First Workshop on the

Analysis of Model Transformations (AMT'12)

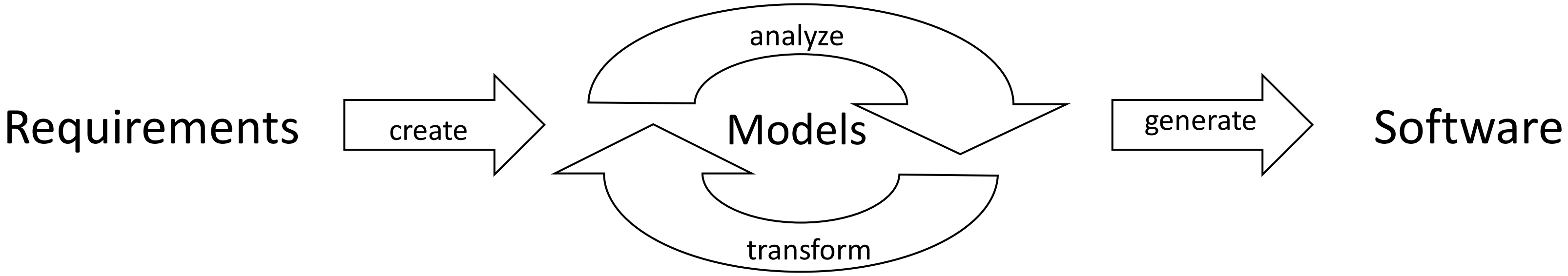
MODELS 2012
innsbruck

October 2, 2012

Organizers:

Juergen Dingel, Levi Lucio, Hans Vangheluwe, Daniel Varro

MT in MDE



> "Model transformation [is] the **heart and soul** of model-driven software development" [SK03]

> "Software has the rare property that it allows us to directly evolve models into full-fledged implementations ***without changing the engineering medium, tools, or methods***." [Sel04]

[SK03] S. Sendall, W. Kozaczynski. "Model Transformation: The Heart and Soul of Model-Driven Software Development", IEEE Software, Sept. 2003

[Sel04] B. Selic. "ENSIETA Summer School on Model-Driven Engineering of Embedded Systems', Sept. 2004

*"We need to evolve a **systematic theoretical understanding** of the various key capabilities that are at the core of MBSE, such as the principles of modeling language design, **model transformations**, code generation, automated verification, and so on".* [Sel08]

*"Central to all model-based design is an **effective theory of model transformations**. …
Model transformation **must preserve essential properties**".* [HS07]

*"More research is needed on **analyzing model transformations**".* [FR07]

[Sel08] B. Selic. "Personal reflections on automation, programming culture, and model-based software engineering". *J of Autom. Softw. Eng*, 2008

[HS07] T. Henzinger, J. Sifakis. "The Discipline of Embedded Systems Design", *IEEE Computer*, Oct 2007

[FR07] R. France, B. Rumpe. Model-driven Development of Complex Software: A Research Roadmap" Future of Software Engineering (FOSE). 2007.

Open Issues Include

- **Make design, implementation and validation more systematic**
- **Support for *all* life-cycle activities**
  - requirements/specification, reuse, evolution, maintenance
- **AMT in the context of**
  - different MT languages/paradigms
    - Impact? Similarities & Differences? Help or Hindrance? Domain-specifity?
  - analysis criteria
    - Which are most relevant? How to express them?
  - large body of existing work
    - Similarities & Differences? Relevance of program/compiler verification, abstraction interpretation, etc?
- **Industrial case studies, more mature tools, benchmarks**

A Brief Look at the AMT Landscape

- **Surveyed 64 papers [GCD12]**
- **Kinds of approaches:**
  - Formal methods (FM):
    1. Transformation-independent & input-independent
    2. Transformation-dependent & input-independent
    3. Transformation-dependent & input-dependent
  - Testing
  - Instrumentation
- **Kinds of formalisms:**
  - E.g., Graph rewrite systems, Petri Nets, Maude, Alloy, Predicate Logic

[GCD12] G. Selim, J. Cordy, and J. Dingel. "Analysis of Model Transformations". Technical Report 2012-592, Queen's University, Aug 2012. 58 pp.

A Brief Look at the AMT Landscape

- **Kinds of MT analysis criteria:**
  - Syntactic
    - Conformance: “Do output models conform to target metamodel?”
    - Completeness: “Does MT cover all possible input models?”
  - Operational
    - E.g., termination, confluence, performance, absence of dead code
  - Semantic
    - E.g., preservation/establishment of properties (from syntactic to semantic)
- **Kinds of techniques:**
  - E.g., theorem proving, critical pair analysis, state space exploration, etc

[GCD12] G. Selim, J. Cordy, and J. Dingel. “Analysis of Model Transformations”. Technical Report 2012-592, Queen’s Univ., Aug 2012. 58 pp.

FM: Trans-indep & Input-indep

- 16 of 64 papers
- Dominated by approaches based on Graph Rewriting

Property	Techniques
confluence	Critical pair analysis
termination	Measurement function, layering

- **Exceptions:**
  - Predicate Logic formalization: consistency, semantic properties by theorem proving [Stenzel+, 11]
  - Petri nets: termination by Petri net analysis [Varro+, 06]
  - Custom MT language: termination [Barroca+, 11]

FM: Trans-dep & Input-indep

- 4 of 64 papers
- All based on Graph Rewriting

Formalism	Property	Technique	References
Graph Rewrite System (TGG)	Preservation of safety properties	Theorem proving	[Giese et al, 2006]
	Absence of forbidden patterns	Static analysis	[Becker et al, 2006]
Graph Rewrite, Category Theory, Predicate Logic	Assertions	Proof system	[Asztalos et al, 2010]
Graph Rewrite	Input/output patterns	State space exploration	[Lucio et al, 2010]

FM: Trans-dep & Input-dep

- 14 of 64 papers
- More mixed

Formalism/language	Property/purpose	Technique	Reference
Graph Rewrite System	Satisfaction of mapping specification	Instrumentation, correspondence rules	[Narayan+, 08]
Alloy	Simulation	Constraint solving	[Anastasakis+, 07]
Maude	Simulation, invariant & LTL checking	State space exploration	[Boronat+, 09]
Graph Rewrite System	Temporal logic checking	State space exploration	[Rensink, 03]
Java	Assertion checking	Software model checking (JPF)	[Ab Rahim+, 10]

Testing & Instrumentation

- **Testing:**
  - 26 of 64 papers
  - 3 phases:
    - Test case generation
    - Test suite assessment
    - Oracle construction
- **Instrumentation:**
  - Only 2 of 64 papers
  - Used for fault localization

Observations

- **Fair bit of work on testing (26/64), little on instrumentation (2/64)**
- **Graph rewriting seems to have a bit of an edge**
- **Obstacles**
  - Variety of MT languages
  - Lack of suitable theories
- **Relatively unexplored seem to be:**
  - Lots of analysis techniques
    - E.g., instrumentation, profiling, abstract interpretation, static analysis
  - Some lifecycle activities
    - E.g., requirements, reuse, maintenance, evolution, debugging
  - Some properties
    - E.g., performance

AMT’12: Objectives and Scope

Ideas for analysis of MTs to support all lifecycle activities of MTs including development, quality assurance, maintenance and evolution by facilitating, e.g.,

- Detection of typing errors, anti-patterns, dead code, slices, likely invariants, performance bottlenecks
- Establishment of correctness- or performance-related properties
- Test suite evaluation
- Code completion and generation
- Metamodel and transformation evolution
- Impact analysis, refactoring

Encourage input from other subdisciplines

AMT’12: Topics

- Testing & test case generation
- Formal specification and verification
- Abstract interpretation
- Static analysis (control-flow, data-flow, slicing)
- Dynamic analysis (runtime monitoring)
- Metrics
- Impact analysis
- Certification and incremental re-validation for model transformations
- Use of higher-order transformations for analysis
- Case studies and tools

AMT'12: Program

Welcome

Session 1: Intents of model transformations and model transformation chains

9:30 "**Towards a Model Transformation Intent Catalog**"
Moussa Amrani, Juergen Dingel, Leen Lambers, Levi Lucio, Rick Salay,
Gehan Selim, Eugene Syriani and Manuel Wimmer

10:00 "**Chaining Model Transformations**"
Anne Etien, Vincent Aranega, Xavier Blanc and Richard Paige

Coffee break (10:30 – 11:00)

Session 2: Model transformation testing and debugging

11:00 "**Model Transformation Testing: an Experience Report**"
Alessandro Tiso, Gianna Reggio and Maurizio Leotta

11:30 "**A Survey on Model Transformation Testing**"
Gehan Selim, James R. Cordy and Juergen Dingel

12:00 "**Towards Tracking Guilty Transformation Rules**"
Loli Burgueno, Manuel Wimmer and Antonio Vallecillo

Lunch (12:30 – 14:00)

AMT'12: Program (Cont'd)

Lunch (12:30 – 14:00)

Session 3: Quality of model transformations

14:00 "**The MDELab Tool Framework for the Development of Correct Model Transformations with Triple Graph Grammars**"
Stephan Hildebrandt, Leen Lambers and Holger Giese

14:30 "**Developing and Visualizing Live Model Queries**"
Zoltan Ujhelyi, Tamas Szabo, Istvan Rath and Daniel Varro

15:00 "**Towards Refactoring of Rule-based, In-place Model Transformation Systems**"
Gabriele Taentzer, Claudia Ermel, Thorsten Arendt and Reiko Heckel

Coffee break (15:30 – 16:00)

16:00 - 17:30 **Group discussions**

Topics for Discussion

- Research topics and problems?
- Relevant related areas/topics:
  - Source code transformation
  - Program specification & verification
  - Compiler verification
  - Typing
  - Program analysis (static, dynamic, monitoring)
  - Program Testing
  - Certification

  How relevant are they? Similarities & differences? How to reach them?
- Scope and future of AMT?
- Anything else?

Enjoy AMT’12!

MODELS 2012
innsbruck