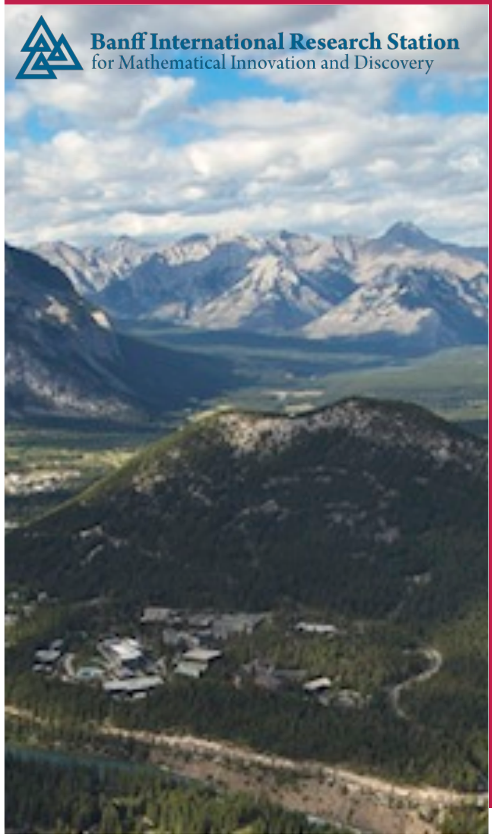
HPI Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam



"The Implications of Optimality Results for Incremental Model Synchronization for TGGs"

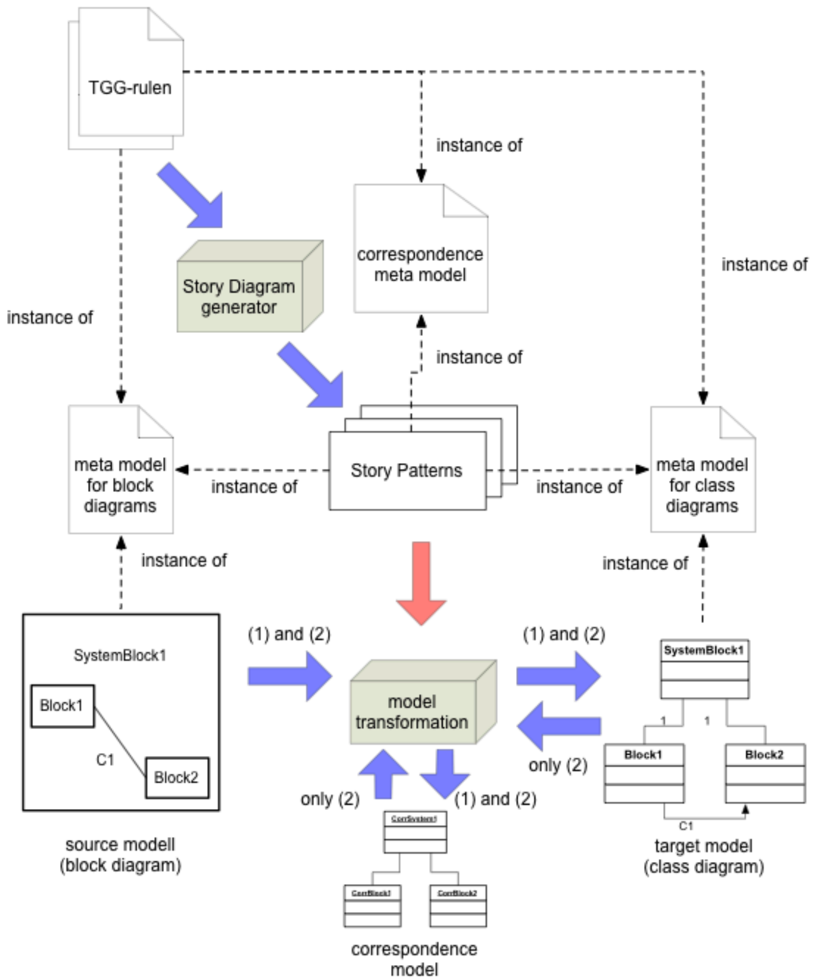
Bi-directional transformations (BX) –
Theory and Applications Across Disciplines (13w5115)
December 1 - 6, 2013

Holger Giese, Stephan Hildebrandt
System Analysis & Modeling Group,
Hasso Plattner Institute for Software Systems Engineering
University of Potsdam, Germany
holger.giese@hpi.uni-potsdam.de

Outline

1. **Introduction**
2. **Optimality Result**
3. **Open Issues & Implications**

1. Introduction: Triple Graph Grammars



Problem:

- Describe full model transformations where source and target can have different meta-models

Solution:

- TGGs: describe **relation** between source and target by means of a grammar that generates all valid pairs (here even triples incl. correspondence)
- Derive operational rules in form of Story Pattern for
  - forward direction,
  - backward direction, and
  - consistency

1. Introduction:
Theoretical Results on TGGs

- MDELab is a toolset for our Triple Graph Grammar dialect (and Story Diagrams) covering also testing and analysis [1]

Consistency, Completeness of TGG:

- Well-formedness rules which guarantee consistency and completeness for our TGG dialect [2] ➔ apply to model synchronization as well

Testing of TGGs:

- Automated test generation for our TGG dialect offering a complete coverage of the TGG specifications [3,4] ➔ apply to model synchronization as well

Verification of TGGs (and related GTS):

- Automatic checking behavior preservation for our TGG dialect [5]
  ➔ apply to model synchronization as well
- Automatic checking of structural constraints for transformation rules [6]
  ➔ apply to TGGs and model synchronization as well

1. Introduction:
Practical Results on TGGs

Applications:

- Industrial Case Study on the Integration of SysML and AUTOSAR with Triple Graph Grammars via model synchronization [7,8]
- Application of model synchronization for Runtime Models [9]

Efficiency:

- Improved performance for Story Diagrams using runtime information [12]
- **Optimal performance for incremental model synchronization for our TGG dialect for practically relevant models [13,14] based on the results for Story Diagrams [12]**

Why is it relevant, what is the result, and why does it work?

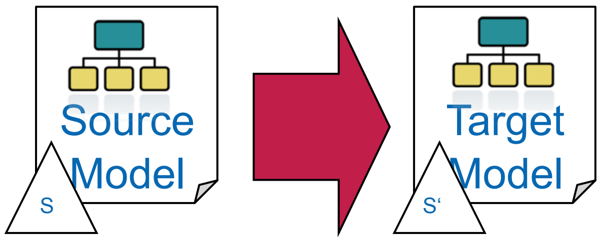
2. Optimality Result: Motivation

Why does performance matter at all?

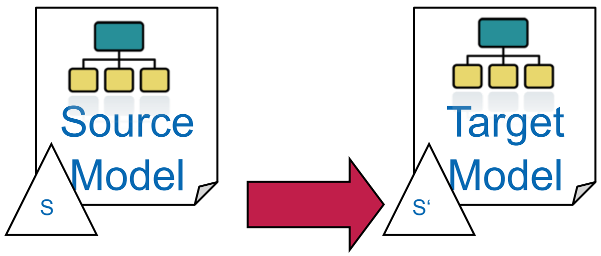
- Reducing the effort also means to minimize the unnecessary changes; **optimal means no unnecessary changes at all!**
- We may have multiple model synchronizations linked together additional steps such as consistency checks, and/or rollbacks and thus the overall performance matters (model management)
- Short execution times make conflicts very unlikely (if models are handled by the same tool or a shared repository)
- Besides case tools we apply model synchronizations also for runtime models where the overhead should be minimal

2. Optimality Result:
Synchronization Problem

- Model Synchronization

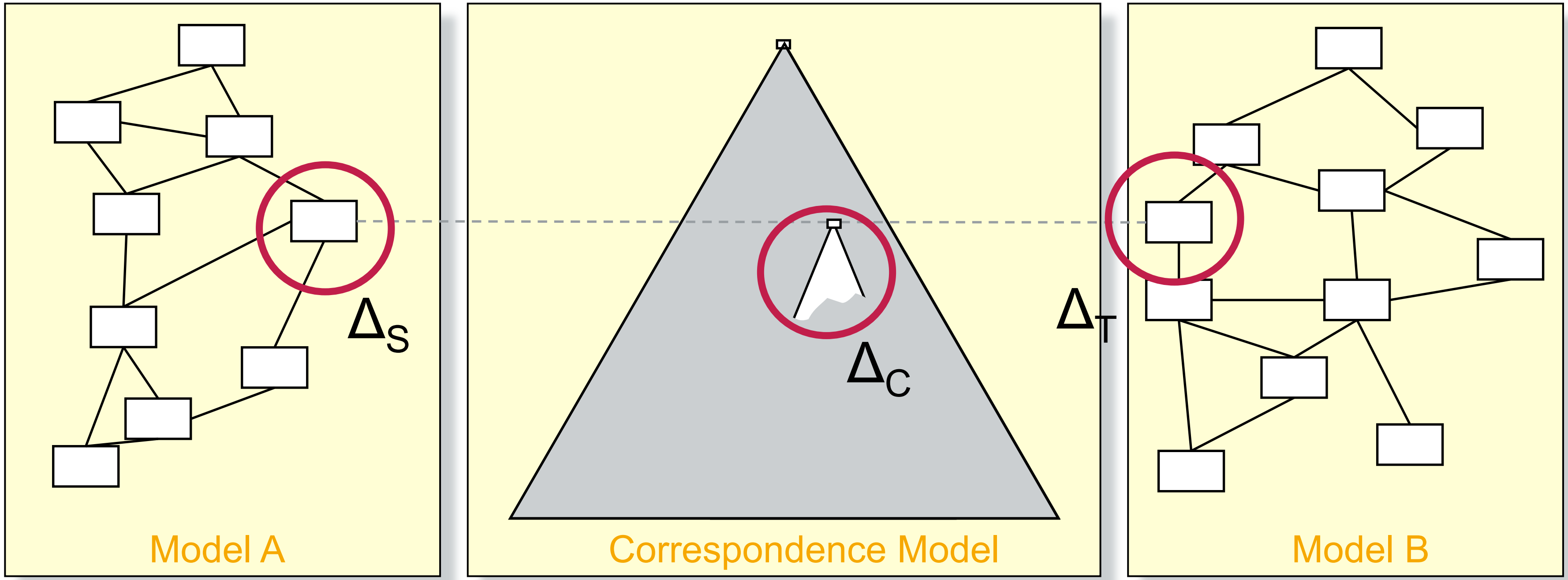


- Incremental Model Synchronization



Remark: concurrent updates are not considered!

2. Optimality Result:
Synchronization Problem



- forward: Δ_S implies to compute $\Delta_C + \Delta_T$
- backward: Δ_T implies to compute $\Delta_C + \Delta_S$ (can be handled analogous to the forward case)

Remark: $O(\Delta_S + \Delta_C + \Delta_T)$ is thus a lower complexity bound

2. Optimality Result:
Restrictions of our TGG dialect

[14]

The rules of our TGG dialect allow only to link finite many elements in the source and target model to each other via one correspondence elements at once. Consequently,

- $O(\Delta_C) = O(\Delta_S + \Delta_C + \Delta_T)$
- $O(\Delta_C)$ is also a lower bound for the synchronization problem

Implications for BX:

- Do similar restrictions apply for other techniques (GT or other domains)?

2. Optimality Result: Practically Relevant Models

[14]

Study of available data concerning model transformations reveals:

- P1 The models contain at most one “unbounded” link (link in O(n)-O(1)).
- P2 Source and target models are weakly connected graphs.
- P3 Expressions in the rules can be evaluated in constant time.
- P4 TGG transform all elements of the source and target models*

Implications for BX:

- Do similar limitations for practical cases also apply in other domains (SE is already considered)?

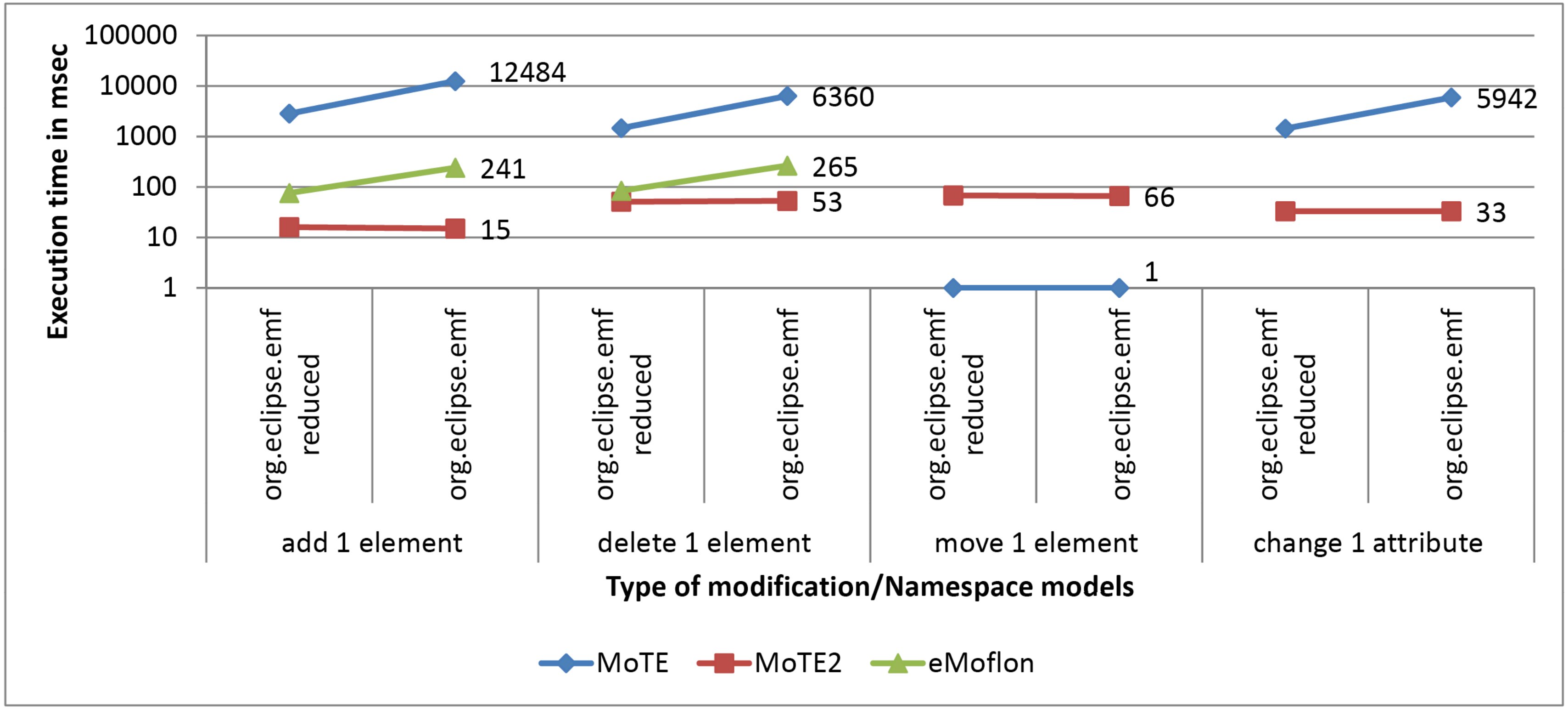
2. Optimality Result: Optimal Model Synchronization

[14]

- synchronization algorithm with repair ➔ The overall synchronization requires $O(\Delta_C)$ rule checks and applications
- dynamic pattern matching strategy ➔ Finding all matches for a single rule starting from a given correspondence is in $O(\Delta_C)$ steps, if it conceptually has to traverse the unbounded link (by using the untransformed elements set) or $O(1)$ otherwise.
- As for one correspondence only for each rule all matches have to be searched (still $O(\Delta_C)$), the combined execution of the checks and applications for the synchronization scheme requires also only $O(\Delta_C)$ steps.

Result: if P1-P4 hold, complexity is $\mathbf{O(\Delta_C)}$ and thus **optimal!**

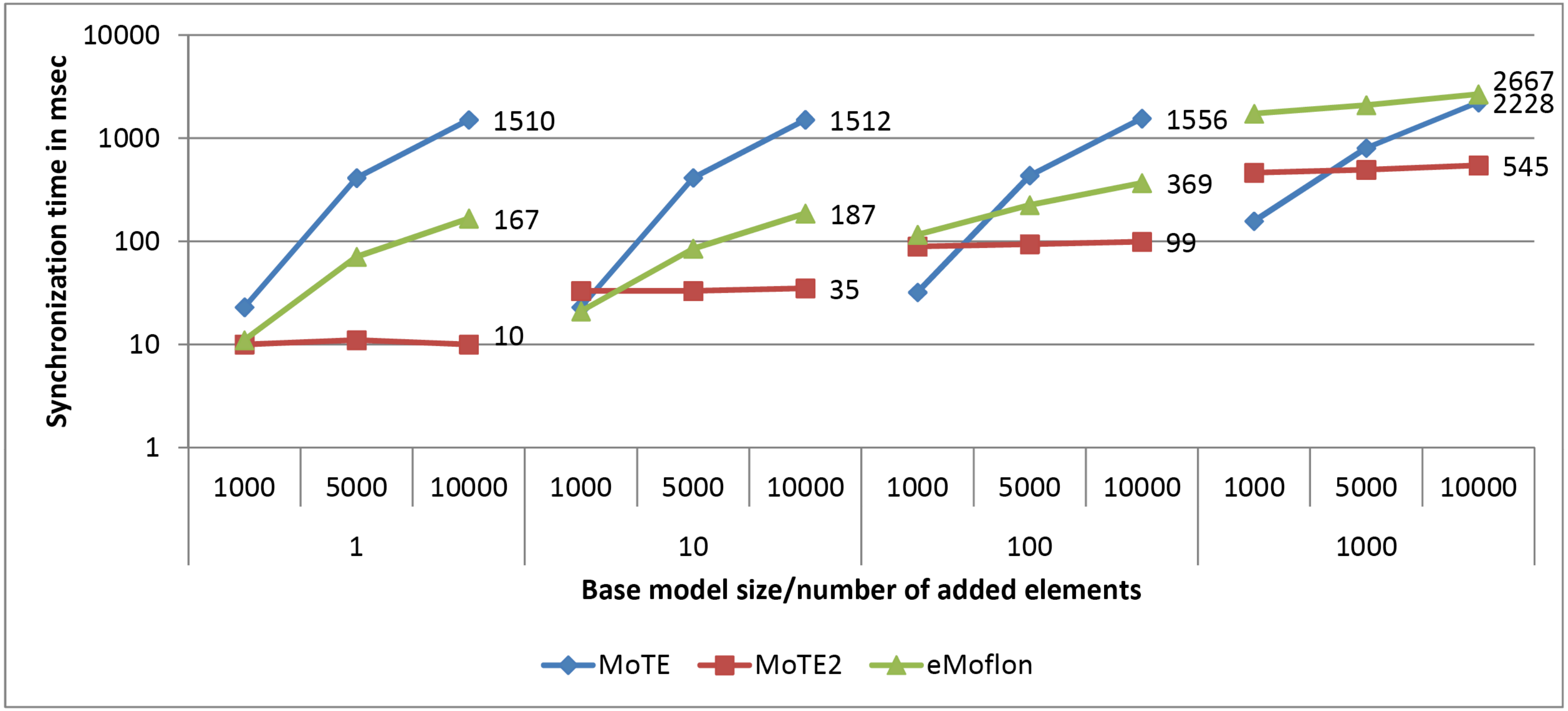
2. Optimality Result: Results for the Implementation



- UML2RDBMS standard benchmark
- org.eclipse.emf reduced: 3227 classes and 2295 associations
- org.eclipse.emf: 6455 classes and 6761 associations

➔ no increase with growing model size!

2. Optimality Result:
Results for the Implementation

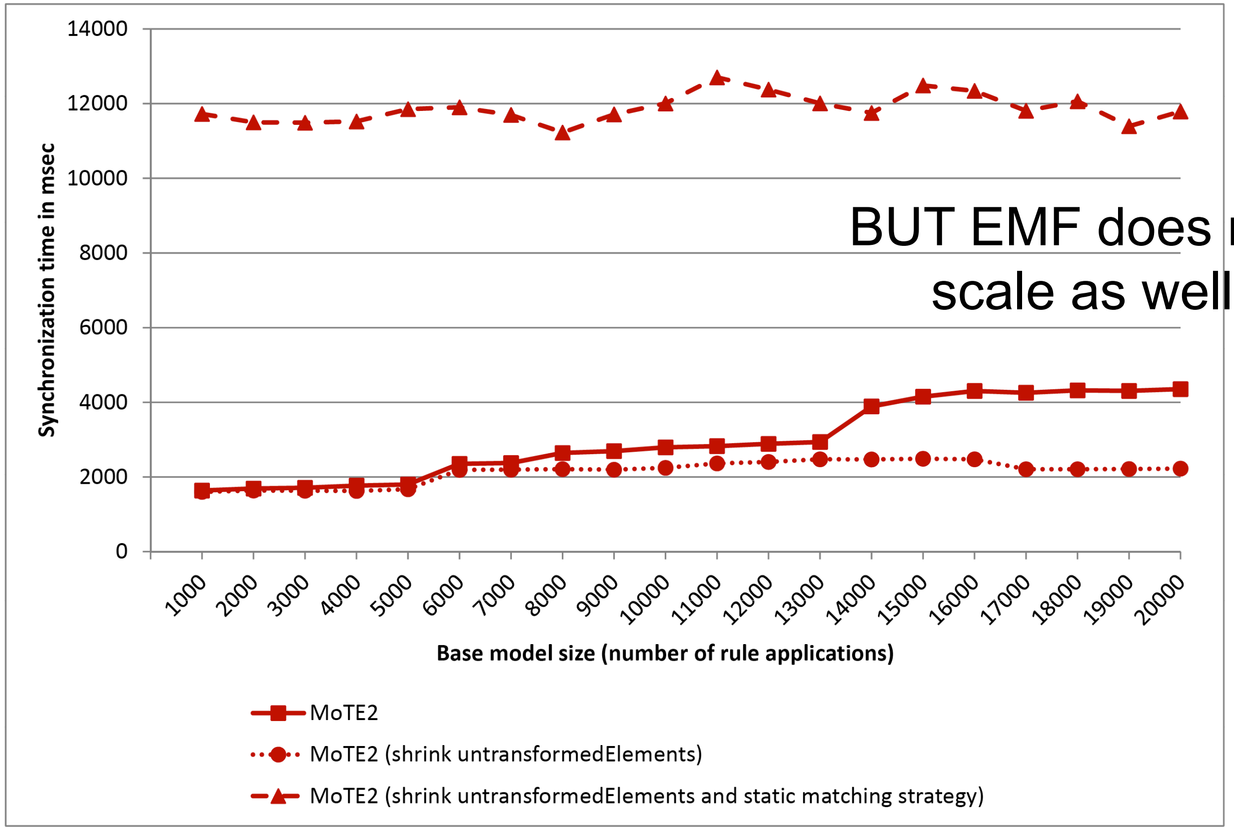


- UML2RDBMS standard benchmark
- generated test data …

➔ linear increase with growing delta size!

2. Optimality Result:
Results for the Implementation

Hasso Plattner Institut



BUT EMF does not scale as well …

- SDL2UML example with adding 1000 elements
- UML model is about six times the number of rule applications

➔ some minimal increase with growing model size due to EMF!

3. Open Issues & Implications

[14]

Open Issues (for TGGs):

- Which not yet supported TGG concepts can be added without loosing the optimality result? Priorities ✓, … ?

Implications (for TGGs):

- Multiple model synchronization with TGGs that link multiple models would still be highly efficient (remark: interpreter vs. compiler!)

Implications (for BX):

Observation: The restrictions for practical problems matter!

➔ **Maybe it is more promising to look for solutions for relevant classes of practical problems rather then the general case?**

➔ Maybe we can decompose relevant problems into some relevant classes and compose our solution from solutions for each class?

References (1/2)

www.mdelab.org

[1] Holger Giese and Leen Lambers and Basil Becker and Stephan Hildebrandt and Stefan Neumann and Thomas Vogel and Sebastian Wätzoldt. Graph Transformations for MDE, Adaptation, and Models at Runtime. In Marco Bernardo and Vittorio Cortellessa and Alfonso Pierantonio, ed., Formal Methods for Model-Driven Engineering, vol. 7320 of Lecture Notes in Computer Science (LNCS), pages 137-191. Springer-Verlag, 6 2012.

[2] Holger Giese and Stephan Hildebrandt and Leen Lambers. Bridging the Gap Between Formal Semantics and Implementation of Triple Graph Grammars - Ensuring Conformance of Relational Model Transformation Specifications and Implementations. Software and Systems Modeling ():1-27, 2012.

[3] Stephan Hildebrandt, Leen Lambers, Holger Giese. Complete Specification Coverage in Automatically Generated Conformance Test Cases for TGG Implementations. In Keith Duddy, Gerti Kappel, ed., Theory and Practice of Model Transformations, volume 7909 of Lecture Notes in Computer Science, pages 174-188, 2013. Springer.

[4] Stephan Hildebrandt and Leen Lambers and Holger Giese and Dominic Petrick and Ingo Richter. Automatic Conformance Testing of Optimized Triple Graph Grammar Implementations. In Andy Schürr and Daniel Varró and Gregely Varró, ed., Applications of graph Transformation with Industrial Relevance, 4th International Symposium, (AGTIVE 2011), volume 7233 of LNCS, pages 238-253, 2012. Springer.

[5] Holger Giese, Leen Lambers. Towards Automatic Verification of Behavior Preservation for Model Transformation via Invariant Checking. In Proceedings of International Conference on Graph Transformation (ICGT'12), volume 7562 of LNCS, pages 249-263, 2012. Springer.

[6] Basil Becker and Leen Lambers and Johannes Dyck and Stefanie Birth and Holger Giese. Iterative Development of Consistency-Preserving Rule-Based Refactorings. In Jordi Cabot and Eelco Visser, ed., Theory and Practice of Model Transformations, Fourth International Conference, ICMT 2011, Zurich, Switzerland, June 27-28, 2011. Proceedings, volume 6707 of Lecture Notes in Computer Science, pages 123-137, 2011. Springer / Heidelberg.

[7] Holger Giese and Stephan Hildebrandt and Stefan Neumann and Sebastian Wätzoldt. Industrial Case Study on the Integration of SysML and AUTOSAR with Triple Graph Grammars. Technical Report 57, Hasso Plattner Institute at the University of Potsdam, 2012.

References (2/2)

[8] Holger Giese and Stefan Neumann and Stephan Hildebrandt. Model Synchronization at Work: Keeping SysML and AUTOSAR Models Consistent. In Gregor Engels and Claus Lewerentz and Wilhelm Schäfer and Andy Schürr and B. Westfechtel, ed., Graph Transformations and Model Driven Enginering - Essays Dedicated to Manfred Nagl on the Occasion of his 65th Birthday, volume 5765 of Lecture Notes in Computer Science, pages 555-579, 2010. Springer Berlin / Heidelberg. 10.1007/978-3-642-17322-6_24

[9] Thomas Vogel and Stefan Neumann and Stephan Hildebrandt and Holger Giese and Basil Becker. Incremental Model Synchronization for Efficient Run-Time Monitoring. In Sudipto Ghosh, ed., Models in Software Engineering, Workshops and Symposia at MODELS 2009, Denver, CO, USA, October 4-9, 2009, Reports and Revised Selected Papers, vol. 6002 of Lecture Notes in Computer Science (LNCS), pages 124-139. Springer-Verlag, 4 2010.

[10] Leen Lambers and Stephan Hildebrandt and Holger Giese and Fernando Orejas. Attribute Handling for Bidirectional Model Transformations: The Triple Graph Grammar Case. In Proceedings of First International Workshop on Bidirectional Transformations, volume 49 pages 1-16, 2012. EC-EASST.

[11] Stephan Hildebrandt, Leen Lambers, Basil Becker, Holger Giese. Integration of Triple Graph Grammars and Constraints. In Christian Krause, Bernhard Westfechtel, ed., Proceedings of the 7th International Workshp on Graph Based Tools, volume 52 pages 1-12, 2012. EC-EASST.

[12] Holger Giese and Stephan Hildebrandt and Andreas Seibel. Improved Flexibility and Scalability by Interpreting Story Diagrams. In Tiziana Magaria and Julia Padberg and Gabriele Taentzer, ed., Proceedings of the Eighth International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT 2009), volume 18 2009. Electronic Communications of the EASST.

[13] Holger Giese and Stephan Hildebrandt. Efficient Model Synchronization of Large-Scale Models. Technical Report 28, Hasso Plattner Institute at the University of Potsdam, 2009.

[14] Stephan Hildebrandt. XYZ. PhD Thesis, Hasso Plattner Institute at the University of Potsdam, 2013.