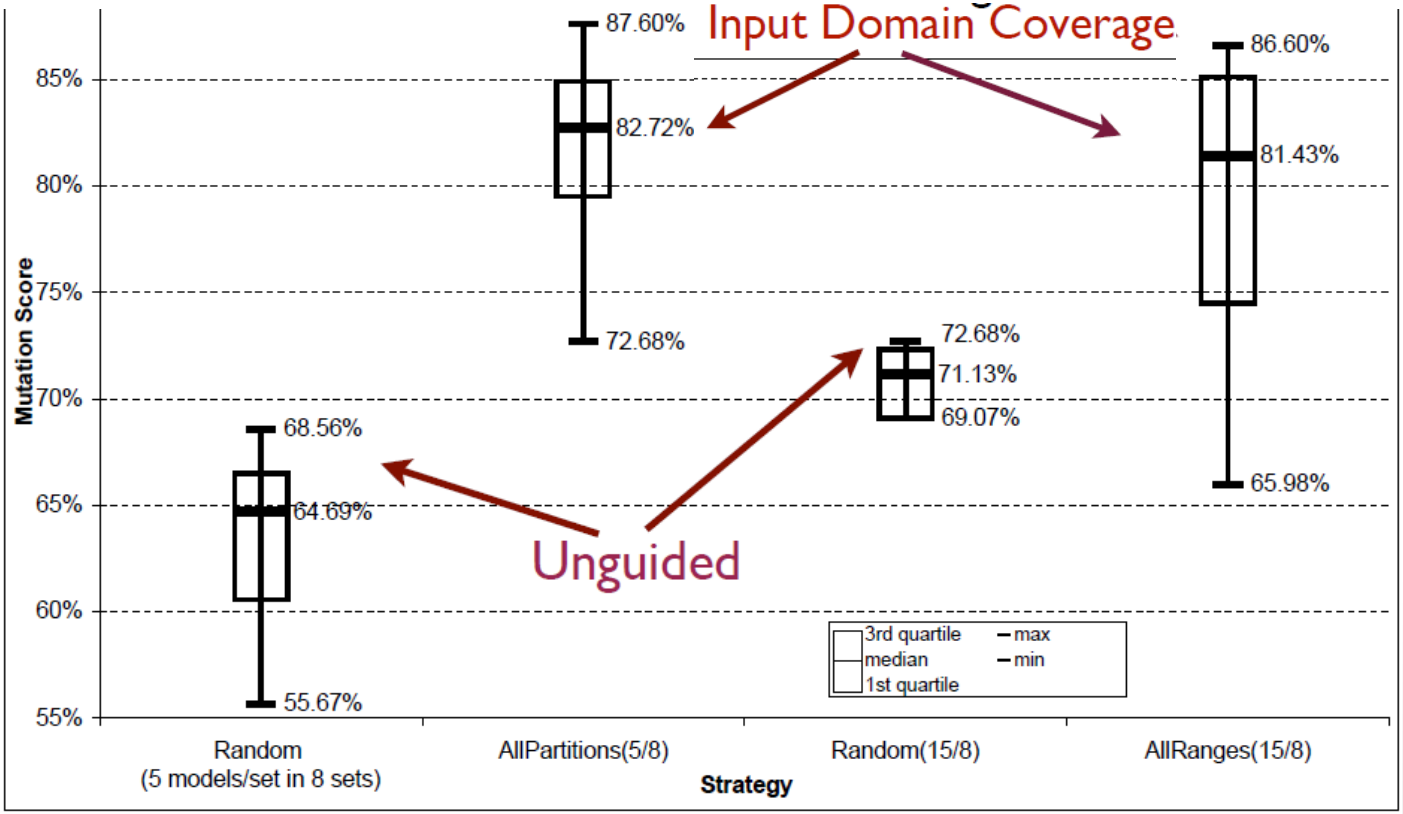
The story so far! Test Model Generation

- 2008: *How to generate models that satisfy knowledge from heterogeneous sources?*
- Published in: *Sen, Baudry, Mottu*. On Combining Multi-formalism Knowledge to Select Test Models, **ICST'08**



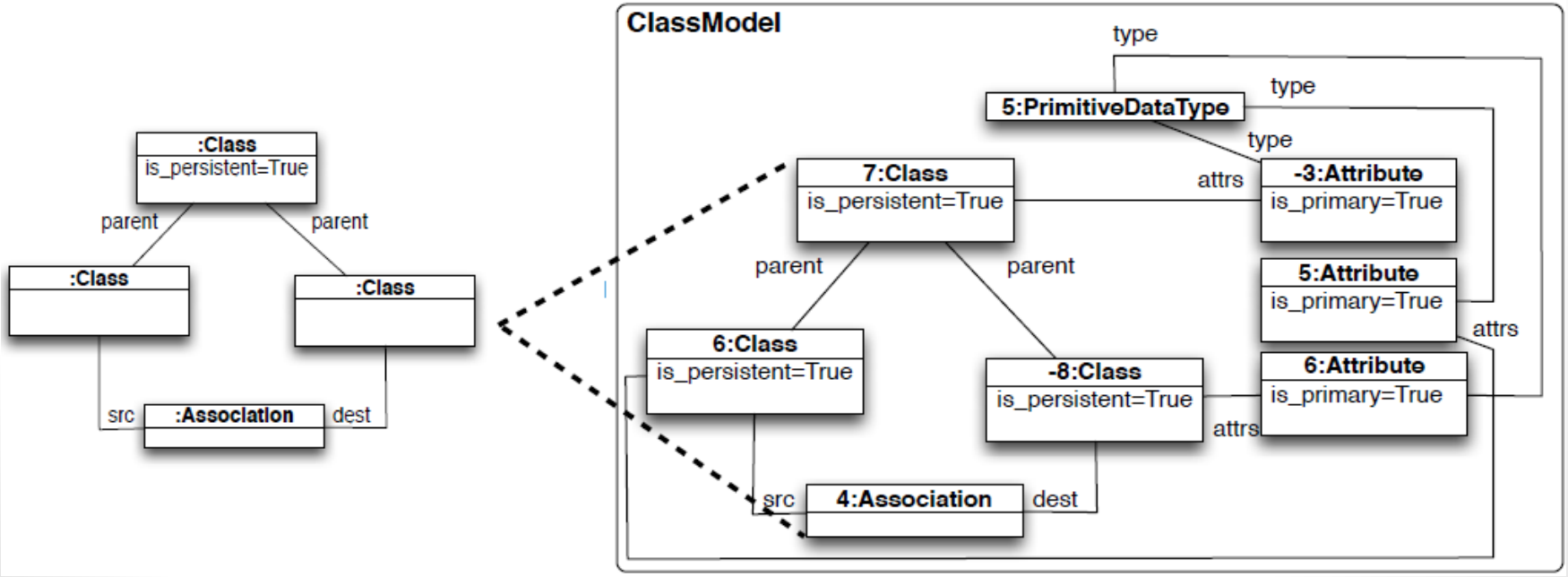
The story so far! Test Model Generation

- 2009: *How to test models satisfying coverage criteria and how to validate the quality of these test models?*
- Published in: *Sen, Baudry, Mottu*, Automatic Model Generation Strategies for Model Transformation Testing. **ICMT'09**
- 40 Test Models Covering Input Domain vs. 200 Unguided Models



The story so far! Test Model Generation

- 2012: How use "**partial knowledge**" by introducing a human-in-the-loop for test model generation?
- Published in: *Sen, Mottu et al.* Using Models of Partial Knowledge to Test Model Transformations, **ICMT'12**
- **Partial models** when completed **give 100% mutation score** just like human-made complete models with the same knowledge.



The story so far! Test Oracle

- 2008: Designed Oracle Function to test Model Transformations
- Published in : *Mottu et al.* Model Transformation Testing : oracle issue. **MoDeVVa'08** workshop at ICST'08

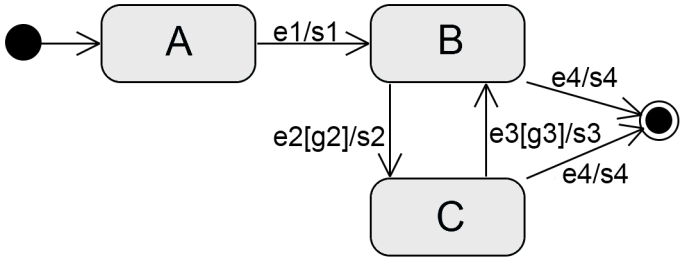
- An Oracle Function processes output models. It is parameterized with an Oracle Data and returns a verdict.
  - For instance an oracle function can use
    - a model comparison to compare output and expected models
    - or contracts

Problem: Expected Models are Difficult to Predict

- Can the tester write and use partial expected models in oracle?
- Tester predicts part of the expected model only
- Tester may not predict remaining part when:
  - the model is large or complex
  - test case does not consider part of the specification
  - the output model is polymorphic
  - the transformation refactors a model

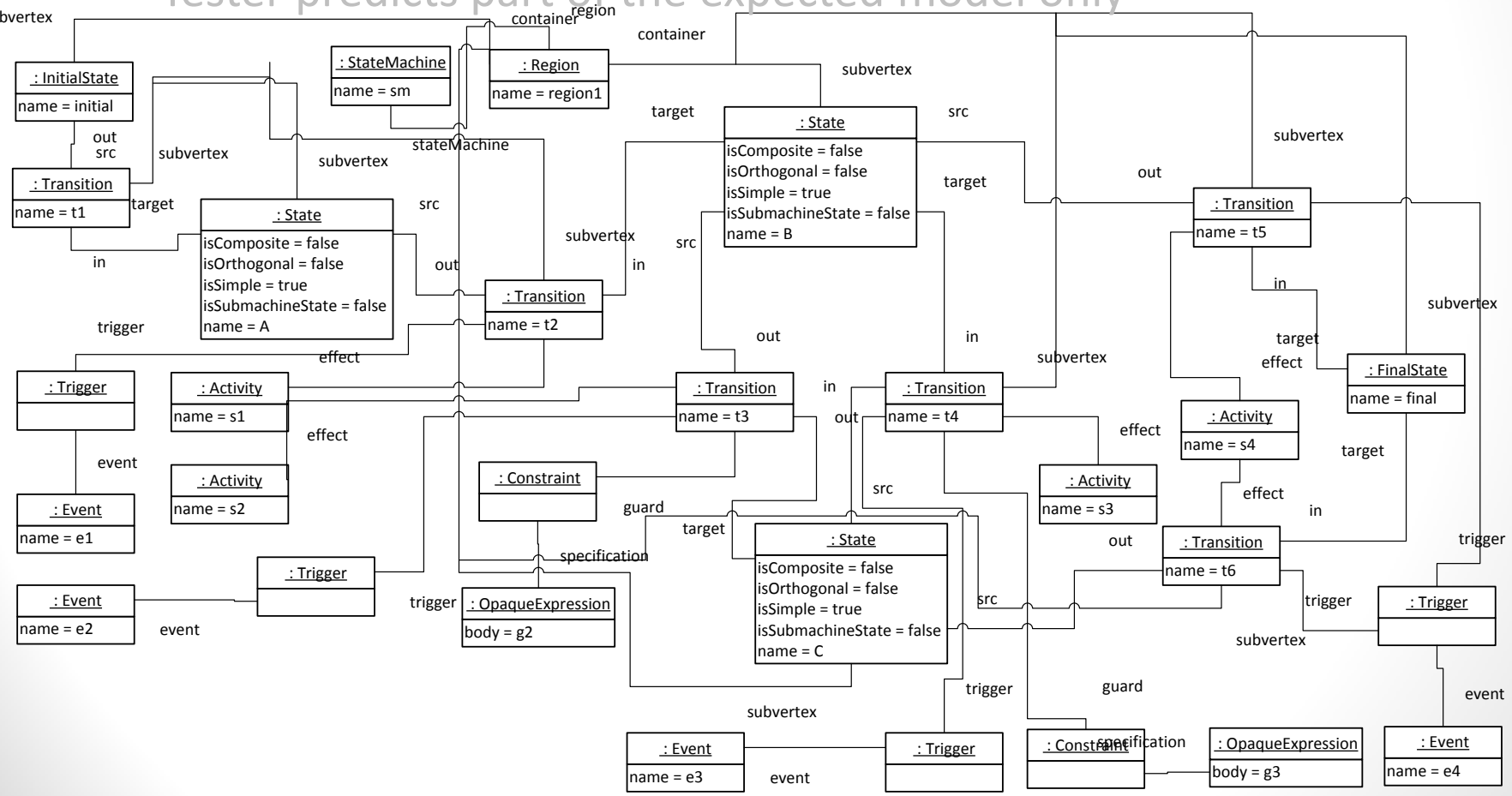
Problem: Expected Models are Difficult to Predict

- Can the tester write and use partial expected models in oracle?
- Tester predicts part of the expected model only
- Test may not predict remaining part when:
  - the model is large or complex
  - test case does not consider part of the specification
  - the output model is polymorphic
  - the transformation refactors a model



Problem:
Expected Models are Difficult to Predict

- Can the tester write and use partial expected models in oracle?
- Tester predicts part of the expected model only



Problem: Expected Models are Difficult to Predict

- Can the tester write and use partial expected models in oracle?
- Tester predicts part of the expected model only
- Test may not predict remaining part when:
  - the model is large or complex
  - test case does not consider part of the specification
  - the output model is polymorphic
  - the transformation refactors a model

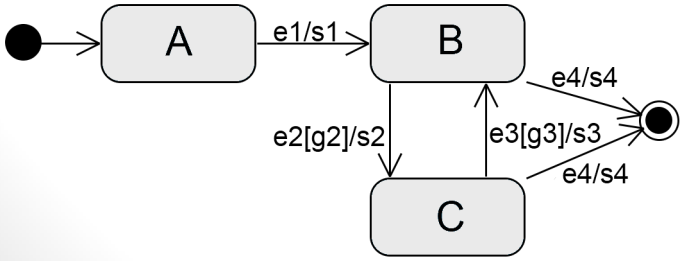
fsm2ffsm :

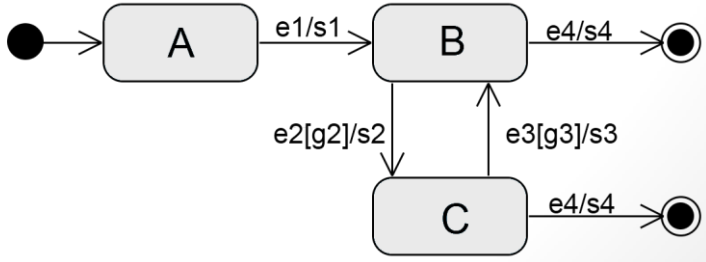
- *No more composite states*
- *Simple states kept*
- *Transition between simple states kept*
- *Final states kept*
- *etc.*

Problem: Expected Models are Difficult to Predict

- Can the tester write and use partial expected models in oracle?
- Tester predicts part of the expected model only
- Test may not predict remaining part when:
  - the model is large or complex
  - test case does not consider part of the specification
  - the output model is polymorphic
  - the transformation refactors a model

A polymorphic model may have different syntaxes but they are equivalent





Problem: Expected Models are Difficult to Predict

- Can the tester write and use partial expected models ?
- Tester predicts part of the expected model only
- Test may not predict remaining part when:
  - the model is large or complex
  - test case does not consider part of the specification
  - the output model is polymorphic
  - the transformation refactors a model

Filtered Model Comparison

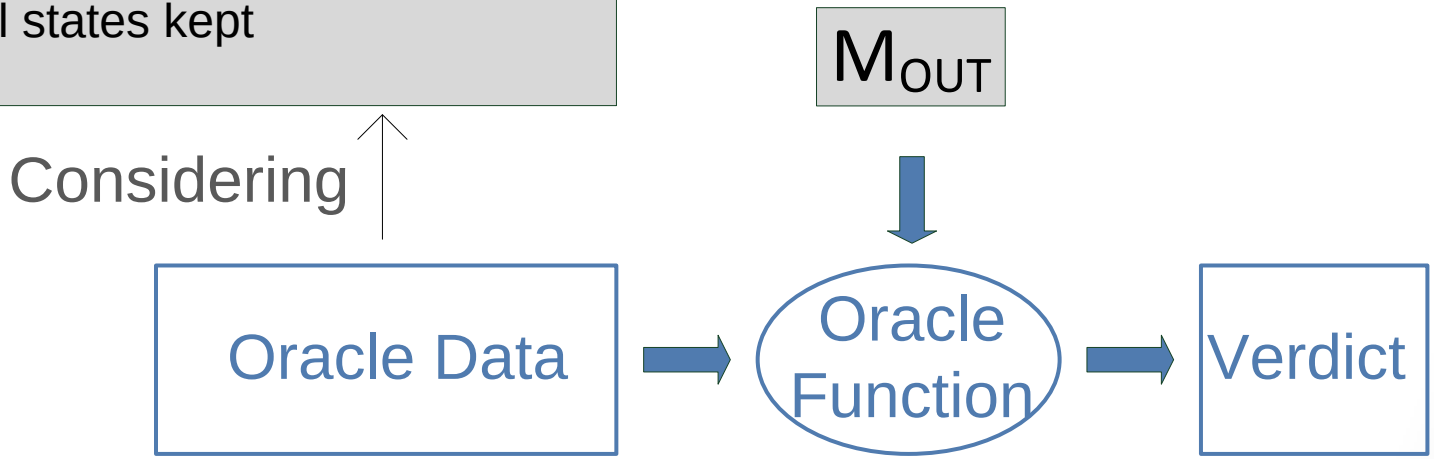
Filtered Model Comparison used in the Oracle Function.

The Oracle Data consists of only one model or one partial model.

The unpredicted part is not considered.

Oracle returns Partial Verdict

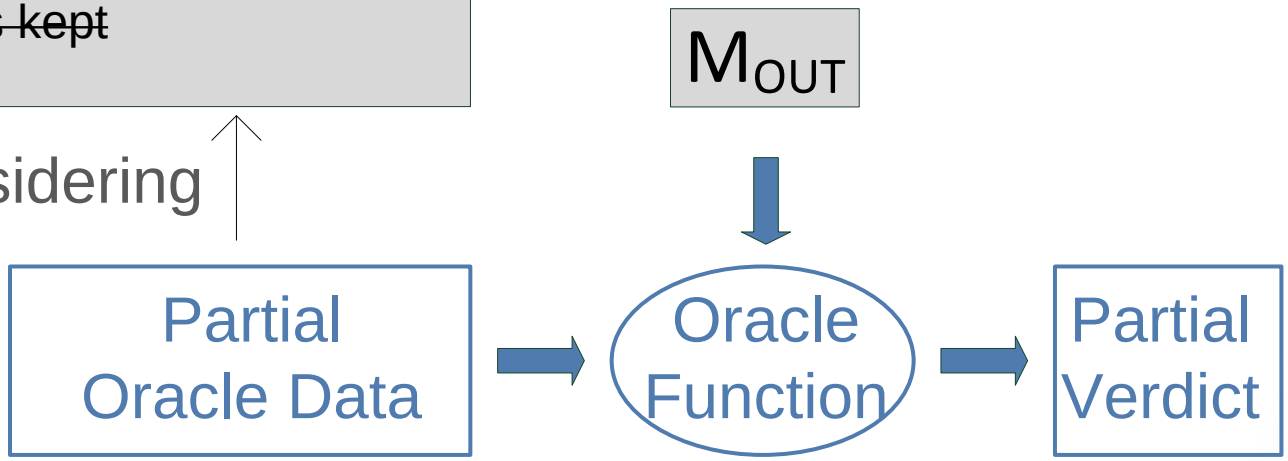
- No more composite states
- Simple states kept
- Transitions between simple states kept
- Final states kept
- etc.



Oracle returns Partial Verdict

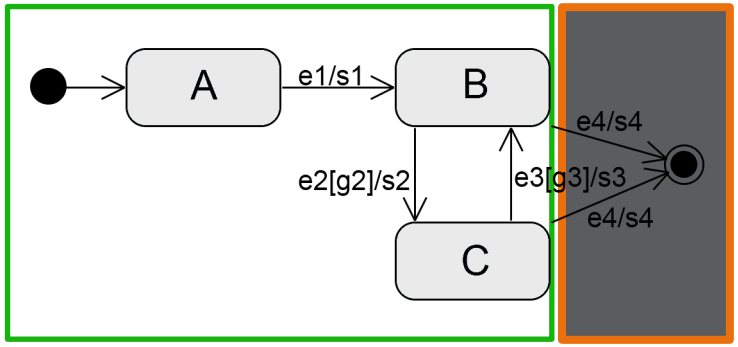
- No more composite states
- Simple states kept
- Transitions between simple states kept
- ~~Final states kept~~
- etc.

Considering



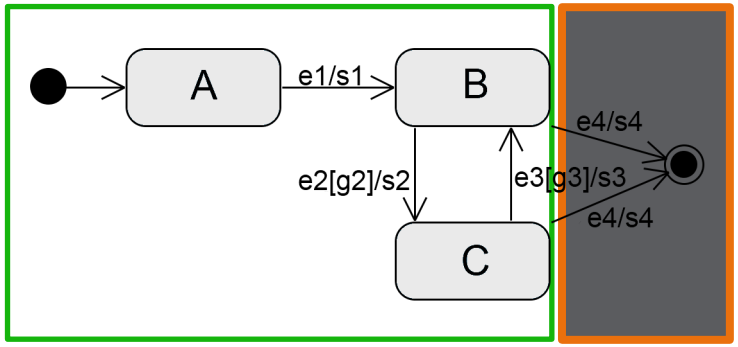
Predicted / Unpredicted Part of the Output Model

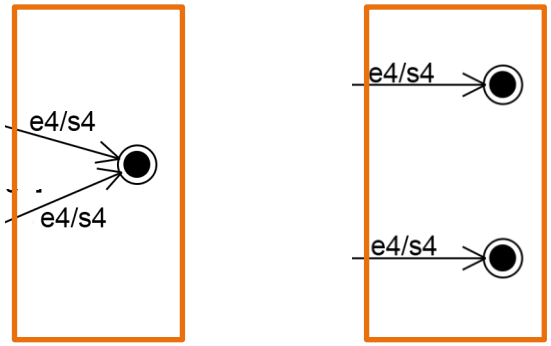
- The Oracle considers the predicted part
- The verdict is not concerned with the unpredicted part



Predicted / Unpredicted Part of the Output Model

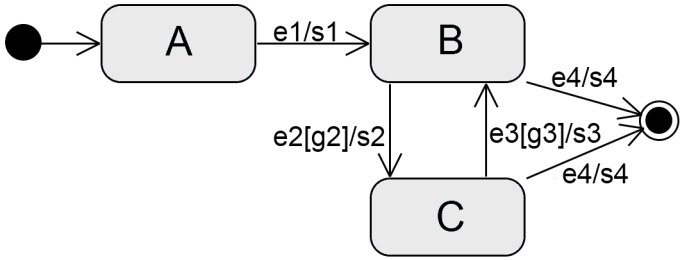
- The Oracle considers the predicted part
- The verdict is not concerned with the unpredicted part



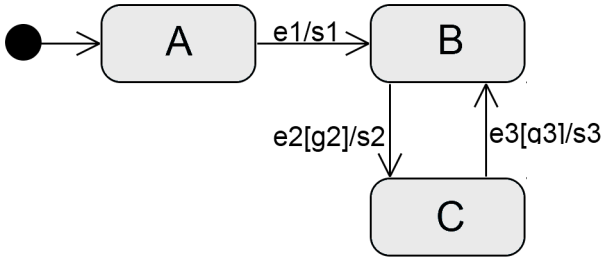


Oracle data is the partial expected model

- The expected model can be entirely writen :

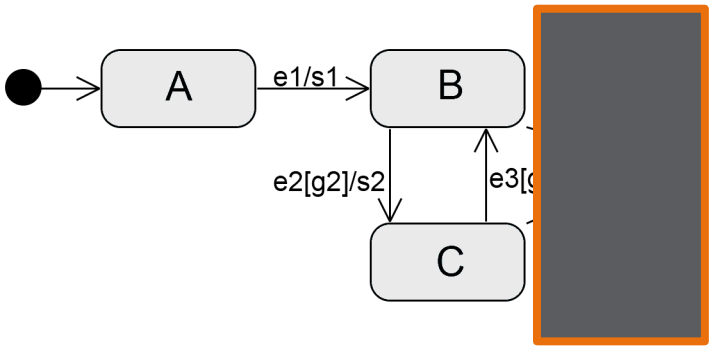


or partially writen :

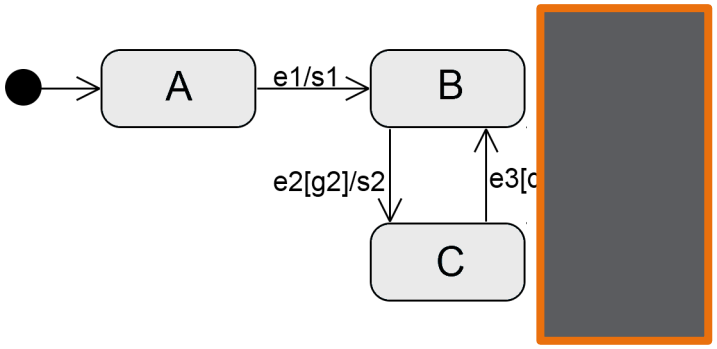


Oracle data is the partial expected model

- The expected model can be entirely writen :

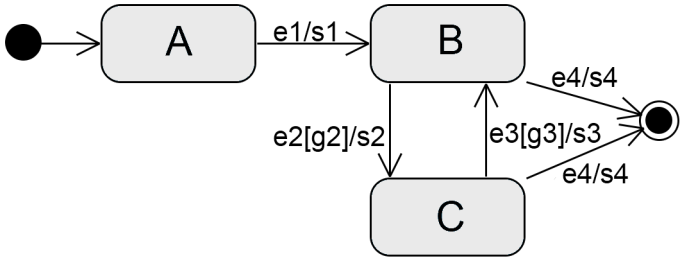


or partially writen :

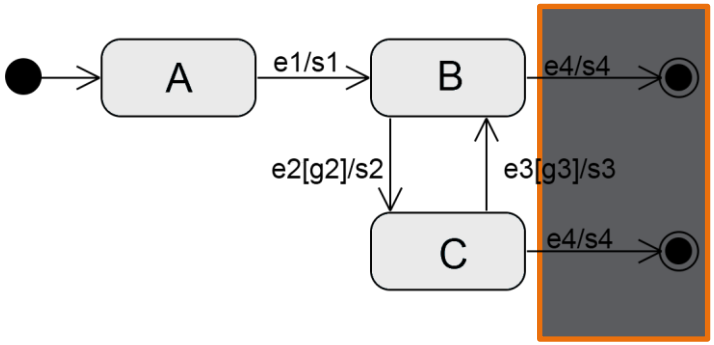


Oracle Function using Filtered Comparison

- Process



Output Model

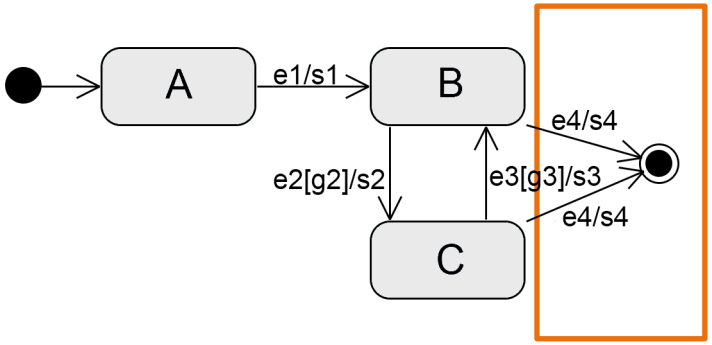


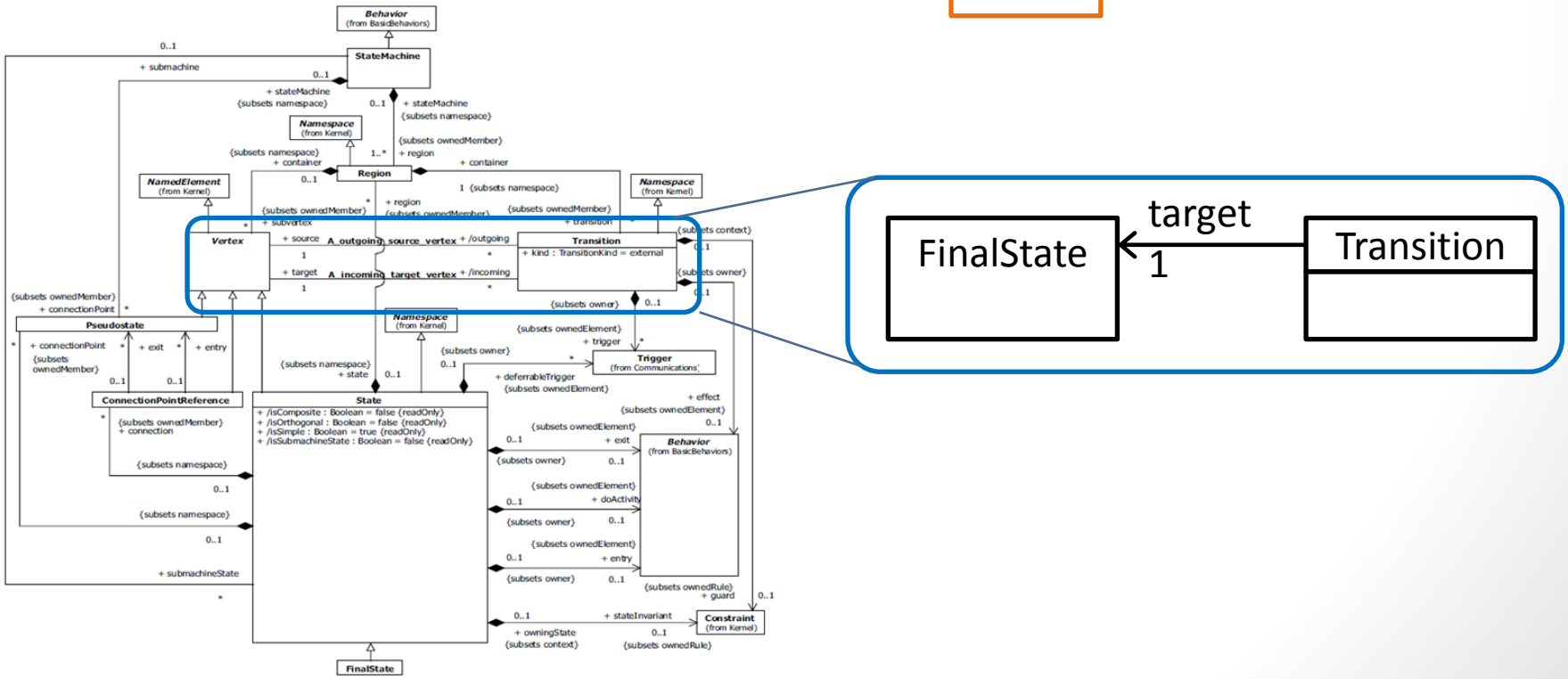
Partial Expected Model

1. Comparaison returns differences
2. Filter is parameterized with a pattern
3. The filter rejects difference about unpredicted part

Oracle Function using Filtered Comparison

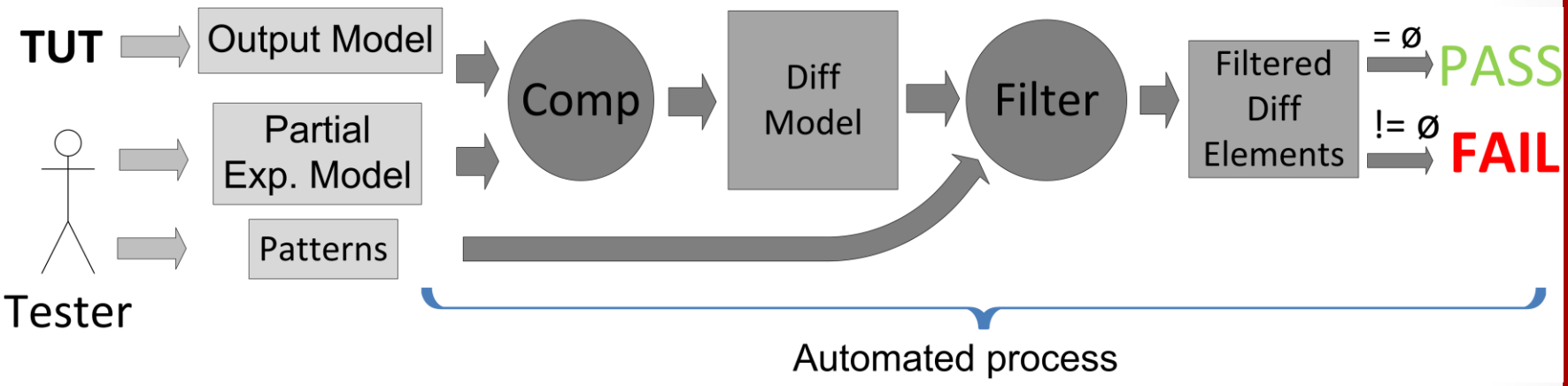
- Filtering with a **pattern** made of metamodel fragments





Oracle Function using Filtered Comparison

- Process



Prototype Implementation

- EMF Framework
- Model Comparison processed with EMF Compare
- Pattern matching processed with IncQuery

- Unpredicted Part defined with Ecore MM fragments.

- Available on : https://sites.google.com/site/partialverdictmt/

Experimentation

- Two model transformations :
  - fsm2ffsm: in Kermeta
  - UML to CSP: in ATL

- 30 + 64 test models
- 20 + 56 output models are polymorphic
  - Complete comparison requires to create **835 expected models**

  versus
  - Filtered comparison requires to create

    94 partial expected models and **8 patterns**
    - Patterns are shared by several test cases of a transformation
- Gain in tester effort is between 70% and 93%
- 4 bugs detected
- Available on : https://sites.google.com/site/partialverdictmt/

CONCLUSION

Benefits

- Allows the use of partial model as oracle data
- Partial verdict detects bugs
- Partial models are smaller and less numerous than complete models (especially polymorphic ones)
- Decreases the number of model comparison
- Helps to concentrate effort on unpredicted part
- Partial model and patterns constructed in the same way as test models

Limits

- Only part of the output model is controlled
- Control of the unpredicted part is postponed
- Describing the pattern is not obvious

Perspectives

- Evaluate the frequency and the size of the unpredicted part
- Empirical studies of the efficiency of these oracles
  - Based on Mutation Analysis
- Measure the completeness of a set of partial oracles

Partial Test Oracle in Model Transformation Testing

ICMT'13 – Budapest

Olivier Finot, Jean-Marie Mottu,

Gerson Sunyé, Christian Attiogbe

Story so far! Test Model Generation

- 2012: Can we extract effective testing knowledge via static analysis of a model transformation?
- Published in: *Mottu, Sen et al.* Static Analysis of Model Transformations for Effective Test Generation, **ISSRE'12**

- From Static Metamodel Footprinting, we identify which rule uses which MM concept. Then we transform this knowledge into testing knowledge.
- We expriment more than 90% mutation score.

Tediousness of Creating Models

- **Problems**
  - 1. Must conform to metamodel **MMi**
  - 2. Must satisfy **MMi invariants (OCL)**
  - 3. Must satisfy pre-conditions **pre(MT) on model transformation (OCL)**
  - 4. Must contain **test knowledge to find bugs**

Tediousness of Analysing Models

- **Problems**
  - 1. Must conform to metamodel **MMo**
  - 2. Must satisfy **MMo invariants (OCL)**
  - 3. Must satisfy post-conditions **post(MT) on model transformation (OCL)**
  - 4. Must control if output model **satisfies the specification depending on the test model**