MODEL TRANSFORMATION TESTING: THE STATE OF THE ART

GEHAN M. K. SELIM, JAMES R. CORDY & JUERGEN DINGEL

Queen's UNIVERSITY

SCHOOL OF COMPUTING

NSERC CRSNG

GM

IBM®

Malina SOFTWARE CORP.

MOTIVATION & OBJECTIVES

Quite a bit of work on analysis of MTs

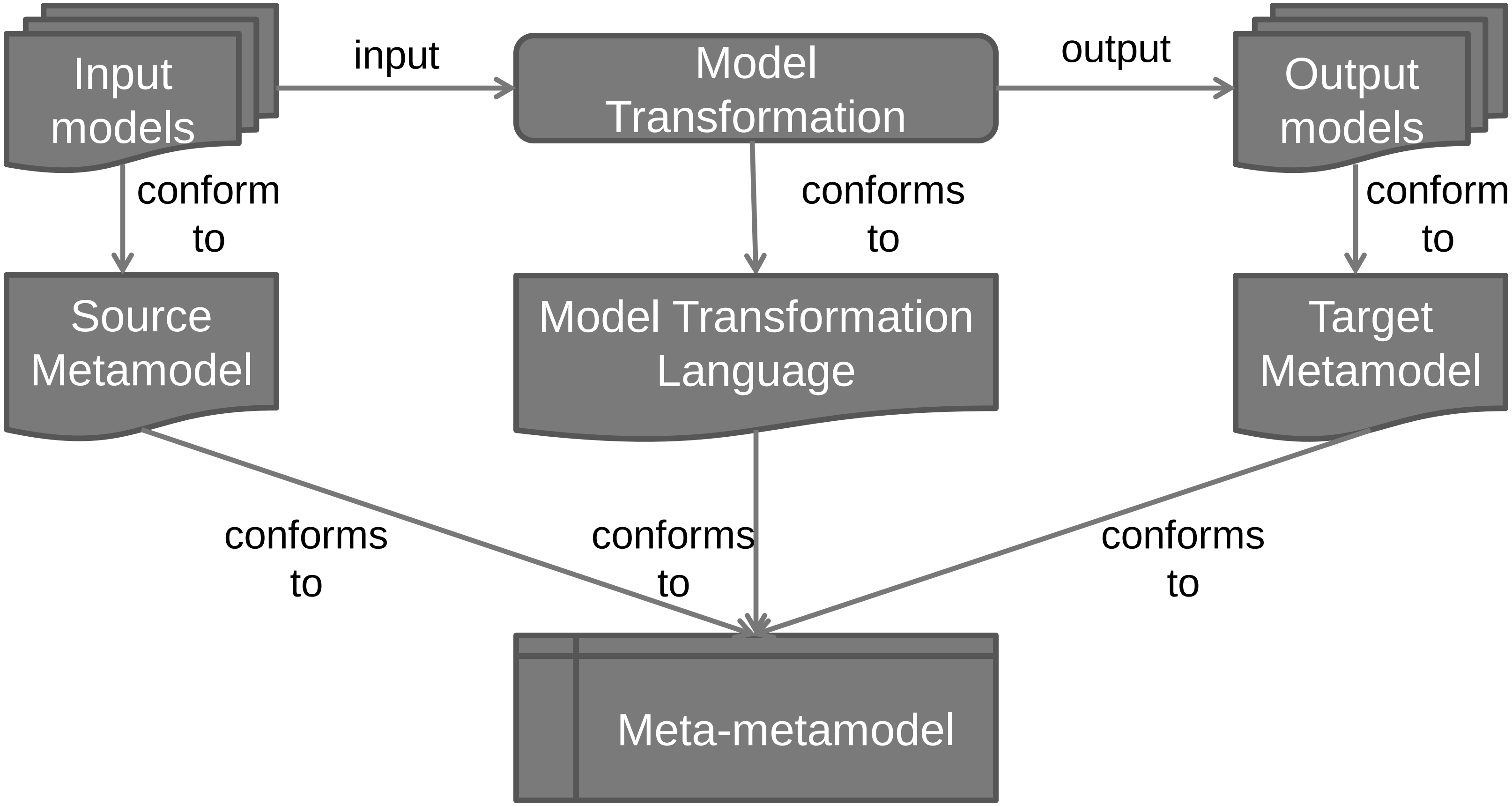
Objectives

- Survey model transformation testing
- Get insight into state-of-the-art
- Identify relevant research directions

Based on survey [GCD12]

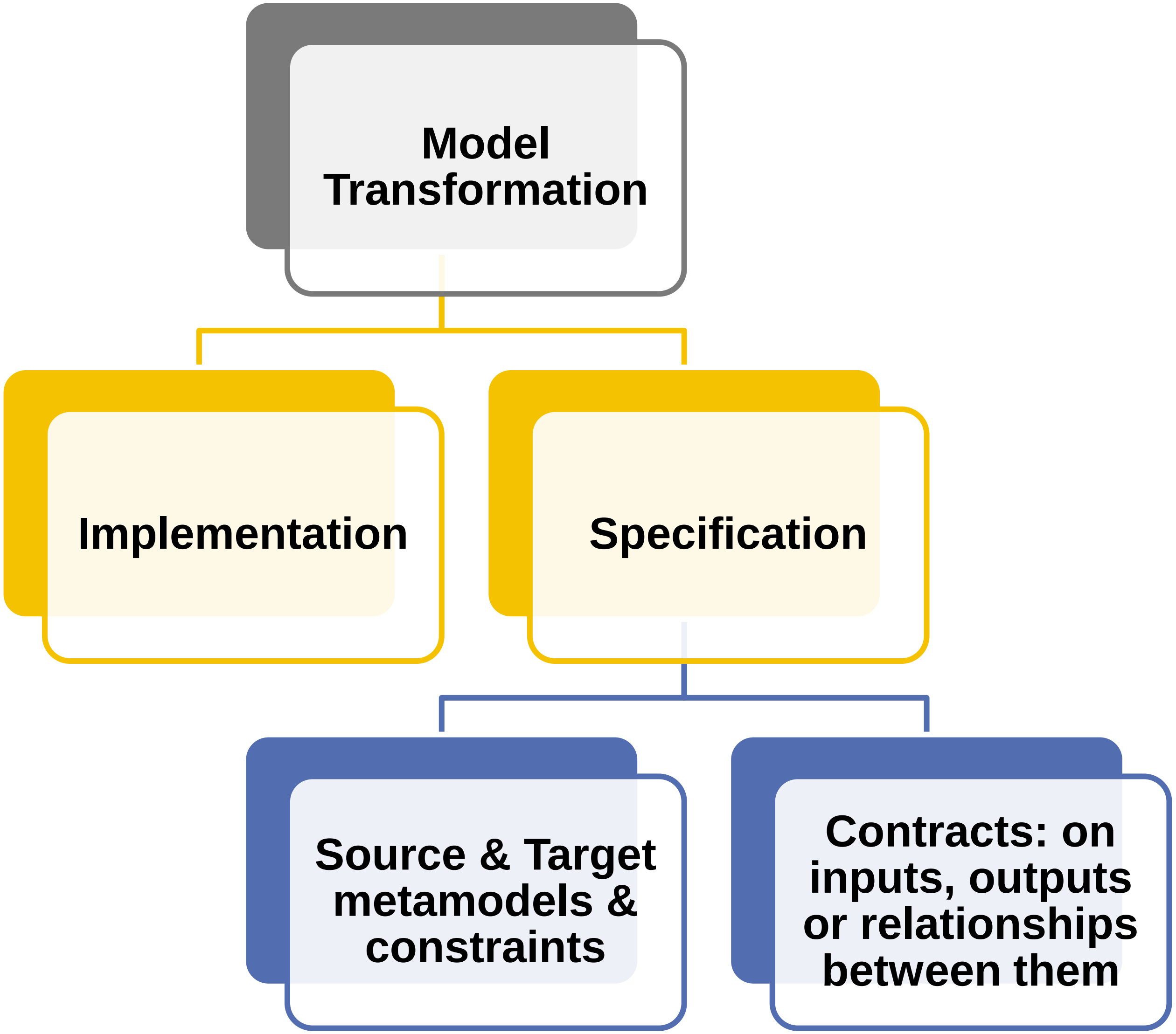
[GCD12] G. Selim, J. Cordy, and J. Dingel. “Analysis of Model Transformations”. Technical Report 2012-592, Queen’s University, Aug 2012. 58 pp.

BACKGROUND

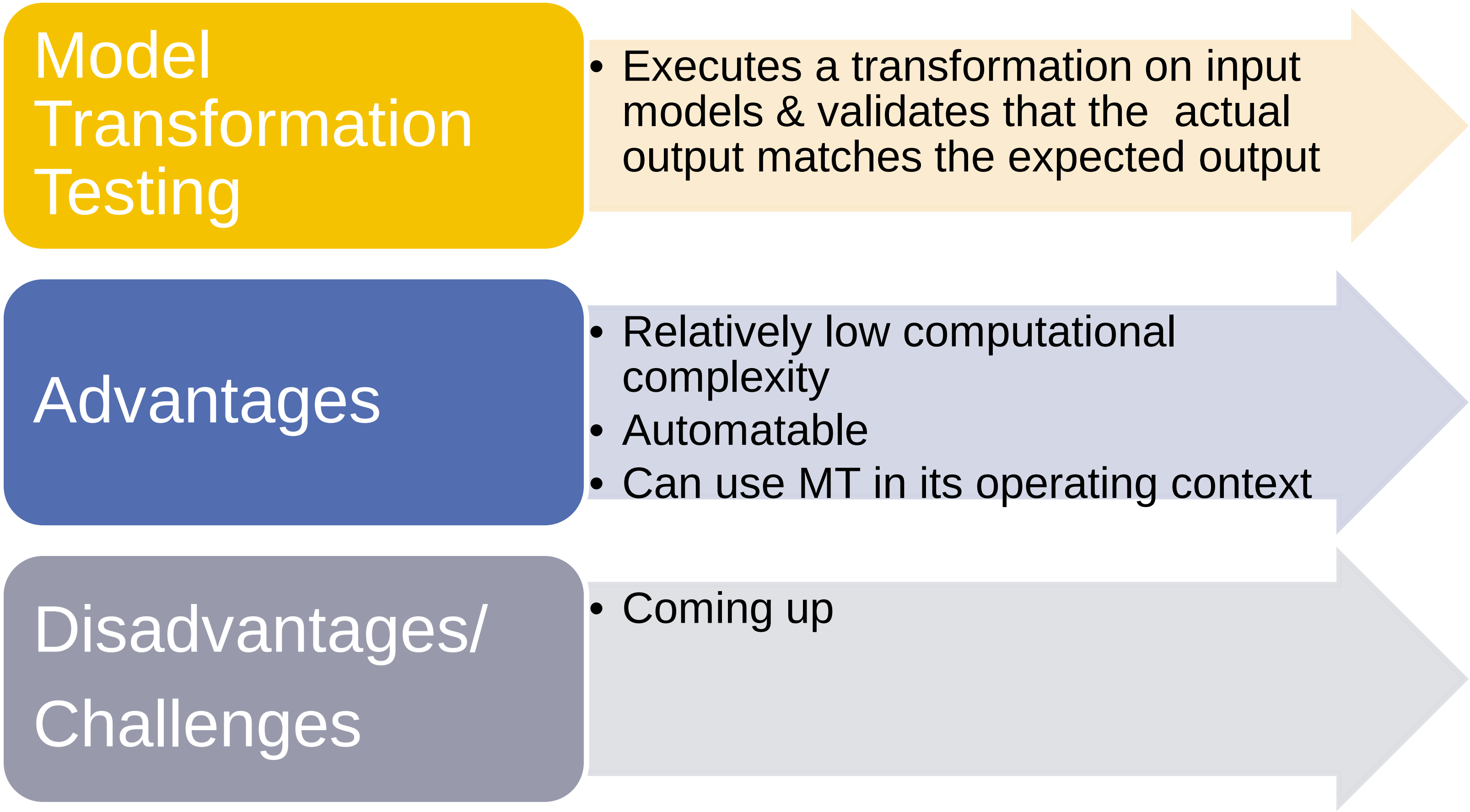


Input models
input
Model Transformation
output
Output models
conform to
conforms to
conform to
Source Metamodel
Model Transformation Language
Target Metamodel
conforms to
conforms to
conforms to
Meta-metamodel

BACKGROUND


Model Transformation
Implementation
Specification
Source & Target metamodels & constraints
Contracts: on inputs, outputs or relationships between them

BACKGROUND



Model Transformation Testing

- Executes a transformation on input models & validates that the actual output matches the expected output

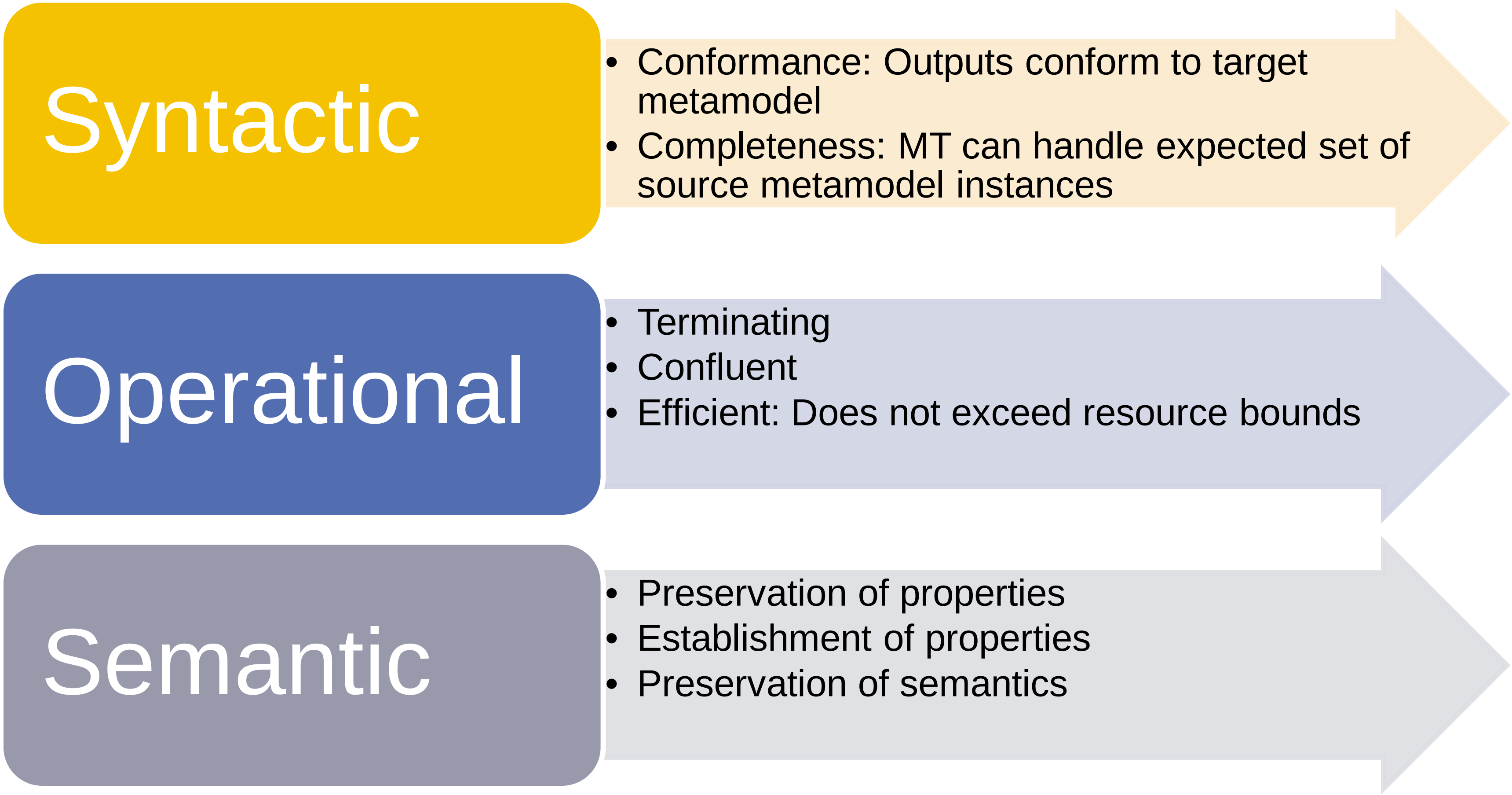
Advantages

- Relatively low computational complexity
- Automatable
- Can use MT in its operating context

Disadvantages/ Challenges

- Coming up

MT REQUIREMENTS



Syntactic

- Conformance: Outputs conform to target metamodel
- Completeness: MT can handle expected set of source metamodel instances

Operational

- Terminating
- Confluent
- Efficient: Does not exceed resource bounds

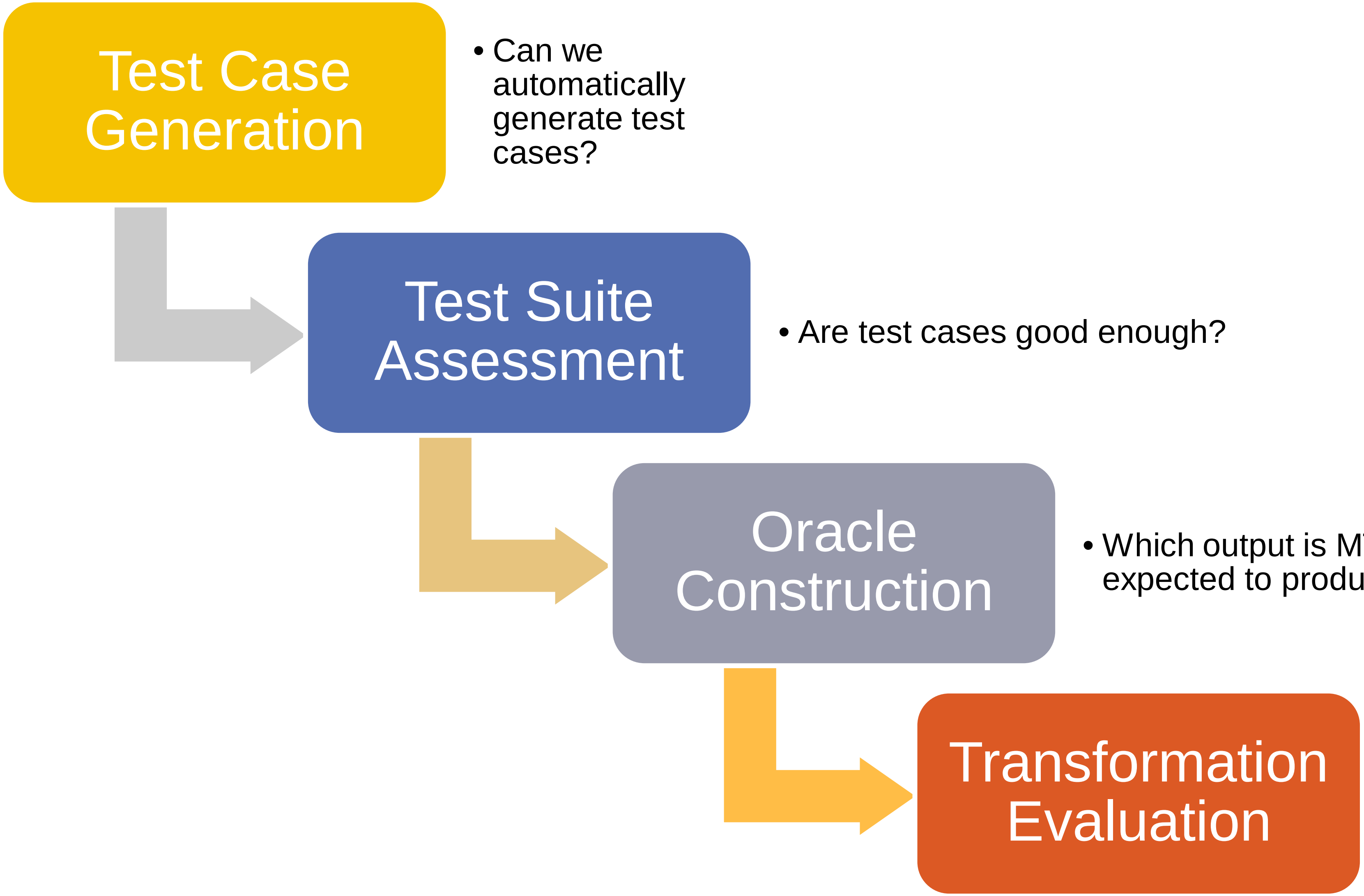
Semantic

- Preservation of properties
- Establishment of properties
- Preservation of semantics

Comments:

1. Useful? Classification of “properties”?

MT TESTING PHASES



Test Case Generation

- Can we automatically generate test cases?

Test Suite Assessment

- Are test cases good enough?

Oracle Construction

- Which output is MT expected to produce?

Transformation Evaluation

- Did MT produce expected output?

AGENDA

- **Motivation & Objective**
- **Background**
- **Model Transformation Testing Phases**
- **Phase 1: Test Case Generation**
- **Phase 2: Test Suite Assessment**
- **Phase 3: Oracle Construction**
- **Summary**

PHASE 1: TEST CASE GENERATION

Definition: Coverage

- Define test adequacy criteria
- Build a test suite that achieves coverage of the criteria
- Coverage=$\frac{|Satisfied\ Criteria|}{|Criteria|}$

Approaches

- Black-box
  - metamodel-coverage
  - contract coverage
- Grey-box
- White-box

PHASE 1: TEST CASE GENERATION

Black-Box TCG based on MM coverage (8/64)

- **Class diagrams:** e.g. Class Attribute (CA) criterion [Andrews+03, Fleurey+04, Fleurey+09, Ghosh+03]
- **Interaction Diagrams**: e.g. All Message Paths (AMP) criterion [Andrews+03, Ghosh+03, Wu+03]
- **Statecharts:** e.g. Transition coverage criterion [Haschemi, 09, Offutt+99, Wu+03]

Black-Box TCG based on contract coverage (2/64)

- **Effective MM via contracts:** Achieve coverage of MM elements referenced in pre-/post-conditions [Fleurey+04]
- **Combined specification-based coverage:** combine MM and contract-based criteria [Bauer+11a]

Comments:

2. MM-coverage-based TCG: Many diagram types not considered
3. Contract-coverage-based: Relatively little work
4. Relatively little work on evaluation and comparison of criteria

PHASE 1: TEST CASE GENERATION (TCG)

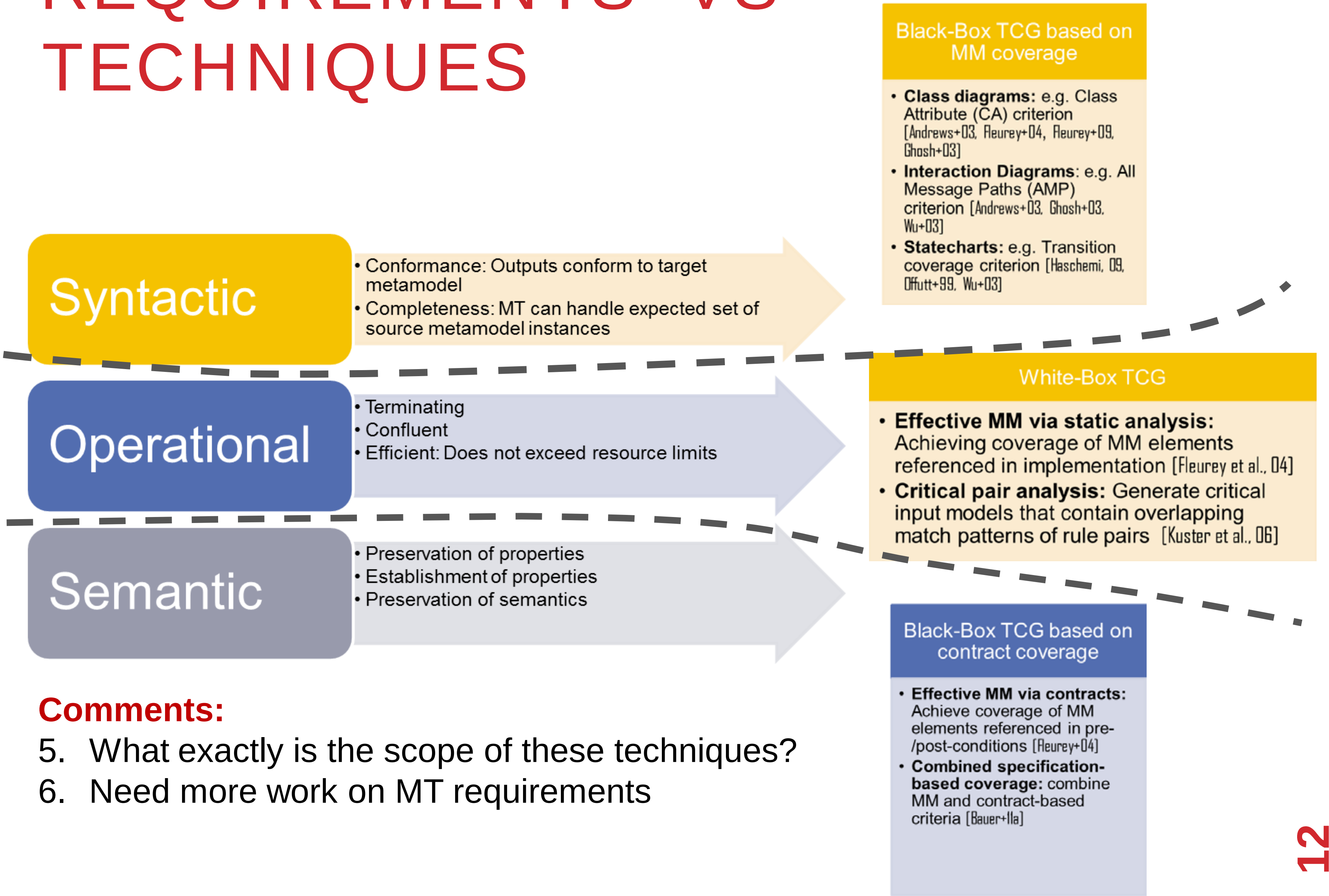
White-Box TCG (3/64)

- **Effective MM via static analysis:** Achieving coverage of MM elements referenced in implementation [Fleurey et al., 04]
- **Critical pair analysis:** Generate input models that contain overlapping match patterns of rule pairs [Kuster et al., 06]

Comments:

5. MT language dependency
6. Little researched

REQUIREMENTS VS TECHNIQUES



Syntactic

- Conformance: Outputs conform to target metamodel
- Completeness: MT can handle expected set of source metamodel instances

Operational

- Terminating
- Confluent
- Efficient: Does not exceed resource limits

Semantic

- Preservation of properties
- Establishment of properties
- Preservation of semantics

Black-Box TCG based on MM coverage

- **Class diagrams:** e.g. Class Attribute (CA) criterion [Andrews+03, Fleurey+04, Fleurey+09, Ghosh+03]
- **Interaction Diagrams**: e.g. All Message Paths (AMP) criterion [Andrews+03, Ghosh+03, Wu+03]
- **Statecharts:** e.g. Transition coverage criterion [Haschemi, 09, Offutt+99, Wu+03]

White-Box TCG

- **Effective MM via static analysis:** Achieving coverage of MM elements referenced in implementation [Fleurey et al., 04]
- **Critical pair analysis:** Generate critical input models that contain overlapping match patterns of rule pairs [Kuster et al., 06]

Black-Box TCG based on contract coverage

- **Effective MM via contracts:** Achieve coverage of MM elements referenced in pre-/post-conditions [Fleurey+04]
- **Combined specification-based coverage:** combine MM and contract-based criteria [Bauer+11a]

Comments:

5. What exactly is the scope of these techniques?
6. Need more work on MT requirements

AGENDA

- Motivation & Objective
- Background
- Model Transformation Testing Phases
- Phase 1: Test Case Generation
- Phase 2: Test Suite Assessment
- Phase 3: Oracle Construction
- Summary

PHASE 2: TEST SUITE ASSESSMENT

1. **Coverage of Adequacy Criteria (12/64)**
   - [Andrews +03], [Bauer+11a], [Bauer+11b], [Fleurey+09], [Fleurey+04], [Ghosh+03], [Haschemi09], [Kuester+06], [McQuillan+09], [McQuillan+05], [Offutt+99], [Wu+03].
2. **Mutation Analysis (5/64)**
   - Evaluates the fault revealing power of a test suite
   - *Step 1:* Injecting faults in the original transformation → mutants
   - *Step 2:* Execute original transformation & mutants using test suite
   - *Step 3:* $MutationScore = \frac{|KilledMutants|}{|Mutants| - |EquivalentMutants|}$
   - [Le Traon+06], [McQuillan+05], [Mottu+06a], [Mottu+06b], [Offutt+99]
   - Suggested **fault model**: navigation, filtering, output model creation, input model modification

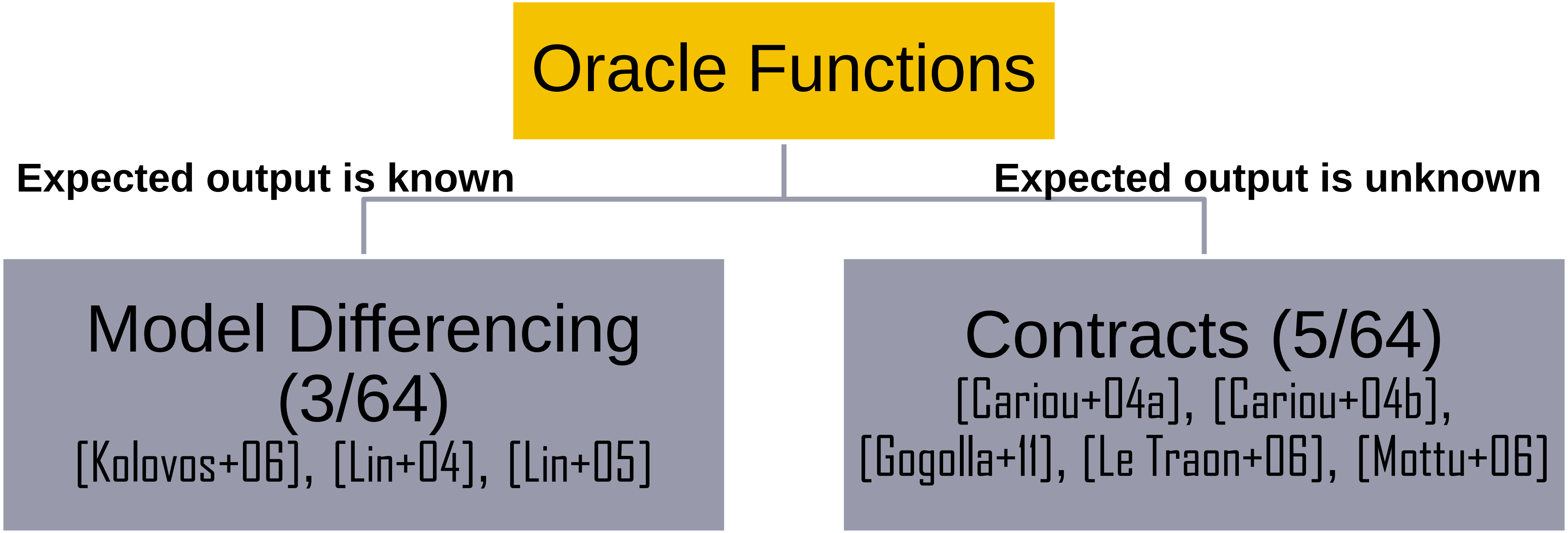
Comments:

7. Some language dependence unavoidable
8. Validity of fault model? Feedback to MT language designers, MT analysis, and MT tool developers?

AGENDA

- **Motivation & Objective**
- **Background**
- **Model Transformation Testing Phases**
- **Phase 1: Test Case Generation**
- **Phase 2: Test Suite Assessment**
- **Phase 3: Oracle Construction**
- **Summary**

PHASE 3: ORACLE CONSTRUCTION



Oracle Functions

Expected output is known

Expected output is unknown

Model Differencing (3/64)

[Kolovos+06], [Lin+04], [Lin+05]

Contracts (5/64)

[Cariou+04a], [Cariou+04b], [Gogolla+11], [Le Traon+06], [Mottu+06]

1. Syntactic (e.g., graph matching)
2. Semantic

1. OCL
2. OCL extension: e.g, tracts, transformation models
3. Other: JML

Comments:

9. Scope? All for checking semantic MT requirements (?)
10. Model differencing doesn’t have to be based on graph matching [Cordy+,12]
11. Using OCL has advantages, but some deeper understanding would be nice, too

AGENDA

- **Motivation & Objective**
- **Background**
- **Model Transformation Testing Phases**
- **Phase 1: Test Case Generation**
- **Phase 2: Test Suite Assessment**
- **Phase 3: Oracle Construction**
- **Summary**

SUMMARY

- **For details on specific techniques, see paper or TR**
  - Warning: more recent work won’t be in there
- **Comments**
  - Evaluation of test coverage criteria & their impact on kinds of faults
  - Mutation testing (fault models, evaluation, tools)
  - MT requirements
    - Specification languages/techniques, contracts
    - (Sub-) classes of MT requirements (syntactic, operational, semantic)?
    - MT testing in context of these kinds of requirements
  - More usable, publically available tools, evaluations, benchmarks, but also more foundations (not just GRS, ATL & OCL)
  - Don’t forget about source code transformation community (e.g., WCRE, ICSM, SCAM)

Thank you!