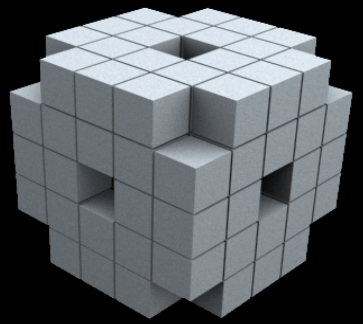


Tihamér Levendovszky

Toward Automated Verification of Model Transformations:
A Case Study of Analysis of Refactoring Business Process Models

Márk Asztalos, László Lengyel
Budapest University of Technology and Economics
Department of Automation and Applied Informatics

Tihamér Levendovszky
Vanderbilt University
Institute for Software Integrated Systems

Visual Modeling and Transformation System (VMTS)

Overview

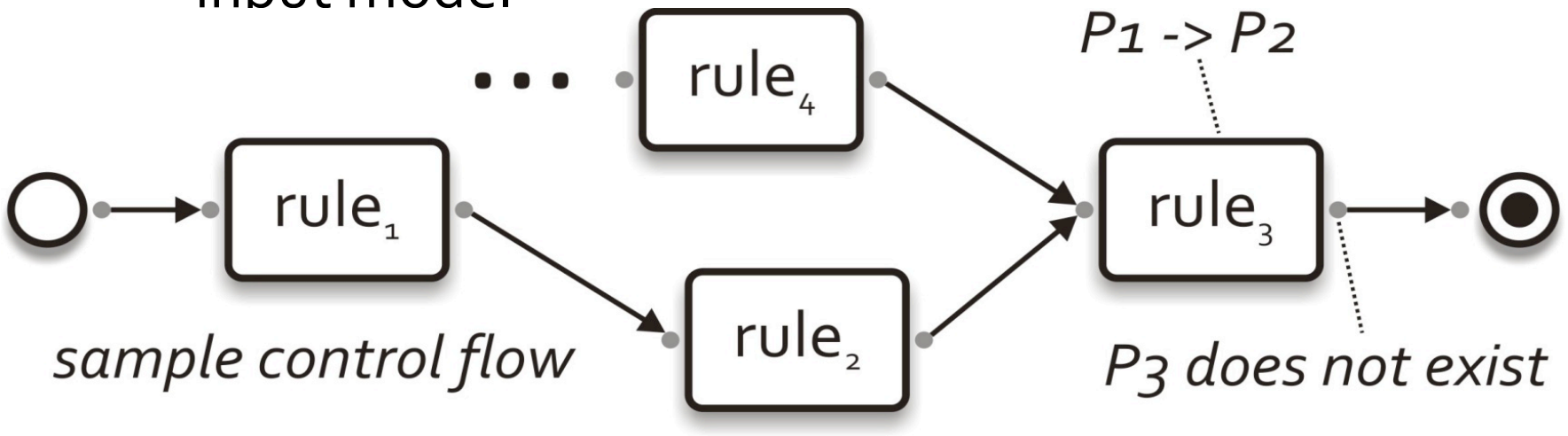
- Introduction to verification of model transformations
- Conceptual overview
  - A formalism to describe model transformations
  - Using a reasoning system to derive proofs of properties
- Case study: transforming business process models
- Conclusions

Introduction

- Model transformations in MPM environments
  - Transforming models between different paradigms (e.g. synchronization)
  - Generating models by composing existing ones even of different paradigms
- Verification of transformations
  - Analyzing the properties of the models under transformation
  - Analyzing the properties of the transformation (functional / non-functional)
- *Offline* analysis
  - Only the very definition of the transformation itself and the metamodels of the input and output languages are taken into account
  - Advantage: the results hold for every input model

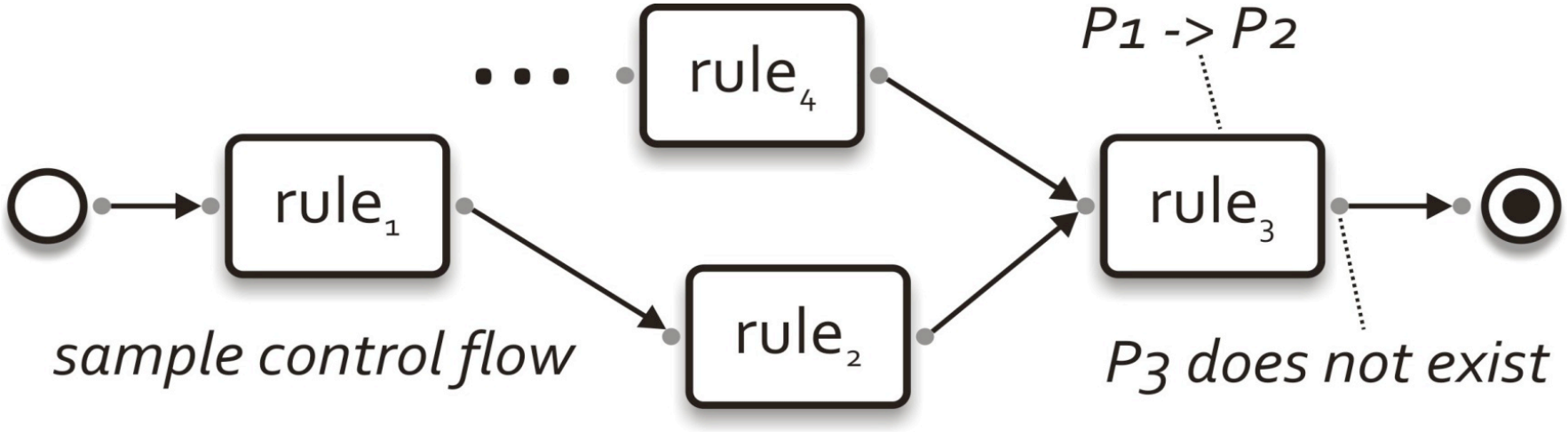
Concept overview

- A model transformation consists of a *control flow* and its *rules*
- *Predicates* are attached to some points of the transformation
  - E.g. in the sample control flow:
    - *rule3* transforms each instance of pattern *P1* to an instance of patter *P2*
      - the rule modifies the input model
      - *P1* and *P2* are model patterns
    - After *rule3* there are no instances of pattern *P3* in the input model



Concept overview – assertions

- A formalism has been defined to describe transformations
  - Predicates are called *assertions*
  - Assertions state something about the models under transformation or the transformation itself
  - Assertions are attached to concrete positions in the transformation control flow
  - Assertions are always true for the transformation and are independent of the concrete input models
  - Assertions reference model patterns



Concept overview – reasoning

- Given an initial set of assertions that are true, we can derive new assertions that are also true for the transformation
  - Deduction rules should be defined
  - A reasoning system should be used that can generate the proof of new assertions using the deduction rules
- Verification of model transformations:
  - For a concrete transformation, a initial assertion set should be generated
  - An automated reasoning system can work on assertions, and derive new assertions
  - Properties to be verified should be expressed in the form of assertions

Realization

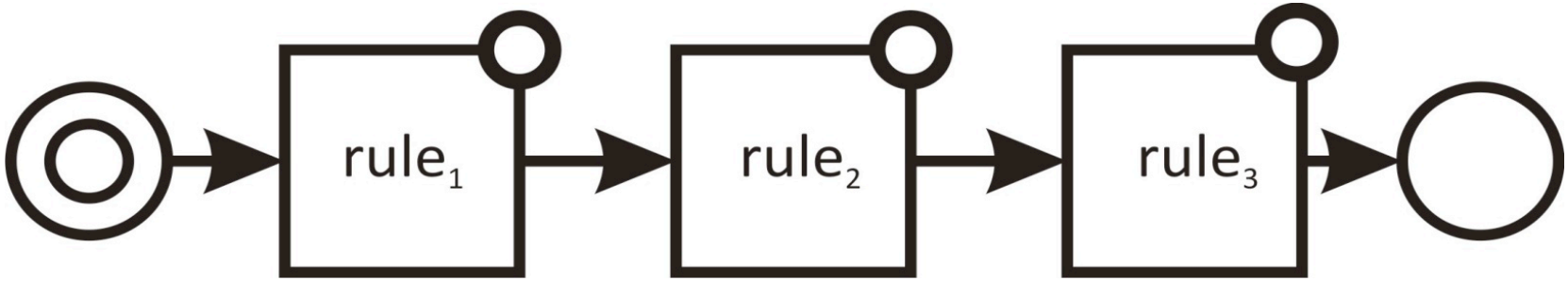
- Visual Modeling and Transformation System (VMTS) is a modeling tool
- For the reasoning system, SWI-Prolog is used
  - Assertions, patterns and deduction rules are compiled to Prolog predicates
- An initial assertion set is generated from the definition of the model transformations in VMTS
- The reasoning system can decide if a new assertion can be derived from the initial assertion set or not

(Semi-)automation

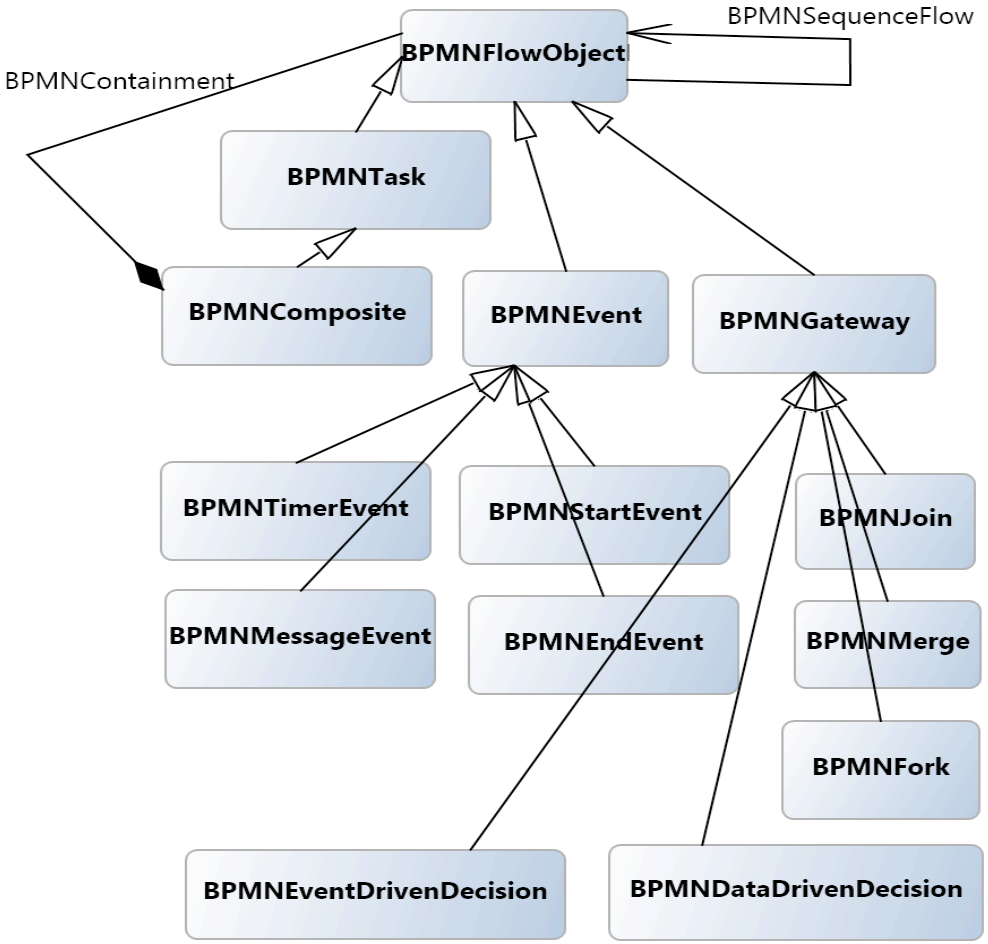
- A transformation description including an initial assertion set can be *automatically* generated after the definition of the model transformation
  - This initial assertion set can be extended manually by domain experts (e.g. the developer of the transformation) with additional assertions
  - The extended assertion set can be used by the reasoning system

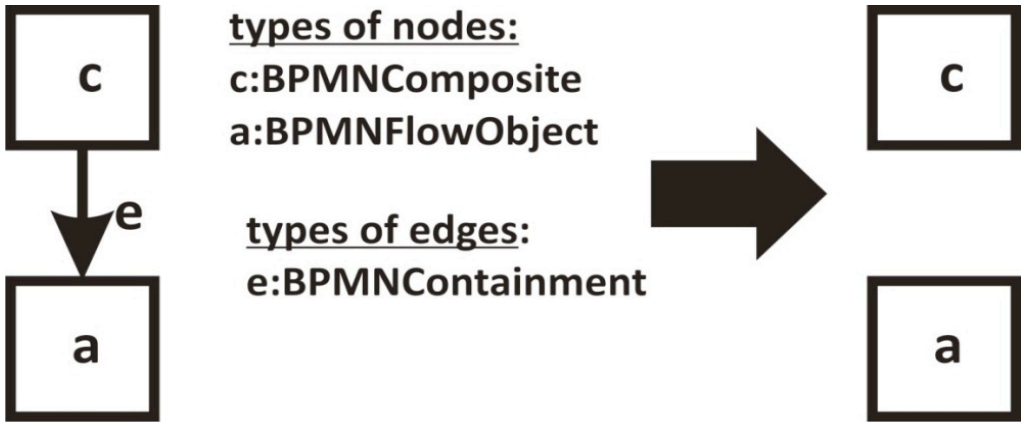
Case study

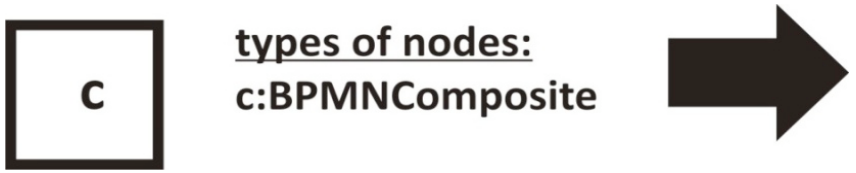
- Flattening BPMN models - transforming hierarchical BPMN models into flattened models (in-place transformation):
  - For each composite node: (rule1 - exhaustive)
    - All incoming edges are redirected to the first contained node in the composite node
    - The source of all outgoing edges from a composite node will be the last contained node in the composite node
  - All containment edges will be deleted (rule2 - exhaustive)
  - All container nodes will be deleted (rule3 - exhaustive)



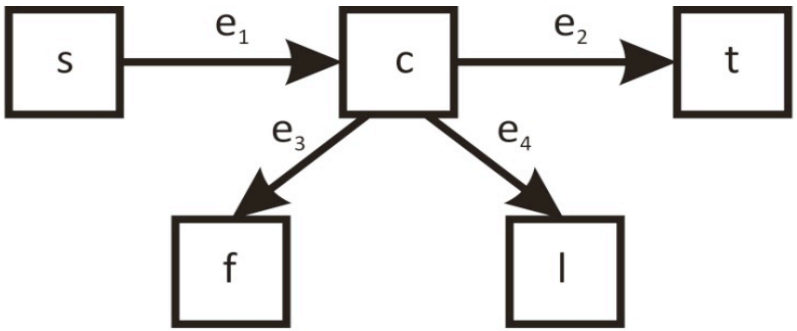
BPMN metamodel, rule2, rule3

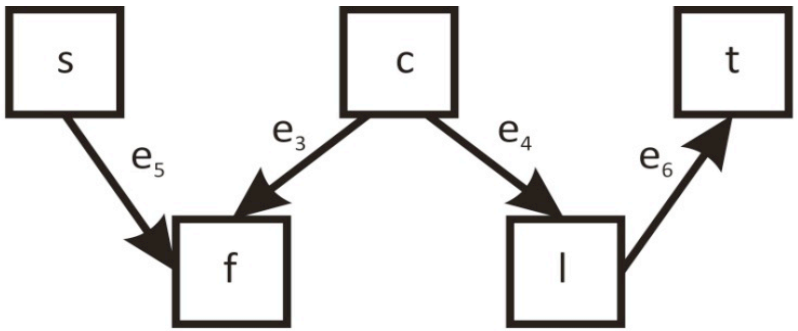






rule1





attribute constraints:
f.IsFirst = true
f.IsLast = false
l.IsFirst = false
l.IsLast = true

types of nodes:
s: BPMNFlowObject
c:BPMNComposite
t:BPMNFlowObject
f:BPMNFlowObject
l:BPMNFlowObject

types of edges:
e_1:BPMNSequenceFlow
e_2:BPMNSequenceFlow
e_3:BPMNContainment
e_4:BPMNContainment

attribute constraints:
f.IsFirst = false
f.IsLast = false
l.IsFirst = false
l.IsLast = false

types of edges:
e_5:BPMNSequenceFlow
e_6:BPMNSequenceFlow

Verification of the transformation

- Simple properties have been selected to be verified in this presentation
  - *At the end of the transformation there are no containment edges in the model*
  - *At the end of the transformation there are no composite elements*
- Deduction rules are used for the reasoning
  - Deduction rules have been separately defined
  - Deduction rules are independent of the concrete transformation
- The reasoning is performed automatically

Sample deduction rules (informally)

- *Deduction rule no. 1*
  - If a rule matches only a certain type of element and deletes it, and the rule is applied exhaustively, then, after the application of the rule, there will be no elements of that type in the model
  - Applying this deduction rule:
    - After rule2, there are no containment edges
    - After rule3, there are no composite nodes
- *Deduction rule no 2.*
  - If before a rule there are no elements of a certain type and the rule is not concerned with elements of that type, then, after the rule there will be no elements of that type either
  - Applying this deduction rule:
    - Before and after rule3 there are no containment edges

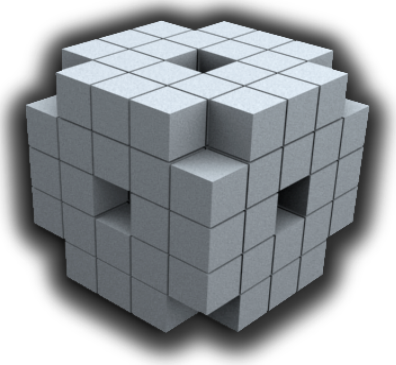
Conclusions

- In the paper, we have demonstrated the applicability of the concept of automated verification of model transformations

- Advantages of the approach:
  - Our approach supports offline analysis
  - Knowledge of domain experts can be exploited

Thank you for your attention!

Thank you for your attention!



http://vmts.aut.bme.hu