Towards Tracking "Guilty" Transformation Rules

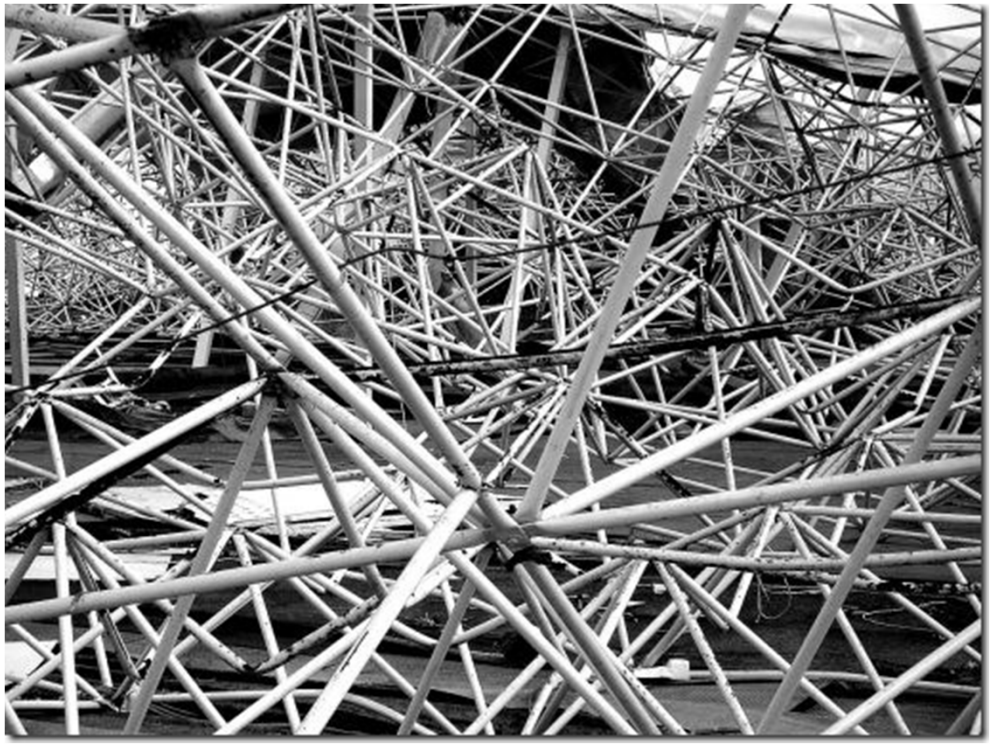
Loli Burgueño, Manuel Wimmer, and **Antonio Vallecillo**
{loli, mw, **av**}@**lcc.uma.es**

AMT Workshop @ MoDELS'12, Innsbruck, October 2nd 2012

This work has been partly funded by the Austrian Science Fund (FWF) under grant J 3159-N23.

Motivation

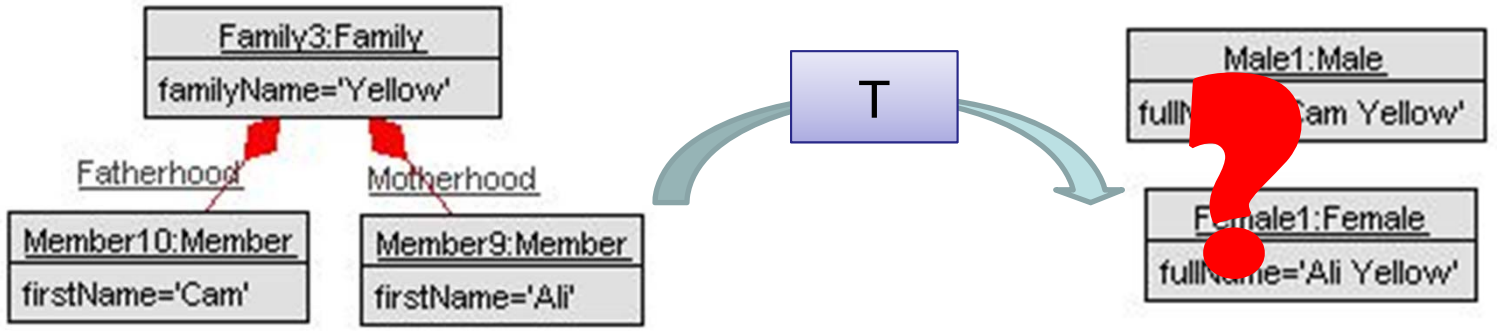
Although specified at a very high level of abstraction, **model transformations are becoming very complex** as the complexity of the relations they are able to describe grows...



Motivation for (Con)Tracts

- In general it is **difficult** and **expensive** (time and computational complexity-wise) to validate in full the correctness of a model transformation (MT).

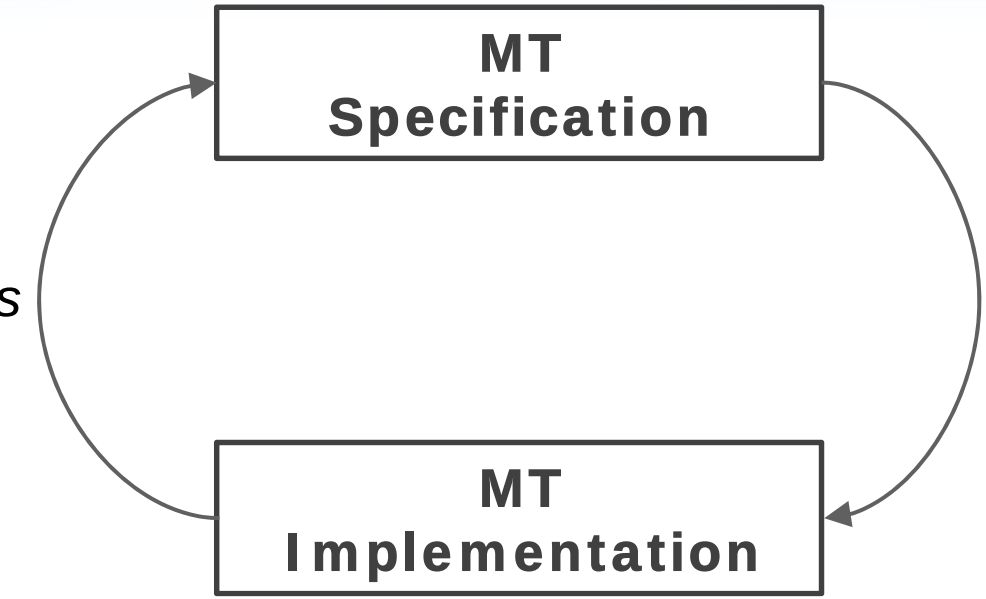
- **Tracts** offer a *cost-effective* MT testing approach, which is a particularization of the concept of **MT Contract**.



Contracts as Specifications

Specification: A document that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a system or component… [IEEE Standard Computer Dictionary]

Tracts, PaMoMo, …



What?

vs.

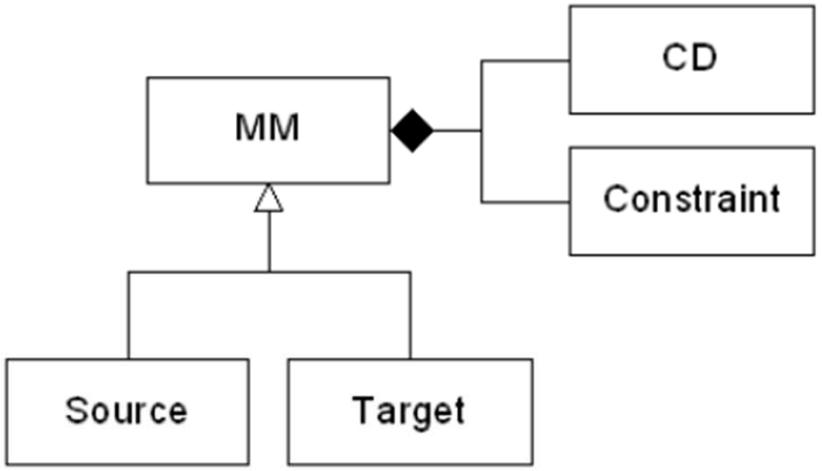
How?

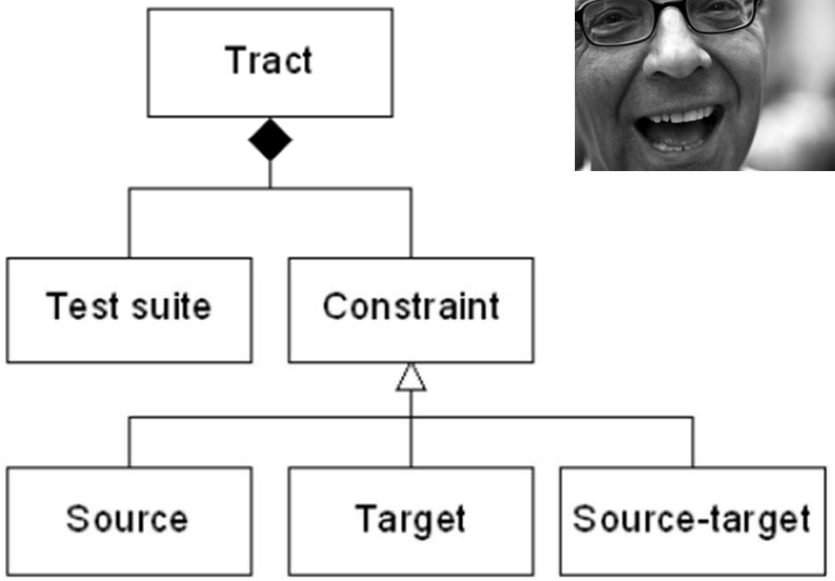
ATL, ETL, QVT-R, QVT-O, RubyTL, TGG, GT, …

Implementation:
(1) The process of translating a design into hardware components, software components, or both.
(2) The result of the process in **(1)** [IEEE Standard Computer Dictionary]

What's in a Tract?

- 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)

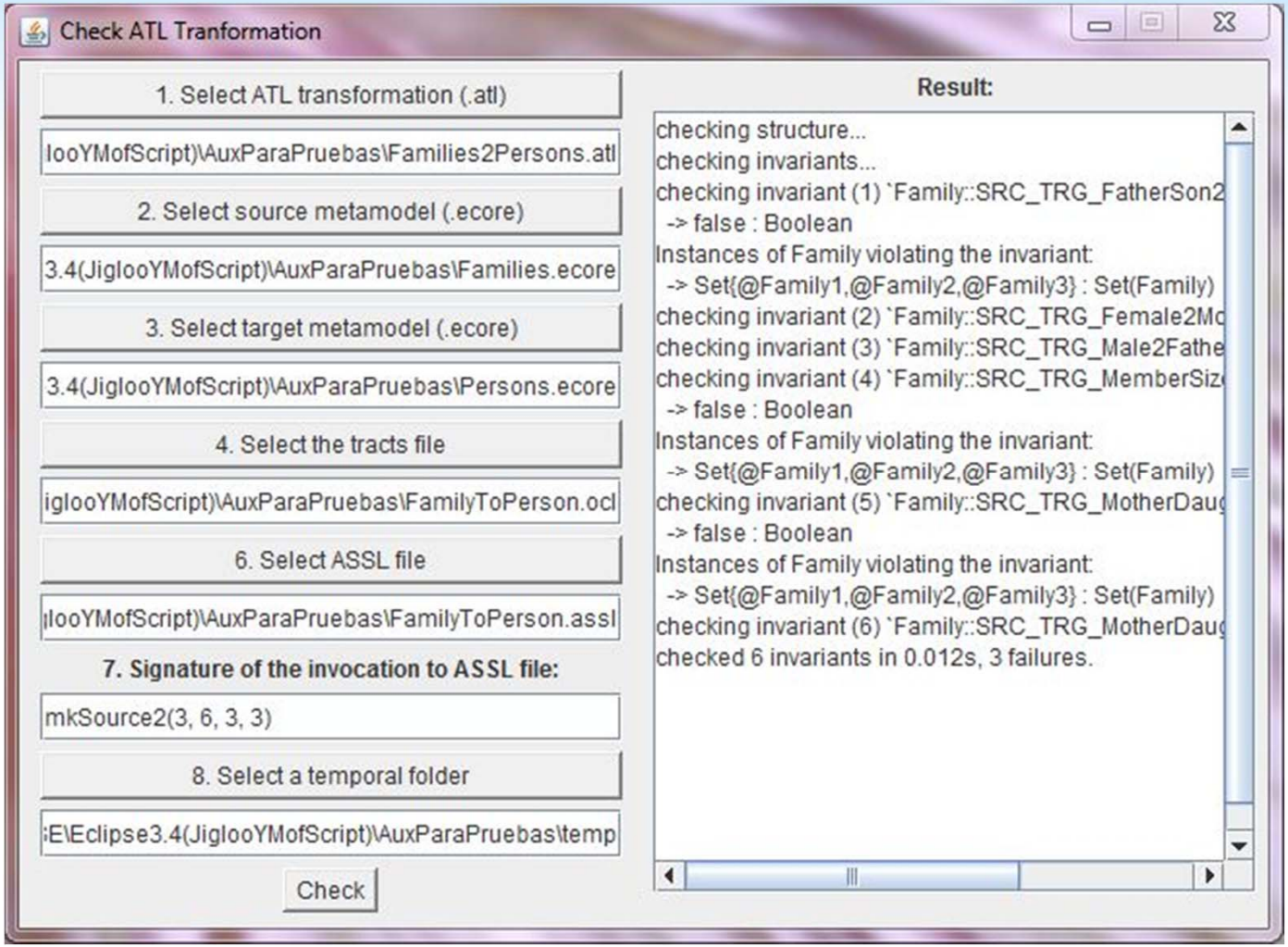




Black-box testing of MTs

- Different tracts are defined for every transformation
  - Each one defines either a use case (scenario) or a special condition or a negative test
  - They are written in OCL and refer to the SMM, TMM and the relationship between the two
- For each tract
  - Input test suite models are automatically generated using ASSL (A Snapshot Sequence Language)
  - 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

TractsTool Screenshot



Consequences

- The specification and implementation of a model transformation are completely separated

- **Advantages** 
  - Several implementations for one specification possible
  - Specification language independent from implementation language
  - Implementations are independent of specification

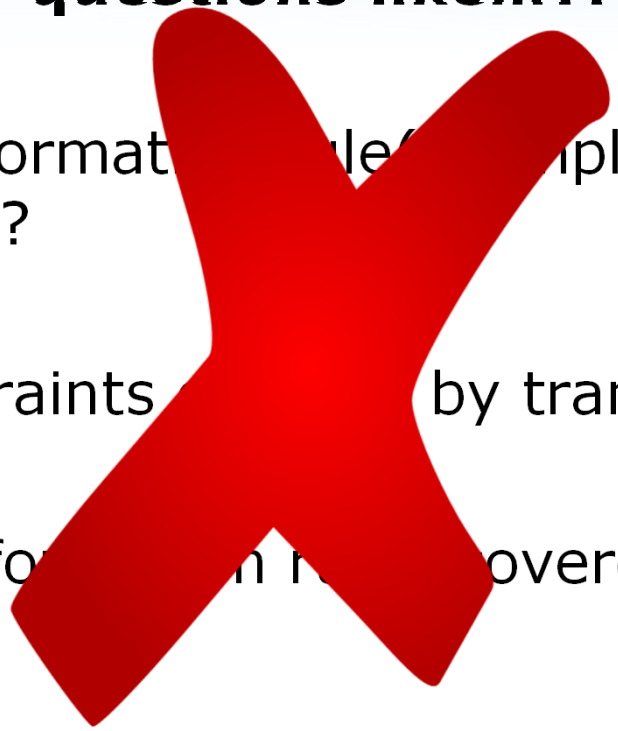
- **Disadvantages** 
  - Relationships between contracts and transformation rules not explicitly given
  - *Artifacts are of different nature and live in different worlds!*
  - Tracing between contracts and transformation rules not possible

Consequences

- **Can we answer questions like…?:**
  - Which transformation rule(s) implement(s) which constraint(s)?
  - Are all constraints covered by transformation rules?
  - Are all transformation rules covered by constraints?

Consequences

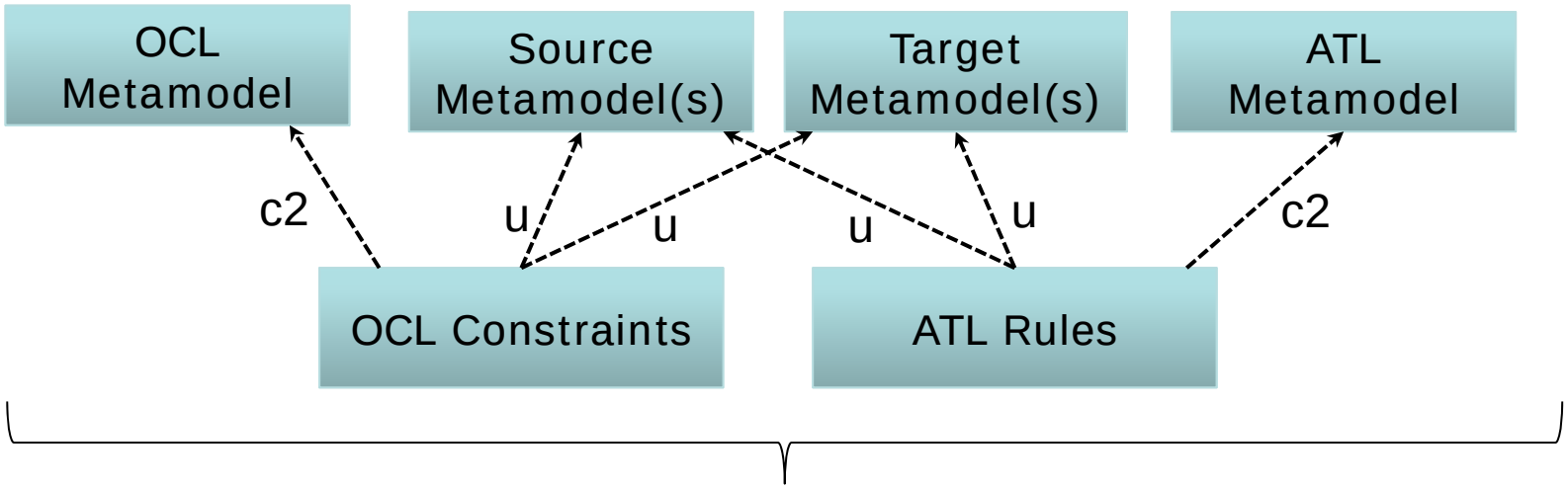
- **Can we answer questions like...?:**
  - Which transformat[illegible]le[illegible]plement(s) which constraint(s)?
  - Are all constraints [illegible] by transformation rules?
  - Are all transfo[illegible]n r[illegible]overed by constraints?



Make the relationship between specifications and implementations explicit!

Matching Constraints and Rules

- Heterogeneities
  - Programming paradigm
  - Granularity
- Common denominator
  - Source and Target Metamodels (i.e., metamodel elements they both refer to)



Base matching function on used metamodel element overlaps

Matching Constraints and Rules

Tract

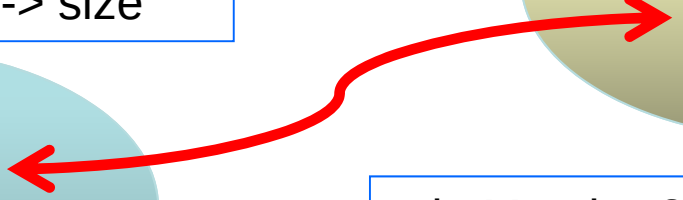
```
Member . allInstances -> size
=
Person . allInstances -> size
```

Types used by Tract

Member
Person

Types used by Rule

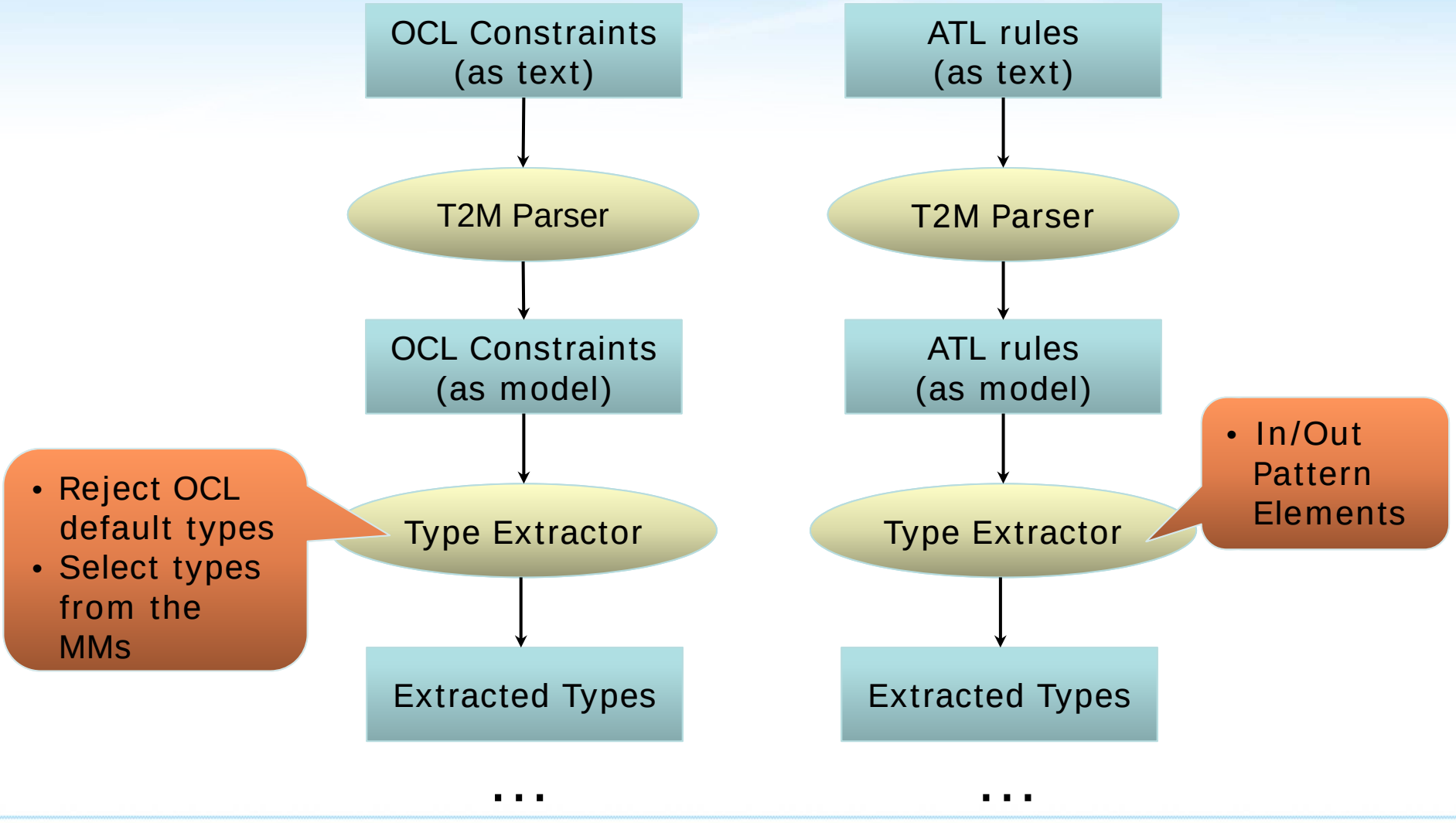
Female
Member



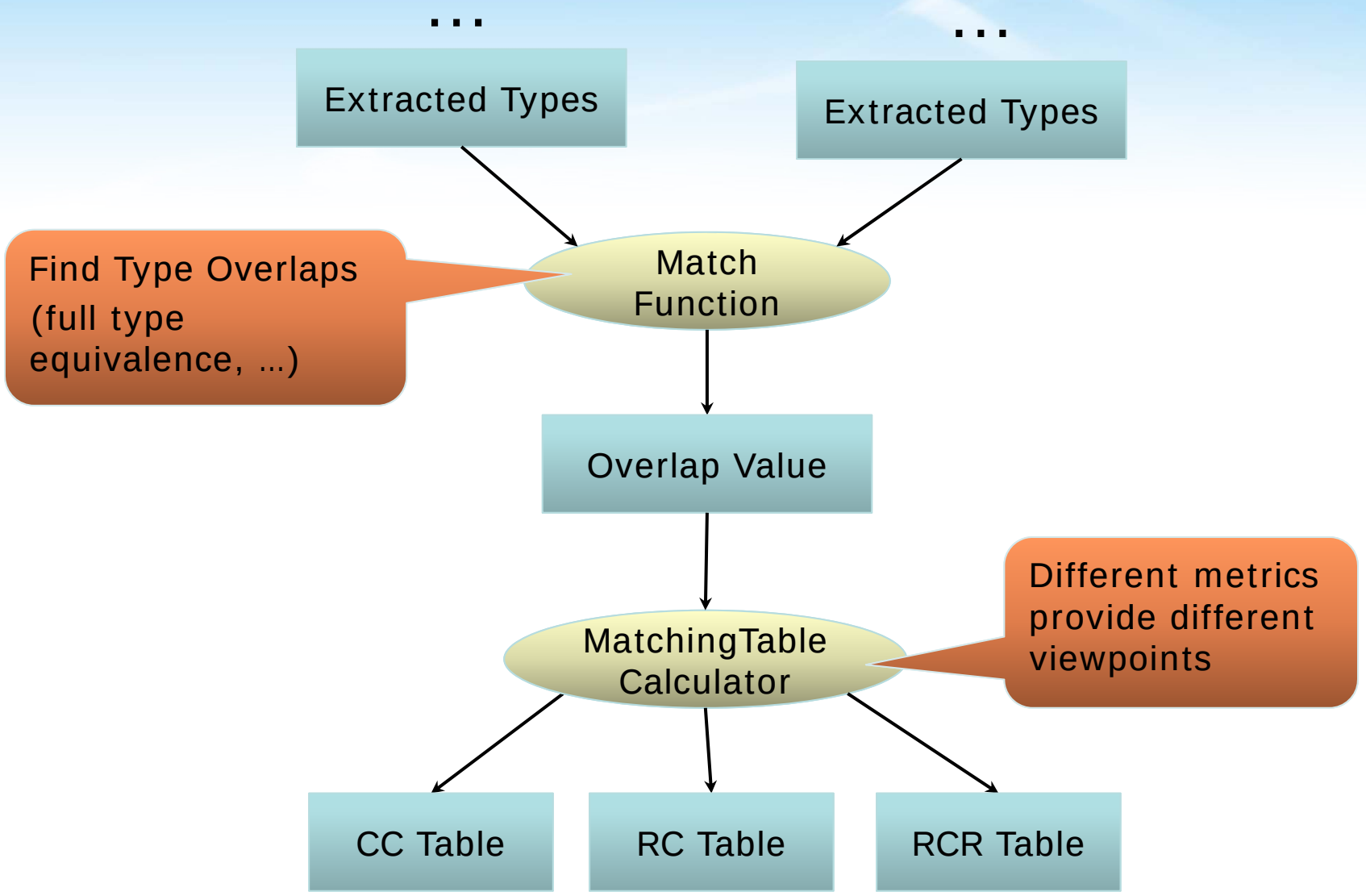
```
rule Member2Female {
  from
    s: Families ! Member (s. isFemale ())
  to
    t: Persons ! Female (
            fullName <- s. firstName + ' ' + s.
               familyName
    )
}
```

ATL Rule

Pre-Matching Step: Extract types



Matching & Post-Matching Step



Current Metrics

- **CC**: coverage for constraint *i* by rule *j*

$$CC_{i,j} = \frac{|C_i \cap R_j|}{|C_i|}$$

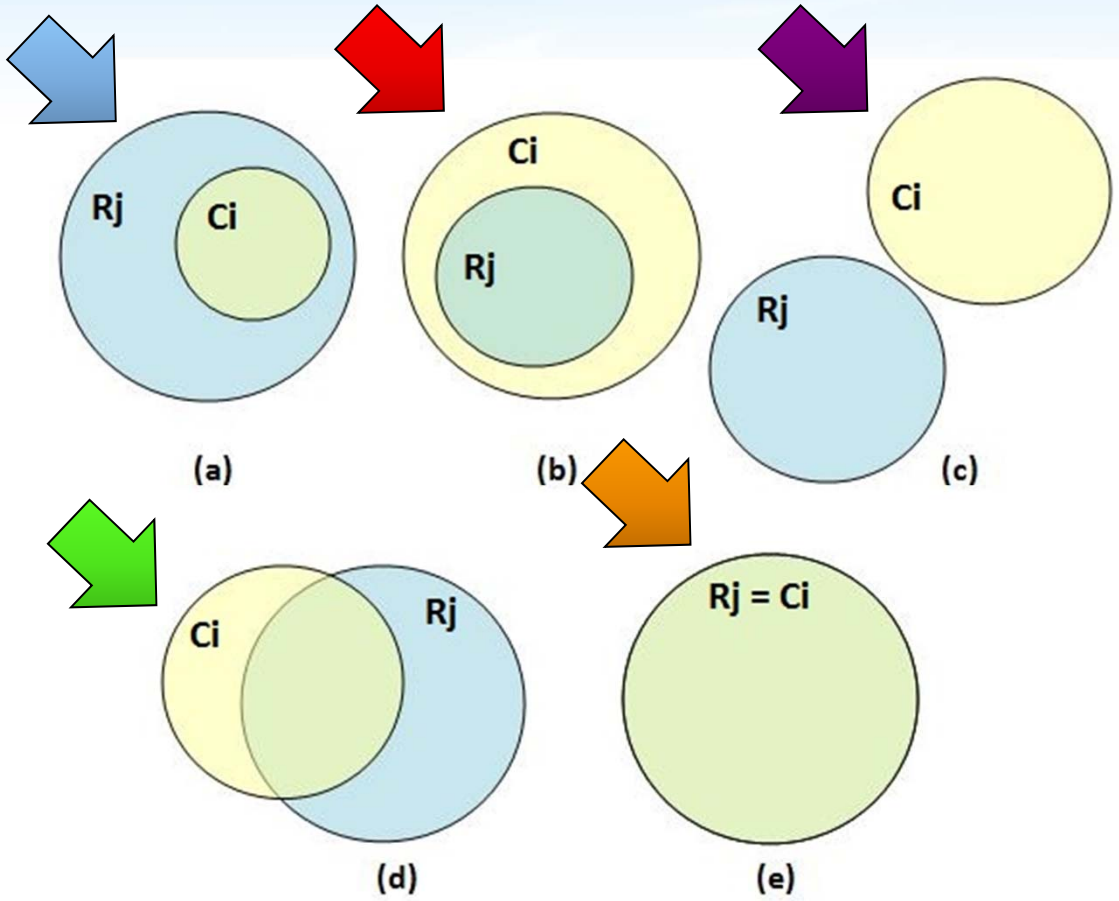
- **RC**: coverage for rule *j* by constraint *i*

$$RC_{i,j} = \frac{|C_i \cap R_j|}{|R_j|}$$

- **RCR**: relatedness of constraint *i* and rule *j*, without a specific direction for interpreting the values

$$RCR_{i,j} = \frac{|C_i \cap R_j|}{|C_i \cup R_j|}$$

Five possible situations



	CC		RC		RCR	
	R1	R2	R1	R2	R1	R2
C1	1	0.5	0.5	0.5	0.33	0.33
C2	0.33	0.66	0.5	1	0.25	0.66
C3	0.33	0.66	0.5	1	0.25	0.66
C4	1	0.33	1	0.5	1	0.25
C5	0.33	0.66	0.5	1	0.25	0.66
C6	0.66	0.33	1	0.5	0.66	0.25
C7	0.5	0.5	0.5	0.5	0.33	0.33
C8	0.0	0.0	0.0	0.0	0.0	0.33

How to Interpret the Matching Tables

CC

	R1	R2	R3	R4	R5	R6	R7	R8
C1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C2	0.5	0.0	0.5	0.25	0.25	0.25	0.25	0.25
C3	0.33	0.0	0.33	0.5	0.33	0.33	0.33	0.33
C4	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
C5	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C6	0.0	0.0	1.0	0.5	0.5	0.5	0.5	0.5
C7	0.0	0.0	0.0	1.0	0.5	0.5	0.5	0.5
C8	0.28	0.0	0.28	0.42	0.42	0.28	0.28	0.28
C9	0.25	0.0	0.25	0.37	0.25	0.37	0.37	0.25
C10	0.25	0.0	0.25	0.37	0.25	0.25	0.25	0.37

RCR

	R1	R2	R3	R4	R5	R6	R7	R8
C1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C2	0.5	0.0	0.5	0.16	0.16	0.16	0.16	0.16
C3	0.33	0.0	0.33	0.5	0.28	0.28	0.28	0.28
C4	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
C5	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C6	0.0	0.0	1.0	0.25	0.25	0.25	0.25	0.25
C7	0.0	0.0	0.0	0.66	0.25	0.25	0.25	0.25
C8	0.28	0.0	0.28	0.42	0.42	0.25	0.25	0.25
C9	0.25	0.0	0.25	0.37	0.22	0.37	0.5	0.22
C10	0.25	0.0	0.25	0.37	0.22	0.22	0.22	0.37

	R1	R2	R3	R4	R5	R6	R7	R8
C1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C2	1.0	0.0	1.0	0.33	0.33	0.33	0.33	0.33
C3	1.0	0.0	1.0	1.0	0.66	0.66	0.66	0.66
C4	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
C5	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C6	0.0	0.0	1.0	0.33	0.33	0.33	0.33	0.33
C7	0.0	0.0	0.0	0.66	0.33	0.33	0.33	0.33
C8	1.0	0.0	1.0	1.0	1.0	0.66	0.66	0.66
C9	1.0	0.0	1.0	1.0	0.66	1.0	1.0	0.66
C10	1.0	0.0	1.0	1.0	0.66	0.66	0.66	1.0

RC

Next steps

- Properties of alignment
  - Reason about design guidelines based on matching tables
- Refinements of alignments
  - Inheritance between rules, lazy rule calls, etc.
- Dynamic approach
  - Based on traces

TractsTool

http://atenea.lcc.uma.es/index.php/Main_Page/Resources/Tracts

Tracts2ATL

http://atenea.lcc.uma.es/index.php/Main_Page/Resources/Tracts-ATL

Contact

Loli Burgueño, Manuel Wimmer, and **Antonio Vallecillo**

{loli, mw, **av**}@**lcc.uma.es**