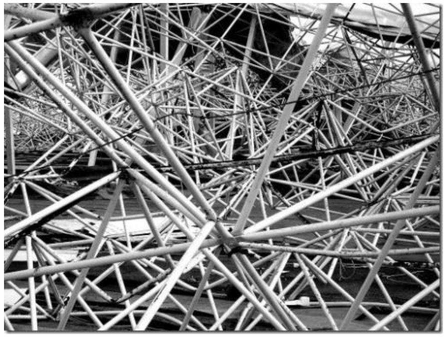
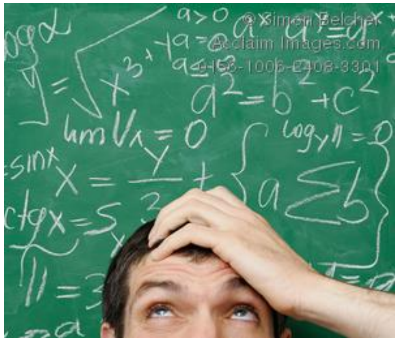
Formal Specification and Testing of Model Transformations



Antonio Vallecillo,
Universidad de Málaga
http://www.lcc.uma.es/~av



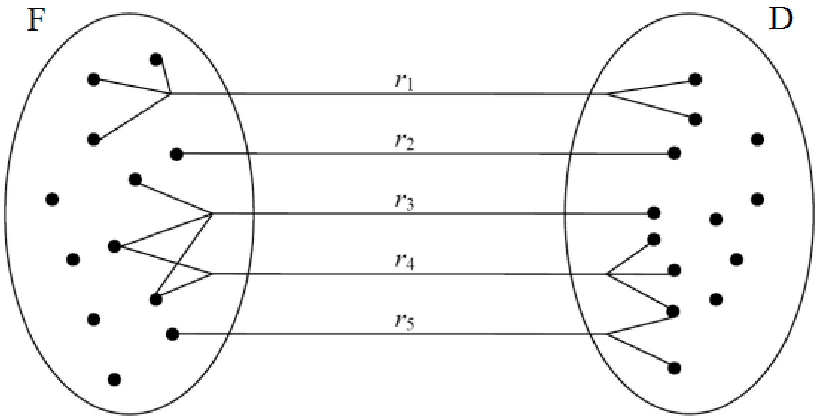
UNIVERSIDAD DE MÁLAGA

Model Transformations

A **model transformation** is

(1) *The specification of the relationship between one set of source models and one set of output models*

(2) *The process that generates such relationship*



- Model Transformations describe the relationships between views of a system, or provide bridges between views of two systems
  - Refinement relations, development relations, abstraction relations, mapping relations, …

Model Transformation Challenges

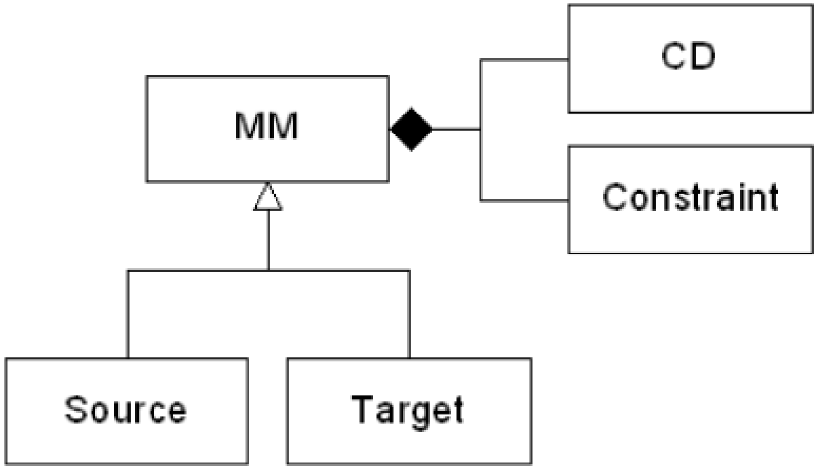
- **Complexity** of Input Models
  - Large graphs, distributed models, streaming models,…
- **Complexity** of model transformations
  - Performance? Scalability? Distribution?
  - Even the simpler transformations are complex to debug (Fam2Persons)
- **Underspecified** Metamodels
  - Consider the UML metamodel: many optional features, …
- **No** model transformation **specifications**
  - Is it possible to reuse an existing MT in a given scenario?
  - How to check the correctness of a MT?
- **Chains** of Transformations
  - Consider the MDA scenario: CIM -> PIM -> PSM -> Code
- **Higher Order** Transformations
  - Transformations may produce transformations that may produce transformations …
- No **mature tooling** for wide adoption in industrial settings

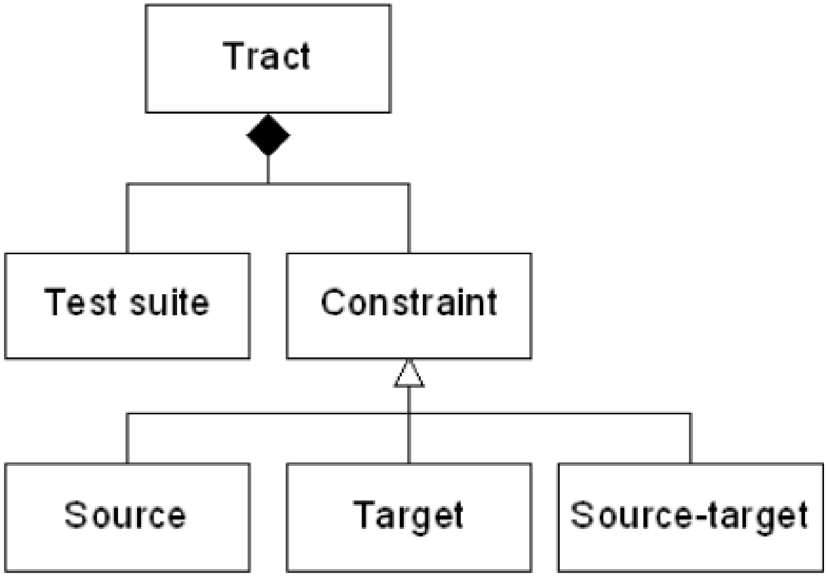
Some questions

- What should be tested on a model transformation?
  - Which are the properties to test?
  - Are they all testable?
- Should all properties to the tested treated equally?
  - BTW, which are those properties?
- Should we always aim for the best?
  - Full verification vs. Lightweight testing
  - Testing is expensive!
- What kinds of tests are required?
  - Static, Dynamic, Syntactic, Semantic,…
- How these tests can be conducted?
  - Tool support for MT testing
- How to test the quality of the tests?
  - Coverage, Usability, Automation, Repeatability, ….

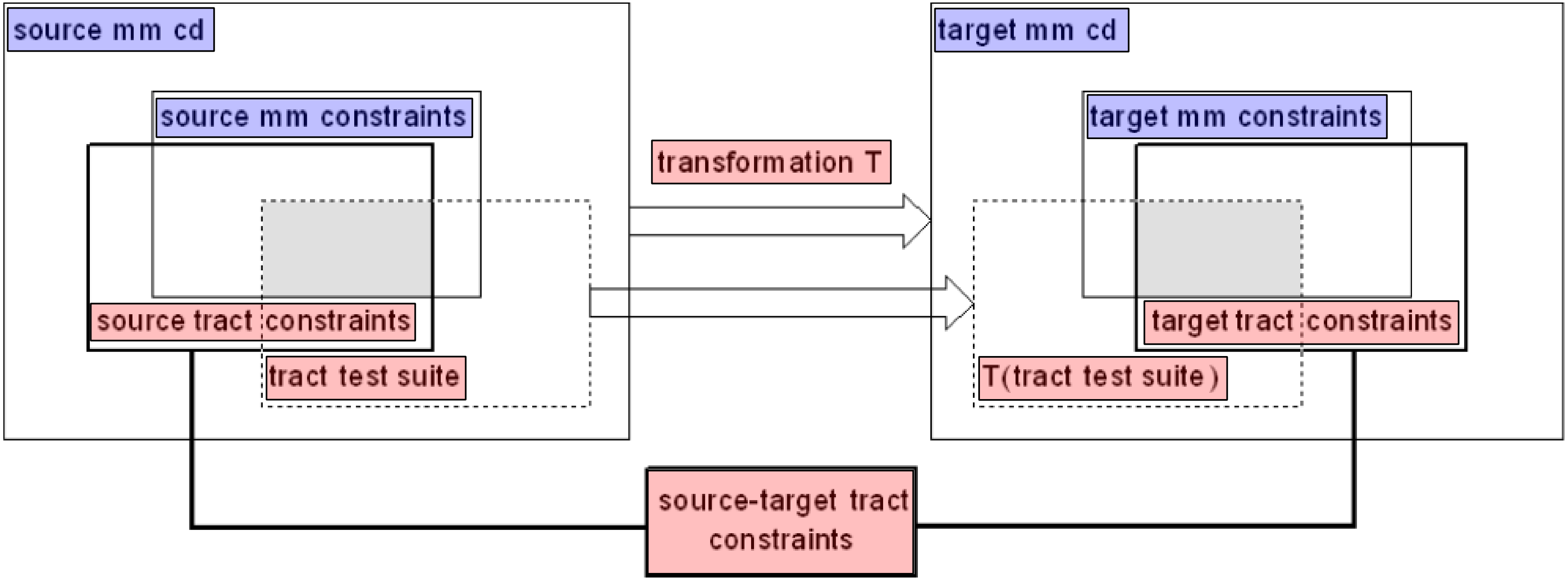
Tracts

- Blackbox Con*Tracts* for Model transformation
- A Tract defines
  - a set of **constraints on the source and target metamodels**,
  - a set of **source-target constraints**, and
  - a **tract test suite** (a collection of source models satisfying the source constraints)





The elements of a *Tract*



Lightweight & Black-box Testing of MTs

- For each tract
  - Input test suite models are automatically generated using ASSL
  - Input models are transformed into output models by the transformation under test
  - The results are checked with the USE tool against the constraints defined for the transformation
- Different tracts are defined for every transformation
  - Each one defines either a *use case* or a *special condition* or a *negative test case*
  - Test suites are key to Tracts (coverage, repeatability, etc.)
- Some challenges and issues for tracts (w.r.t. this WS)
  - **Test Suites** can be very effective (if well-defined)
  - How to define them to improve: Effectiveness? Coverage?
  - How to combine tracts (esp. test suites) with Mutation Testing?