



Budapest University of Technology and Economics

Visual Modeling and Transformation System

László Lengyel

<http://vmts.aut.bme.hu>

vmts@aut.bme.hu

Research /1: Validated Model Transformation

➤ Motivation

- At the implementation level, system validation can be achieved by testing. There is no real possibility that the testing covers all the possible cases.
 - In case of model transformation environments, it is not enough to validate that the transformation engine itself works as it is expected. The transformation specification should also be validated.
- A precondition assigned to a transformation rule is a boolean expression that must be true at the moment when the transformation rule is fired. A postcondition assigned to a transformation rule is a boolean expression that must be true after the completion of a transformation rule.
 - An OCL expression in LHS is a precondition to the transformation rule, and an OCL expression in RHS is a postcondition to the transformation rule.
 - Successful execution of the rule guarantees that the output model fulfills the conditions required by high-level constructs.

Research /2: Aspect-Oriented Constraint Management

- Motivation
 - Transformation consists of several rules, many times not only a transformation rule but a whole transformation is required to validate, preserve or guarantee a certain property.
 - The same constraint appears numerous times in the transformation → crosscuts the transformation.
- Aspect-oriented constraint management
 - Aspect-oriented constraints
 - Constraint aspects
 - Weaver algorithms
- Results
 - Consistent constraint management
 - Reusable constraints and transformation rules
 - Weaving algorithms facilitates to require from not only individual rules, but from whole transformations to validate, preserve or guaranty certain properties.

Research /3

- Animation and Simulation in VMTS
 - DSLs
 - Transformation debugging
 - UI Programmability / Layout Animation
- Supporting Domain-Specific Design Patterns
 - The goal is to make model transformation-based development easier, more efficient and rapid.
 - In graph rewriting-based model transformation, there are several recurring problems that should be solved again and again in the context of different transformations or different environments → Design Patterns.
 - A pattern is a reusable entity, which describes a frequent design or implementation problem, and gives a general but customizable solution to it.
- Supporting Round-trip Engineering
 - Model – Source code and Source code – Model Synchronization
 - Traceability
 - Sophisticated diff mechanisms

Research /4: Applying Multi-Paradigm Modeling to Multi-Platform Mobile Development

- Static UI
 - Different enough to be different paradigm
 - Related enough to synchronize them
 - MPM is the best solution: must be split into paradigms instead of one huge model
- Behavior
- Data services and data binding
- Communication: another domain
- Multi-Paradigm Modeling
 - Within the UI, data, behavior
 - Extensible with the other services

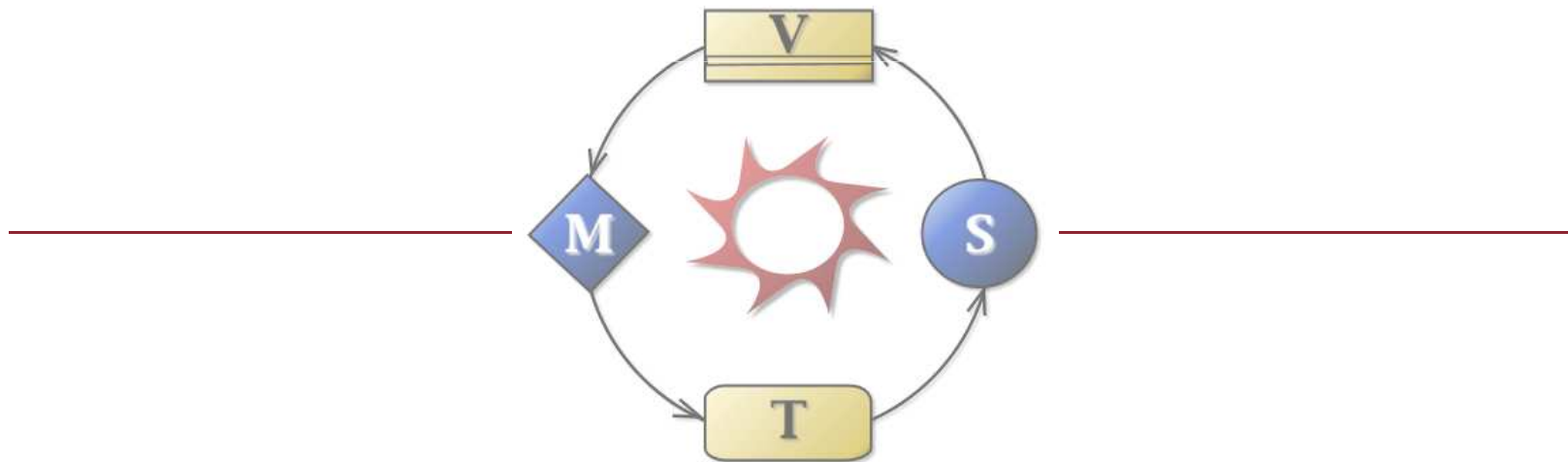
Topic selection/specific problems

- Analysis and design of complex systems (MPM)
- Modeling of transformations (transformations as first-class models)
- Model testing
- Metamodeling and model transformation
 - Metamodel and model evolution
 - Testing model transformations
 - Debugging model transformations

Criteria for success (from my point of view)

- Connections
- CAMPaM related references
- Ideas how to analyze, design and build complex systems
 - Real model-based development
 - Model transformation-based round-trip engineering
- To understand how others on the same are do research

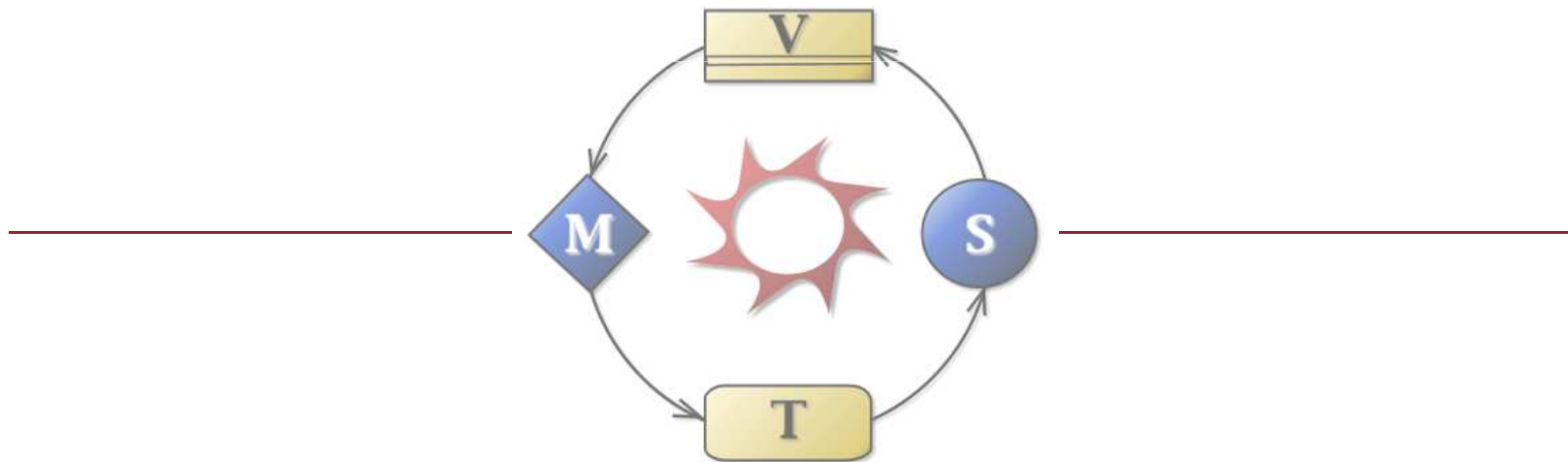
Questions?



<http://vmts.aut.bme.hu>

vmts@aut.bme.hu

Thank you for your attention!



<http://vmts.aut.bme.hu>

vmts@aut.bme.hu