Model Transformation and Semantics: The Evolution of Meaning

Reiko Heckel, Gregor Engels

University of Paderborn, Germany

Claims

- Evolution should be dealt with at the level of models.
- This requires to formalize and check the consistency between models based on their meaning.
- Such relations can be checked by static analysis using partial mappings of models into specialized semantic domains.

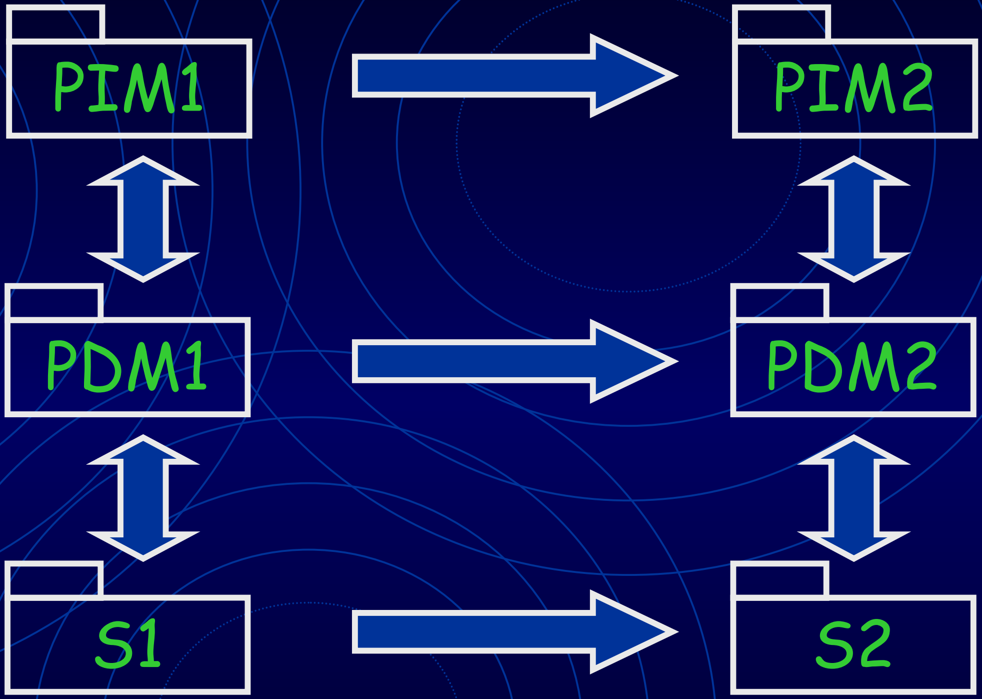
Evolution should be dealt with at the level of models.

Models are central to software development

- for capturing requirements
- as means for communication and documentation
- to support integration and evolution of systems in a heterogeneous environment
  - OMG's model driven architecture (MDA)

Model-driven Evolution

- plattform-independent models
  →UML core
- plattform-dependent model:
  → specific profile
- software systems:
  → specific PL and plattform



Separate

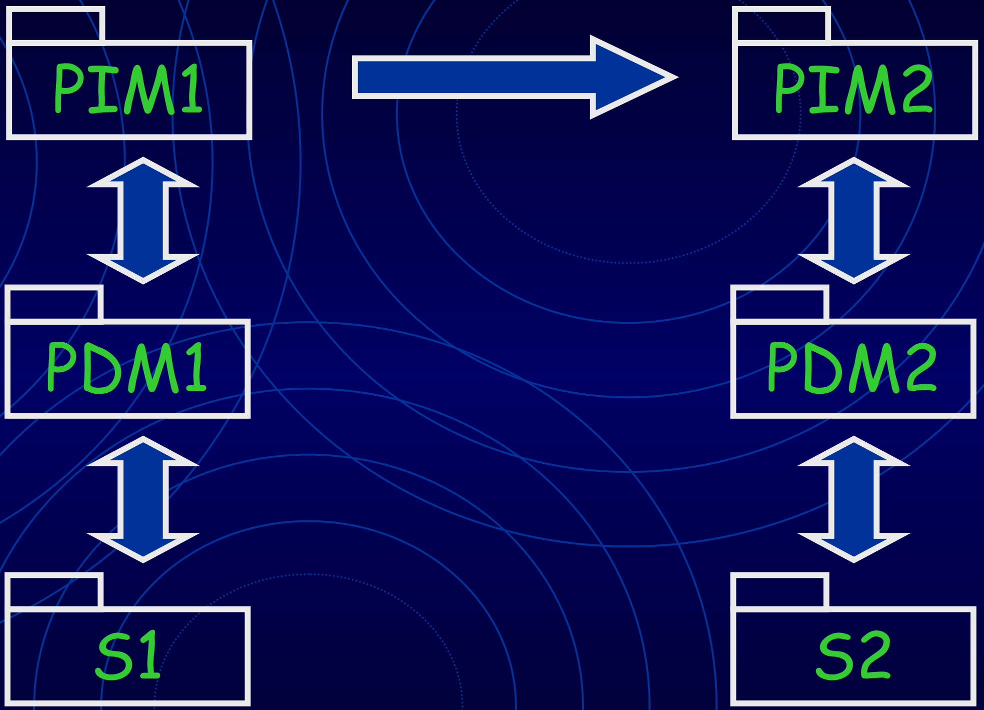
- migration to new technology from introduction of new functionality
- forward and reverse engineering from evolution

This requires to check consistency between models…

a) horizontal: between views of the same model, e.g.
   - logical
   - dynamic
   - functional

b) vertical: between abstraction levels

c) over time: 1 → 2



Requires to relate artefacts

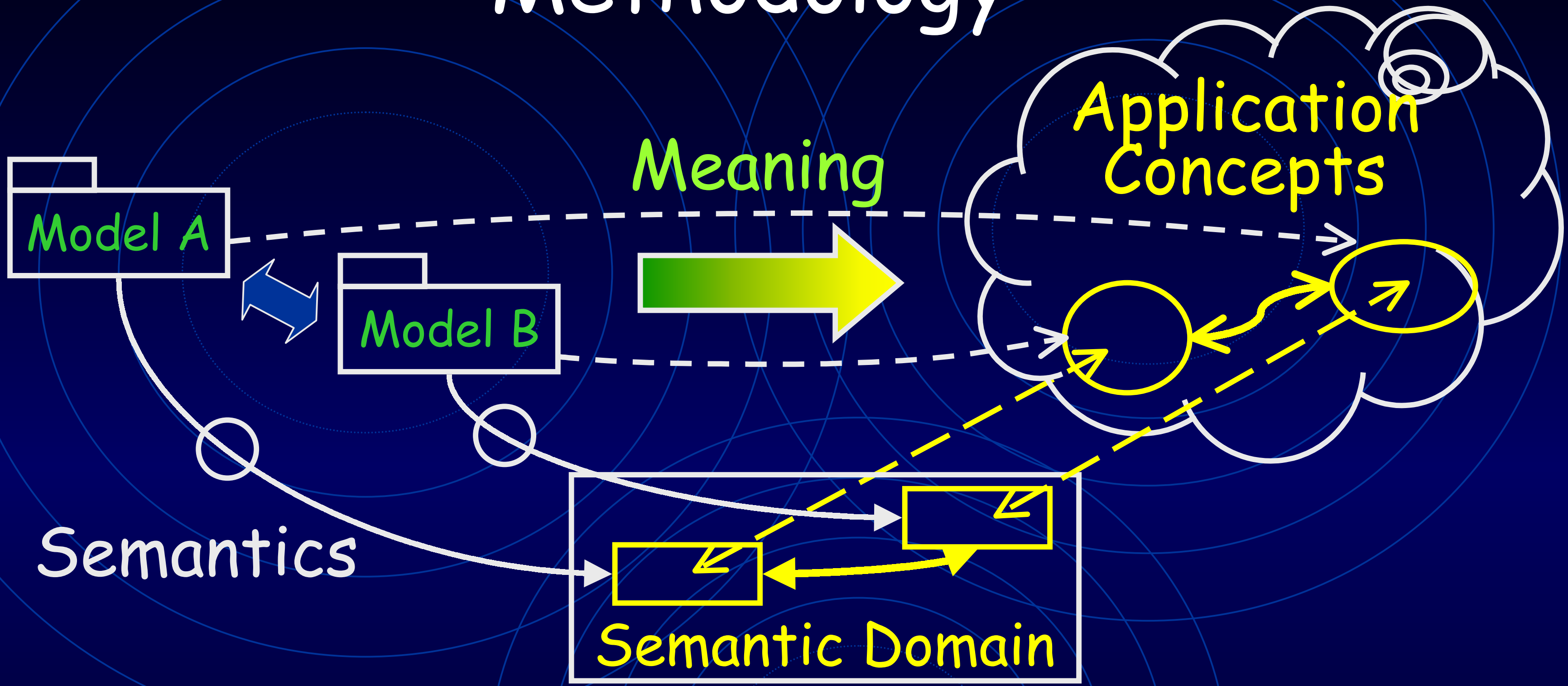
- expressed in the same language (PIM1 → PIM2)
- expressed in different languages (PIM1 → PDM1)

… based on their meaning.

This requires a formalization of their meaning (semantics), but:

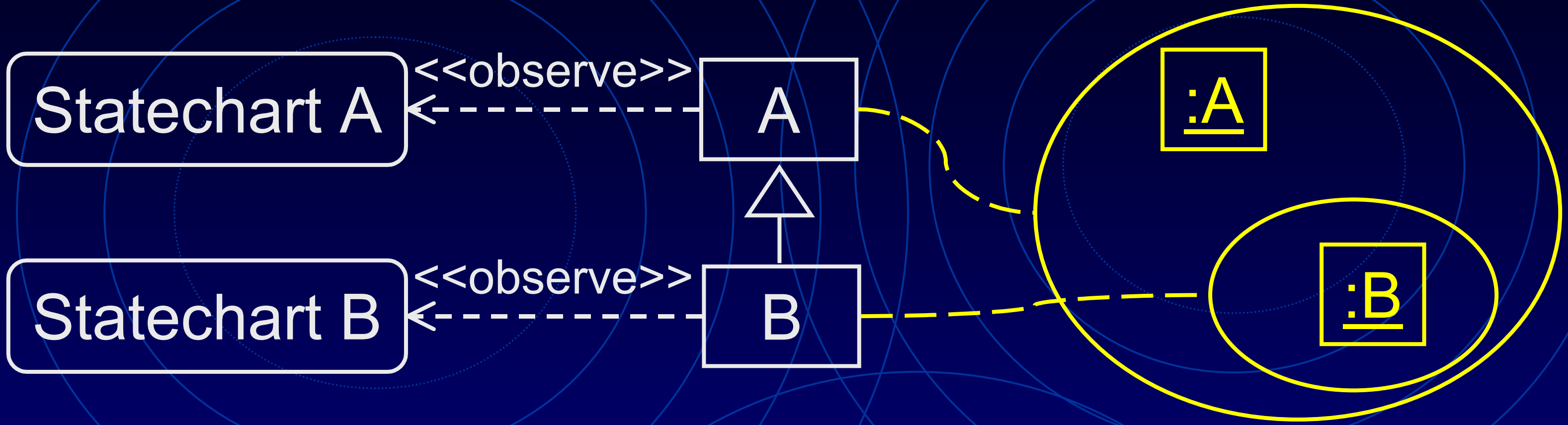
- Is a complete formal semantics realistic ?
  - Can we justify/agree upon semantic choices to make ?
  - Can we explain it to practitioneers ?
- How can we still formalize (and implement) semantic consistency rules ?
  - ✓ Map only those aspects where the consistency problem occurs to a specialized semantic domain with language and tool support for specifying and analysing those rules.

Methodology



1. identify conceptual relation between the meanings of models
2. choose semantic domain with language and tool support
3. define partial mapping for those aspects of models that are relevant to conceptual relation
4. specify semantic relation using language of semantic domain

Sample Consistency Problem



1. Identify Conceptual Relation:

Statecharts A and B specify two views of the behavior of instances of Class B.

→ behavior inheritance, dual interpretations:

- invocable behavior: substitution principle
- observable behavior: projection

intended interpretation may be indicated by stereotypes

2. Choose Semantic Domain: Communicating Sequential Processes

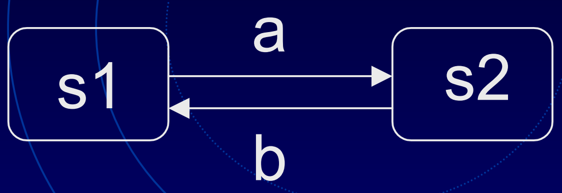
- Language for behavior: CSP processes

P	::=	STOP |	termination
		$a \rightarrow P$ |	action prefix
		$P \;\square\; P$ |	external choice
		$P \setminus a$ |	restriction
		…	

- Semantically: traces, failures, …
- Refinement relation between processes

  $P \bullet_T Q$ iff traces(Q) $\subseteq$ traces(P)
- Tool support: FDR (Formal Systems)

3. Define partial mapping: Statecharts → CSP

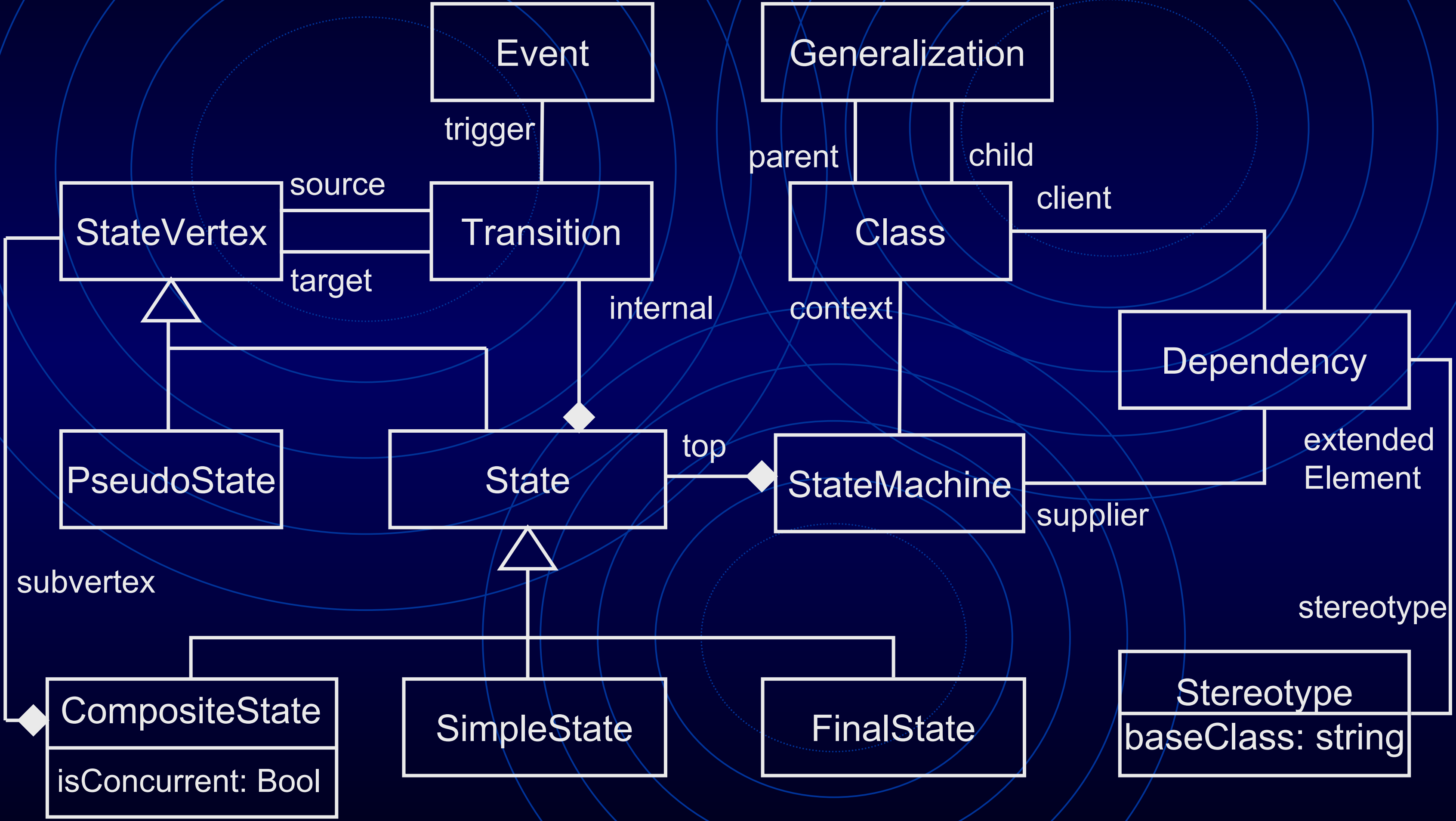


State(s1) = a → State(s2)
State(s2) = b → State(s1)

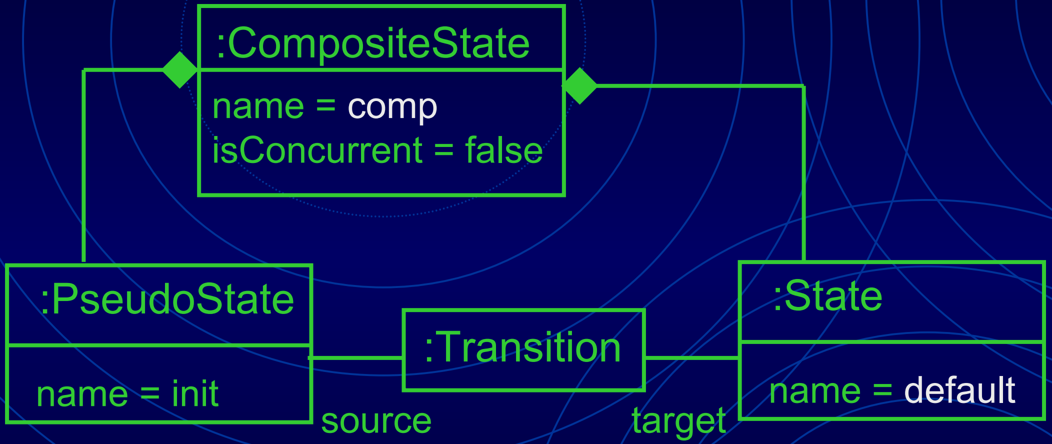
How to define this formally ?

- metamodel presentation of UML statecharts → graphs as abstract syntax
- mapping rules from graphical syntax to textual language of semantic domain

Metamodel Fragment



Mapping Rules Statecharts → CSP



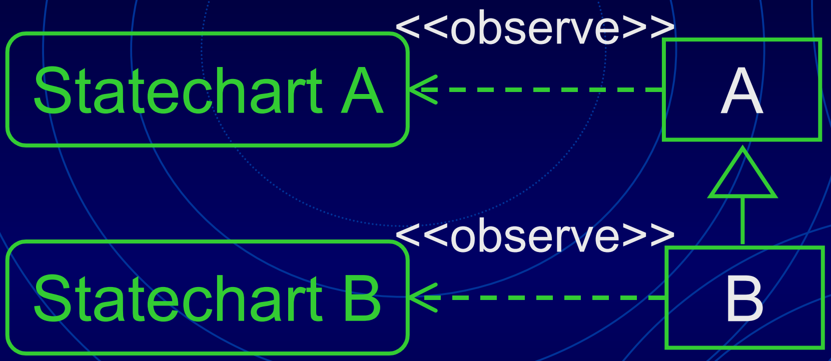
State(comp) =
  State(default)

Formally: attributed graph grammar rules
with lhs = rhs

→ no change of graph structure

→ computation of semantic attributes

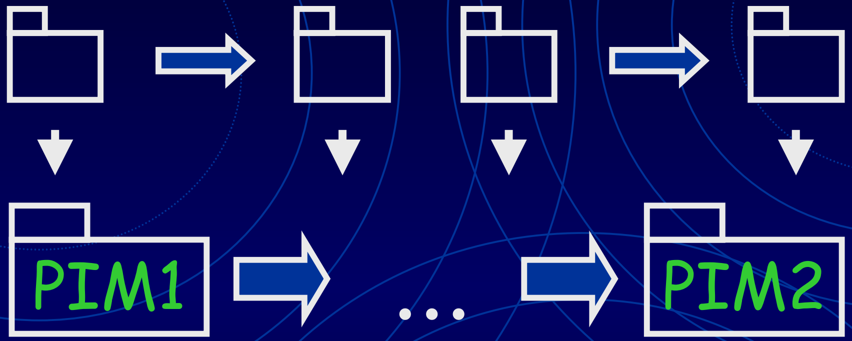
4. Specify Semantic Relation



$$\varepsilon ::= \text{State}(A) \bullet_{T} \text{State}(B) \setminus \text{NewMethods}$$

Observable behavior: each sequence of method calls observable with respect to B must result, under projection, in a sequence observable of A.

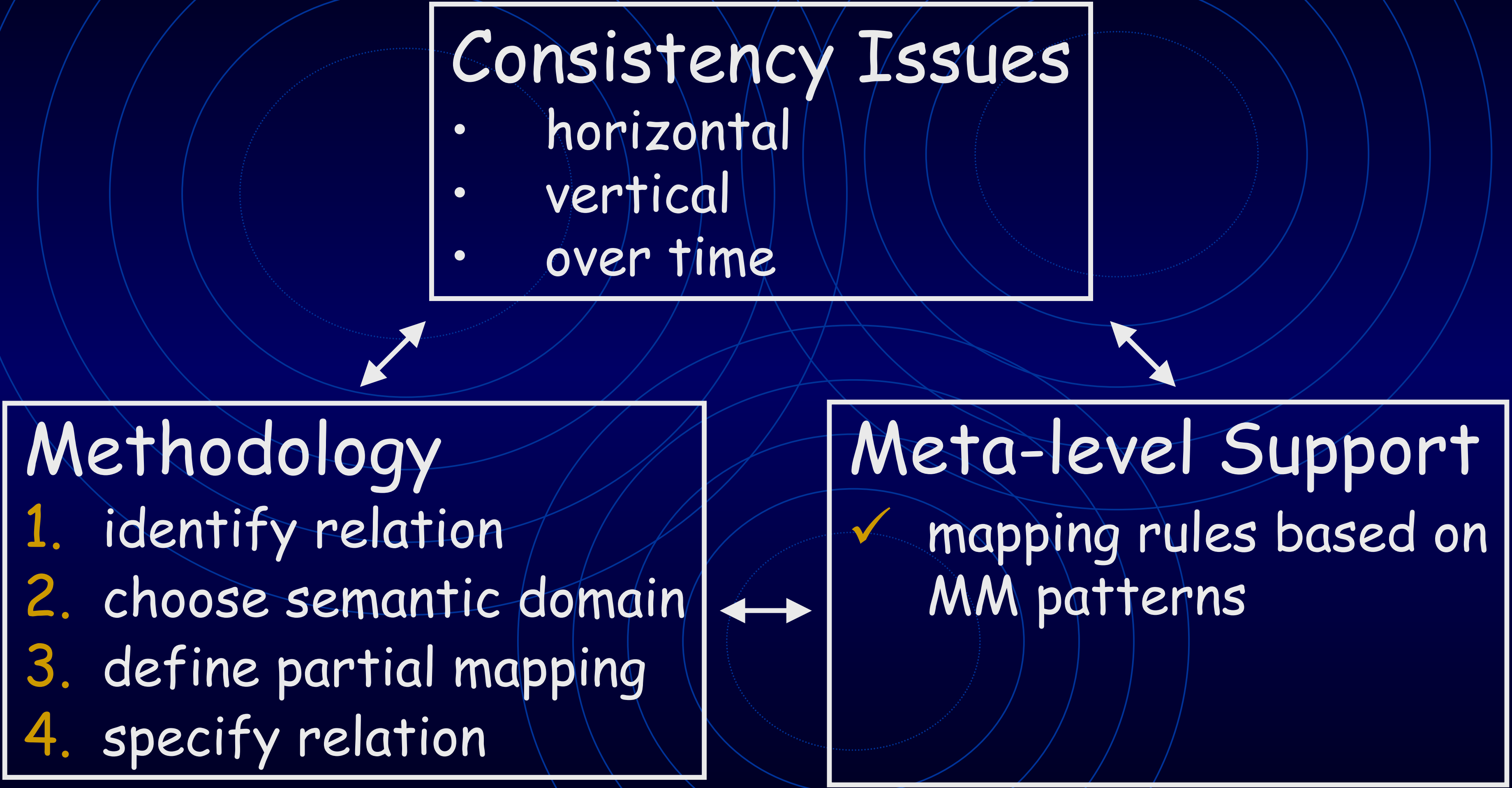
And What About Model Transformations?



- ✓ Evolution generated by model transformation rules
- ✓ Analysis of small rules instead of large models

Requires: consistency relation closed under embedding of rules into context

Summary



Consistency Issues

- horizontal
- vertical
- over time

Methodology

1. identify relation
2. choose semantic domain
3. define partial mapping
4. specify relation

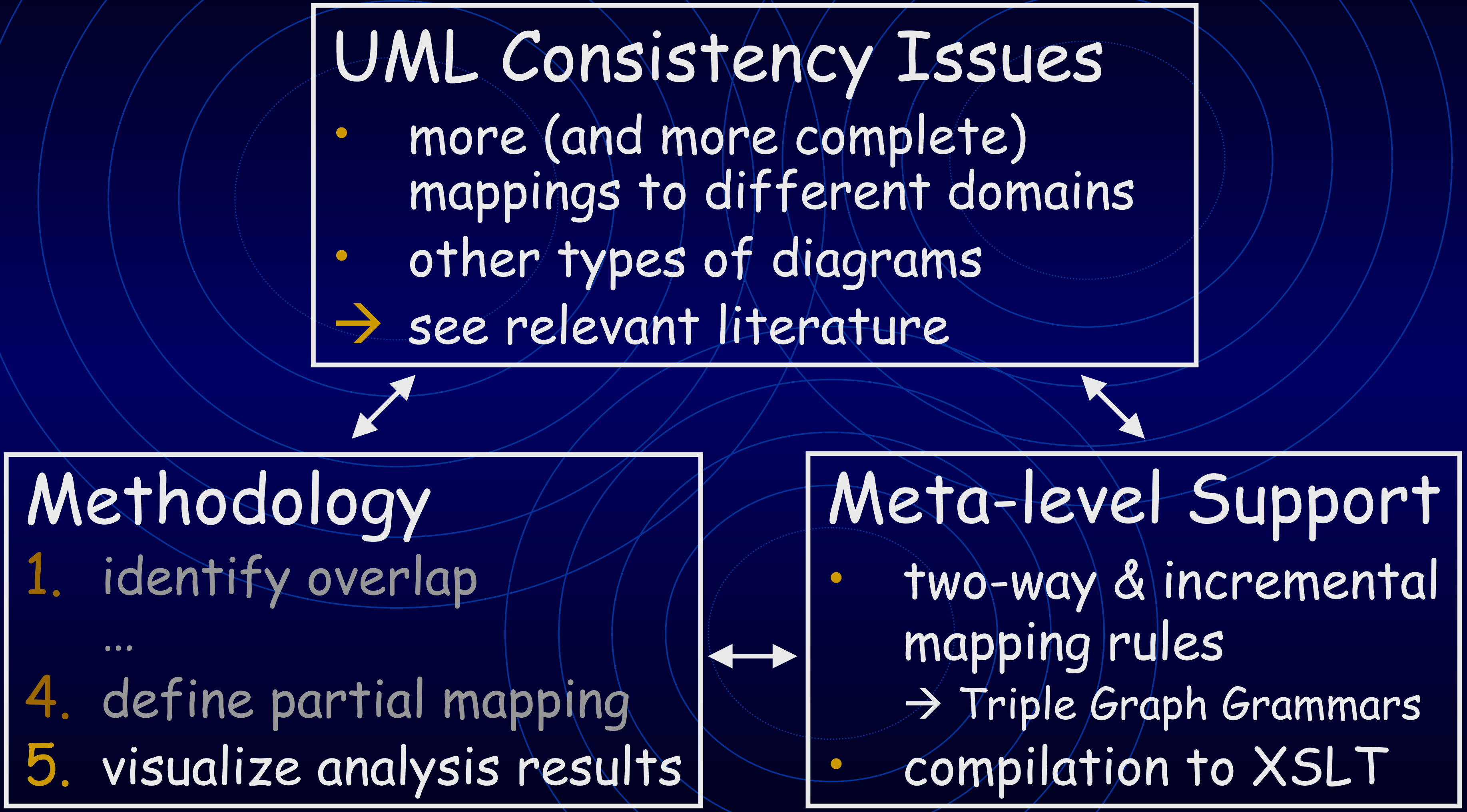
Meta-level Support

✓ mapping rules based on MM patterns

Pros and Cons

- ✓ not relying on a complete formal semantics
- ✓ flexible and extensible specification of consistency rules
- ✓ use of existing formal methods (and tools) as semantic domains
- ✗ knowledge of these domains is required

Future Work



UML Consistency Issues

- more (and more complete) mappings to different domains
- other types of diagrams

→ see relevant literature

Methodology

1. identify overlap

…

4. define partial mapping
5. visualize analysis results

Meta-level Support

- two-way & incremental mapping rules

  → Triple Graph Grammars
- compilation to XSLT