Mapping-Aware Megamodeling: Design Patterns and Laws

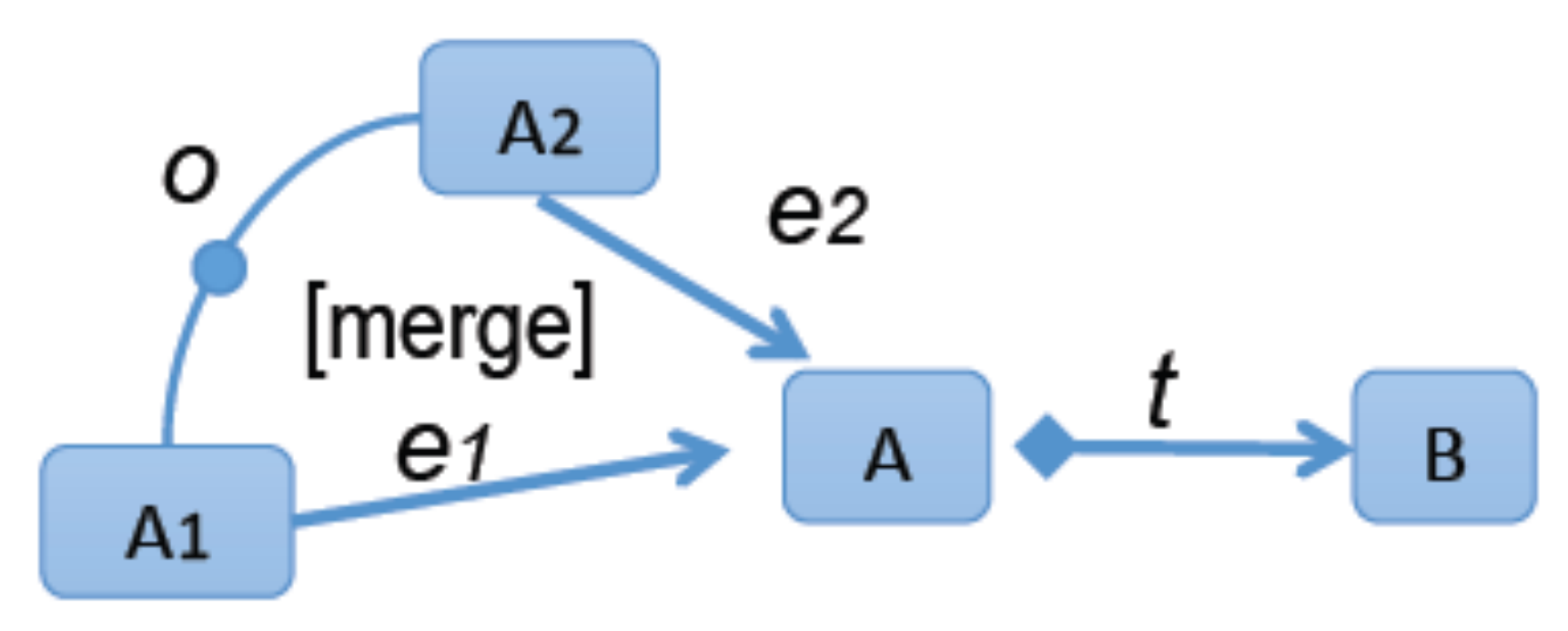
Zinovy Diskin[1,2], Sahar Kokaly[1], Tom Maibaum[1]
[1]NECSIS, McMaster University, Hamilton, Canada
E-mail: diskinz@mcmaster.ca, kokalys@mcmaster.ca, tom@maibaum.org
[2]University of Waterloo, Canada

Summary

Megamodeling is the activity of specifying systems of models and mappings, their properties, and operations over them. The latter functionality is the most important for applications, and megamodels are often used as an abstract workflow language for model processing. To be independent of a particular modeling language, typical megamodels reduce relationships between models to unstructured edges encoding nothing but a labeled pair of models, thus creating a significant gap between megamodels and code implementing them.

To bridge the gap, we propose mapping-aware megamodels, which treat edges as model mappings: structured sets of links (pairs of model elements) rather than pairs of models. The workflow can then be represented as an algebraic term built from elementary operations with models and model mappings.

Megamodeling and its challenges



a)

```
import models A1, A2
import overlap O=O(A1, A2)
merge (A1,A2) into A by O
import transformation t
apply t to A result B
```

b)

Fig. 1. Workflow megamodel

Mapping-Aware Megamodeling

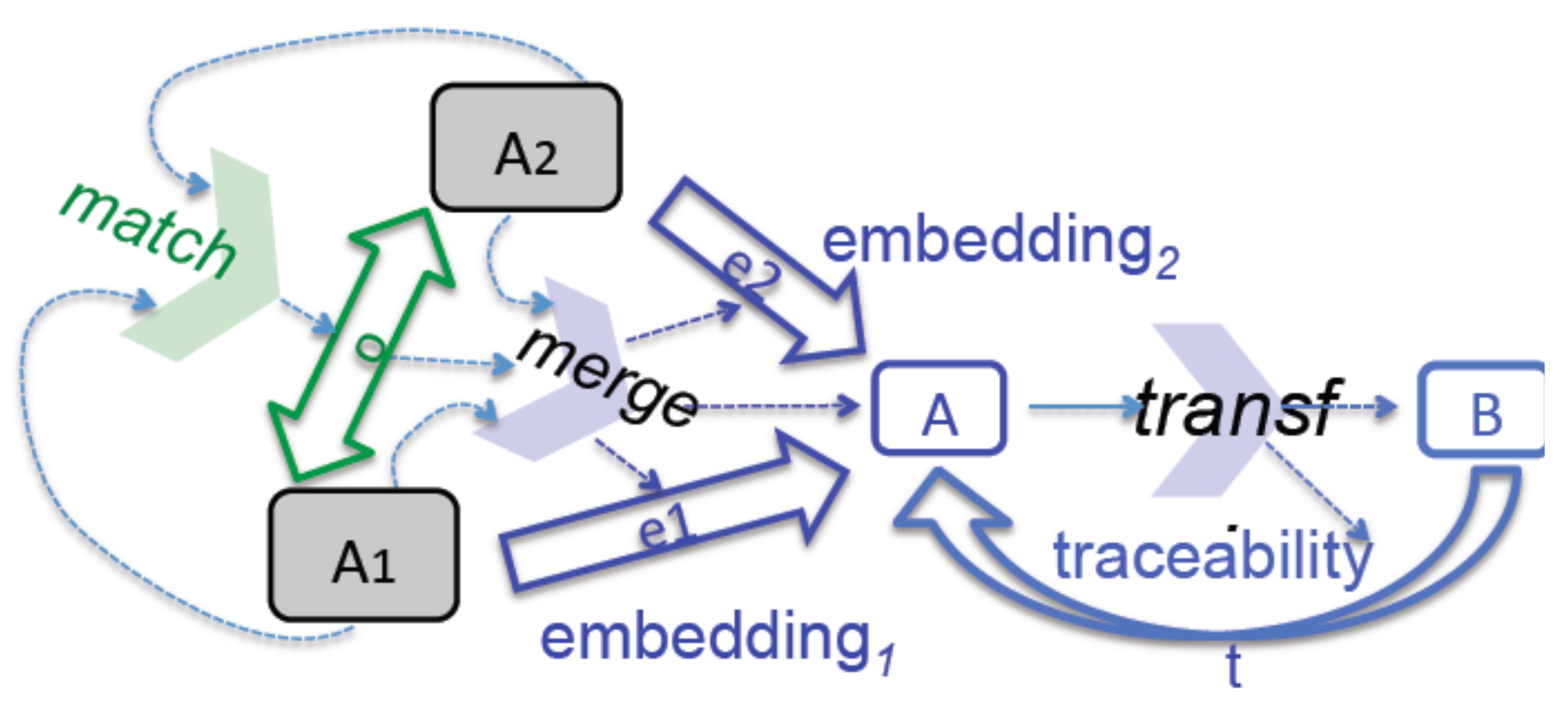


Fig. 2. A dynamic megamodel

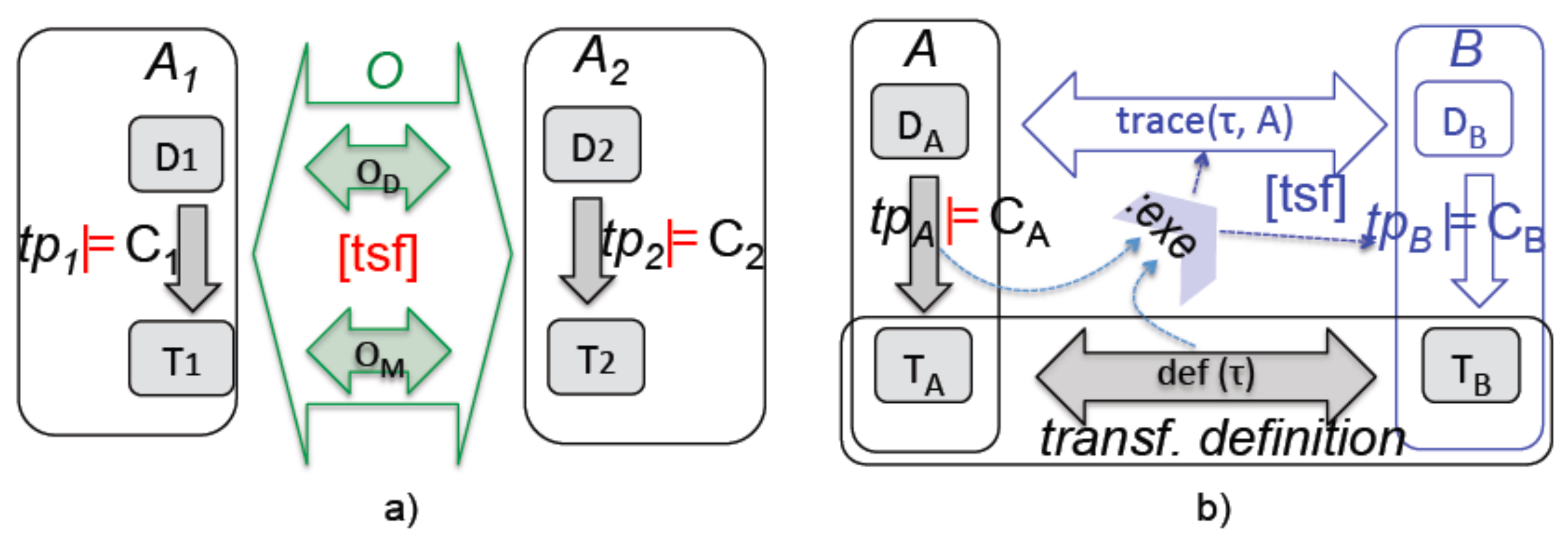


Fig. 3. Zooming into elements of megamodel Fig. 2

Design Patterns and Laws

Pattern 1: A model is a total typing mapping from model's data graph to model's metadata graph, which comprises a type graph and a constraint graph.
The Laws: Typing must be a correct graph morphism, and all constraints de-clared in the metamodel are to be satisfied.

Pattern 2: A model mapping is a pair of total correspondence mappings between the respective data and metadata parts of the models. Together with the respective typing mappings, they form a square of mappings.
The Laws: To ensure type-safety, the model mapping square is required to be commutative. Moreover, translations of constraints declared for the source of a mapping are to be implied by the constraints in the target: the target is to be at least as constrained, perhaps more constrained, than the source.

Pattern 3 (Model Overlap): Overlap of two models is a span of model mappings. The latter are either total, if overlapping amounts to correspondence equations between elements, or partial, if new constraints are introduced. Overlap of **n**-models is a set of **m**-ary (total and partial) spans with 2 ≤ **m** ≤ **n**.
The Laws: The merge of a system of models modulo their overlap span is a correct premodel. However, it can violate inter-model constraints. This is what we call inconsistency.

Pattern 4 (Descriptive views), Pattern 5 (Prescriptive views)

Pattern 6: A model transformation definition is a span of metamodel mappings, which can be executed in both directions. A full set of laws is an open question. Two basic laws , identity propagation and weak invertibility, are specified in [4].

Conclusions:

Mapping-Aware megamodeling as a program to be realized:
Everything in the MMt world consists of graphs and graph mappings.
- There is a set of **basic structural blocks** to build all well-structured intermodel relations by composing these blocks.

MMt operations are operations over graphs and graph mappings.
- An extensible set of **basic operations** to build all well-structured operations over models by composition.

Design Patterns for MMt based on category theory. Hence,
- Mathematical Laws for rewriting and optimization
- Clear semantics: honest and manageable

Future Work:

- Workflow/programming language for megamodeling.
- Application to various complex MDE scenarios.
- Case studies.

Acknowledgement:

This work was done as part of the NECSIS project, funded by Automotive Partnership Canada and NSERC.

References:

1. Diskin, Z., Kokaly, S., Maibaum, T.: Mapping-Aware Megamodeling: Design Patterns and Laws. SLE'13
2. Diskin, Z., Maibaum, T., Czarnecki, K.: Intermodeling, queries, and kleisli categories. FASE'12
3. Diskin, Z.: Model synchronization: Mappings, tiles, and categories. GTTSE'09
4. Diskin, Z., Xiong, Y., Czarnecki, K., Ehrig, H., Hermann, F., Orejas, F.: From state- to delta-based bidirectional model transformations: The symmetric case. MoDELS'11