On the Realization of TractsTool

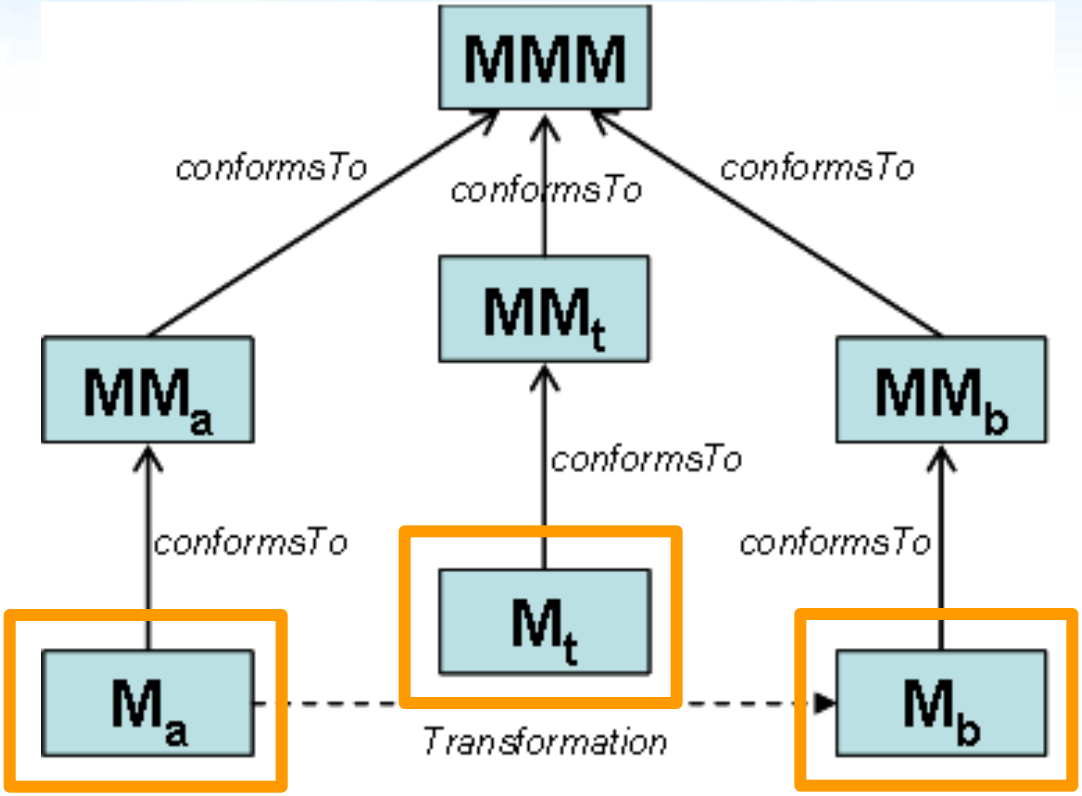
Loli Burgueño, Manuel Wimmer,
Javier Troya, and Antonio Vallecillo

University of Málaga & Vienna University of Technology

TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

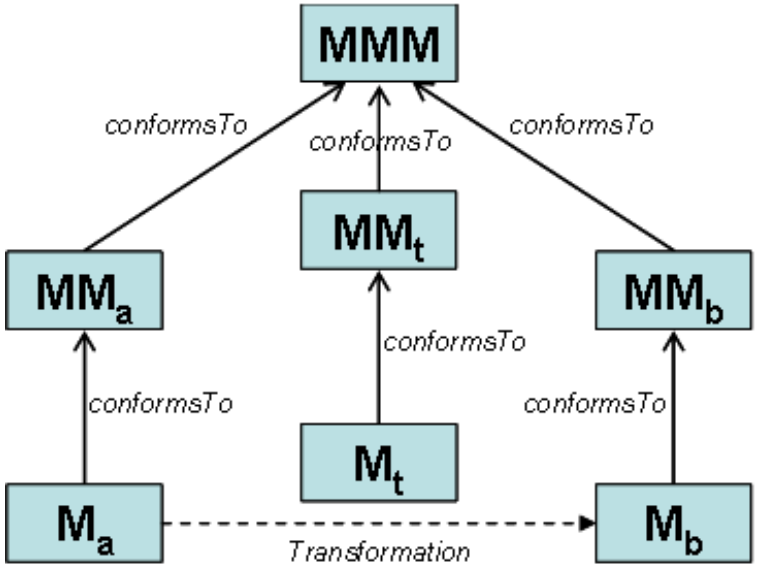
UNIVERSIDAD DE MÁLAGA

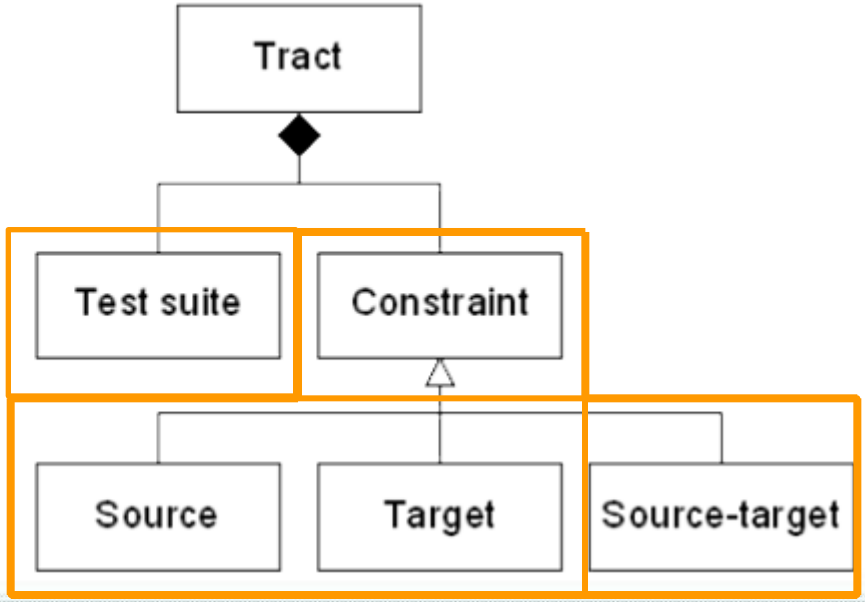
Testing Model Transformations



Tracts

- 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)





Tracts

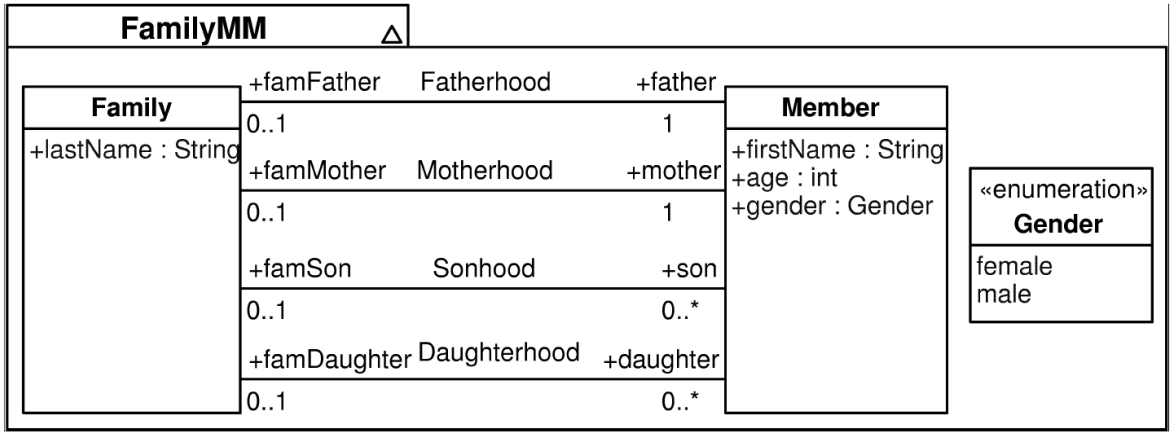
- **Principles**
  - Reuse existing languages and techniques of MDE
    - Contract specification
    - Test model generation
  - Light-weight, black-box testing approach
    - Partial specification
    - Transformation language & implementation independent
  - Support for various transformation kinds
    - Model-to-model
    - Model-to-text
    - Text-to-model
  - Open design
    - Several testing scenarios
    - Integration with different transformation tools

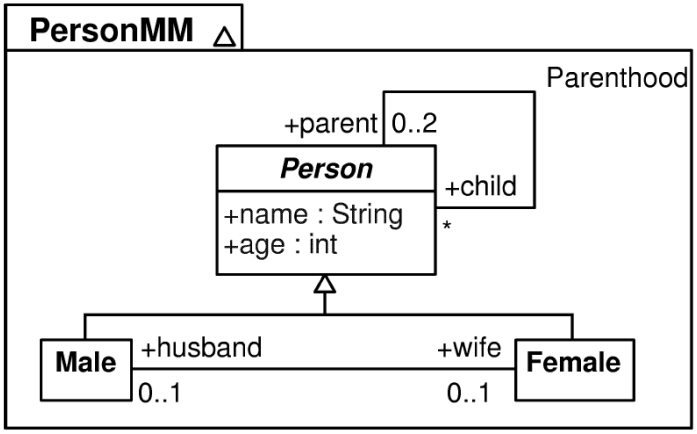
TractsTool's Ecosystem

- EMF (Eclipse Modeling Framework)
  - Metamodeling (Ecore-based) & modeling support
- USE Tool (UML-based Specification Environment)
  - Analyzes model structure (with respect to its metamodel & OCL constraints)
- OCL (Object Constraint Language)
  - Allows to define model constraints
- ASSL (A Snapshot Sequence Language)
  - Imperative programming language with features for randomly choosing attribute values or association ends
  - Allows to automatically generate models
  - ASSL supports backtracking for finding models with particular properties
- ATL (ATLAS Transformation Language)
  - Hybrid language to define model transformations

Tracts for the Families2Persons MT

- **Source Metamodel: Family**
- **Target Metamodel: Person**





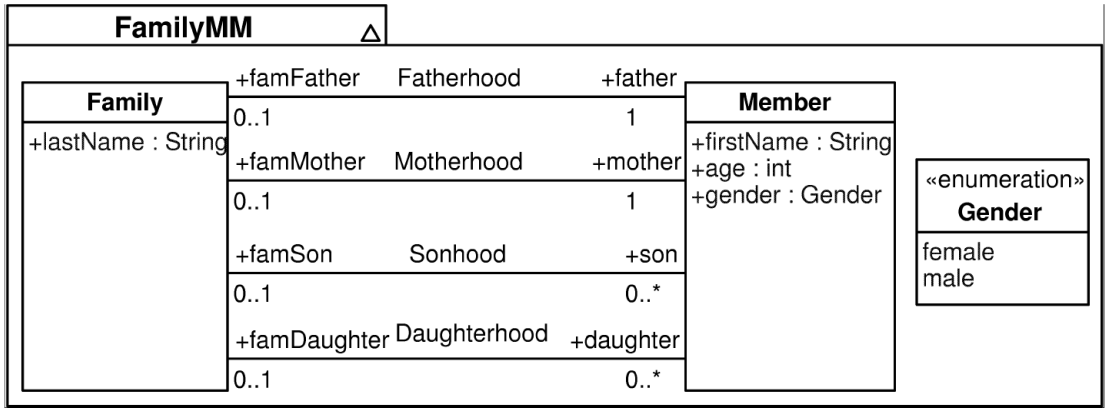
Example of *Tract*: "Members only"

Tract: Members only

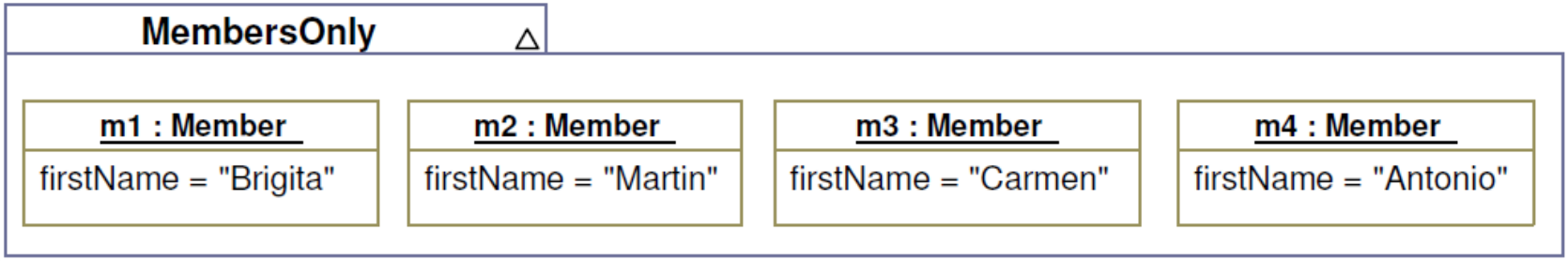
- Interested in families consisting only of members

Precondition

```
inv SCR_MembersOnly:
    src_Member.allInstances -> forAll (m |
        m.famFather->size() +
        m.famMother->size() +
        m.famSon->size() +
        m.famDaughter->size()  = 0)
```



Test Source Model

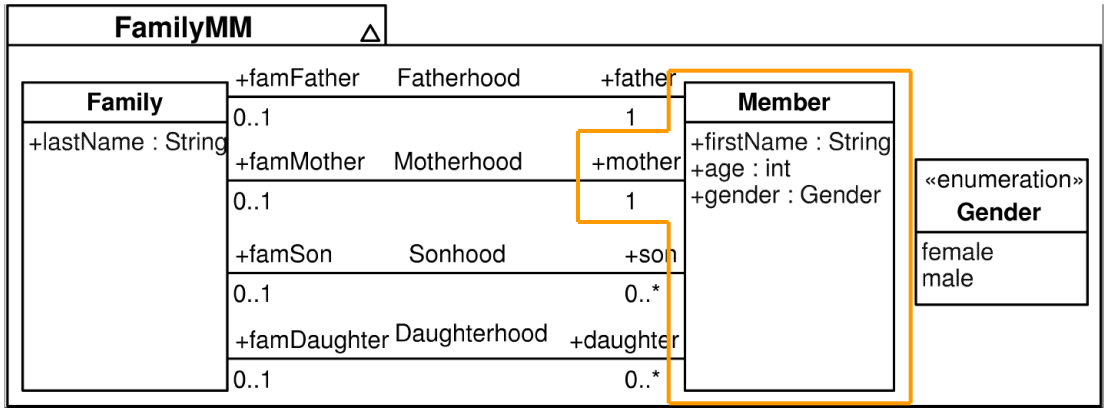


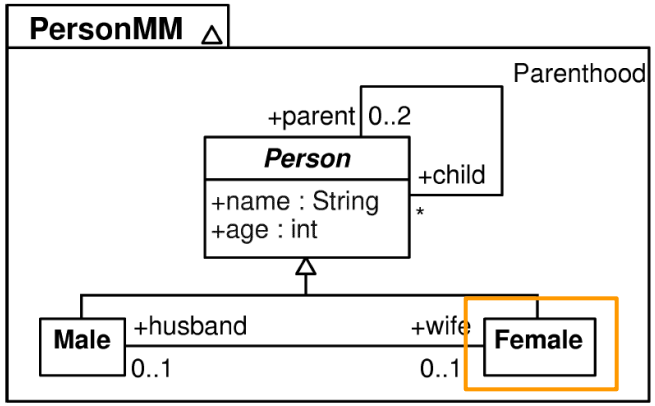
Example of *Tract*: "Mother2Female"

- **Tract: Mother2Female –** a female has to be created from each mother

- **Constraint** on the **source-target relationship**

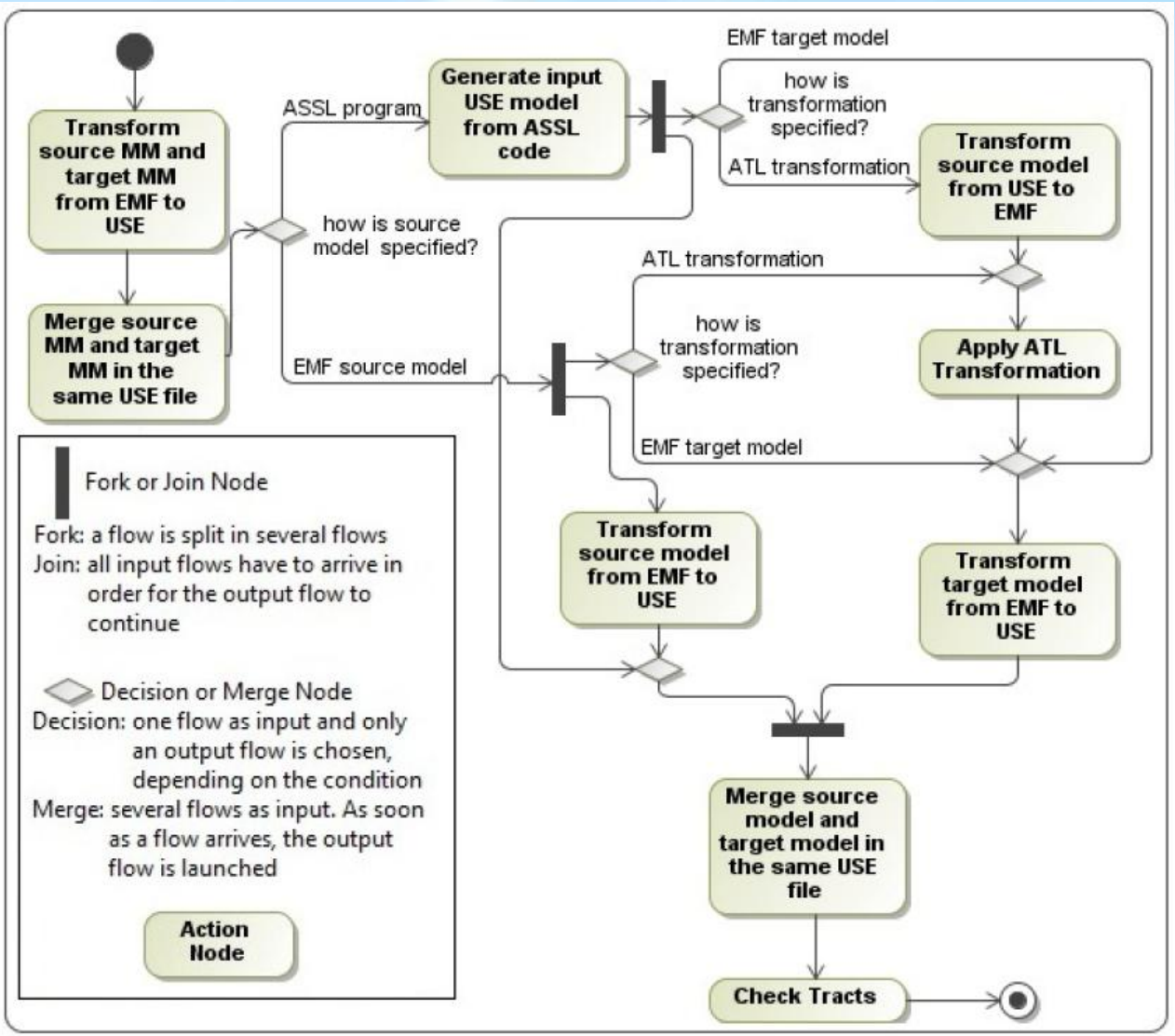
```
inv SRC_TRG_Mother2Female:
    src_Family.allInstances -> forAll (fam|trg_Female.allInstances-> exists(m|
     fam.mother.firstName.concat(' ').concat(fam.lastName) = m.name))
```



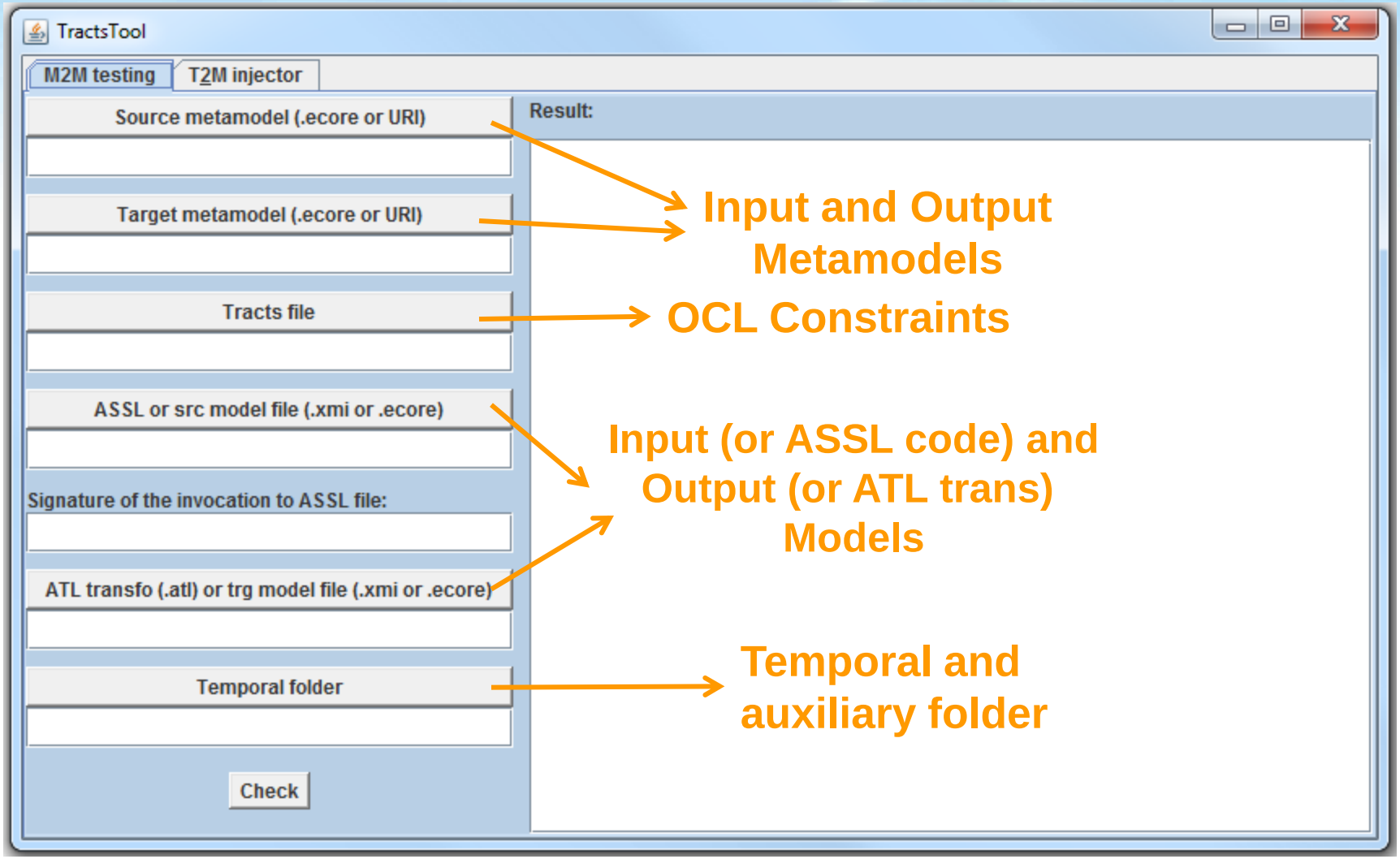


Internal Model Transformation Chain (M2M transformations)

- **Input Models**
  - *Intensional*
    - ASSL program
  - *Extensional*
    - EMF models
    - Text artefacts
- **Transformation**
  - *Intensional*
    - ATL transformation
  - *Extensional*
    - EMF models
    - Text artefacts

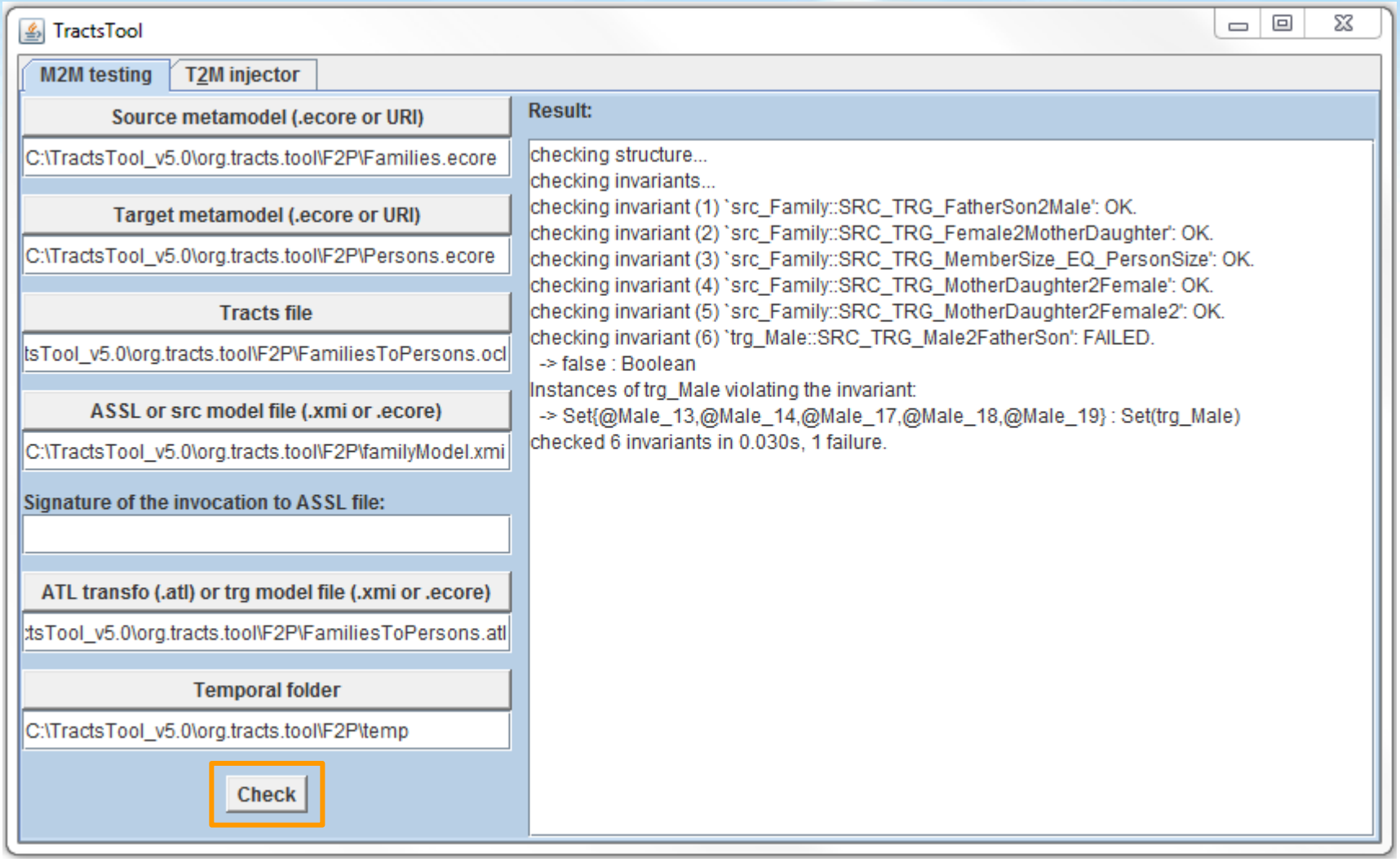


TractsTool



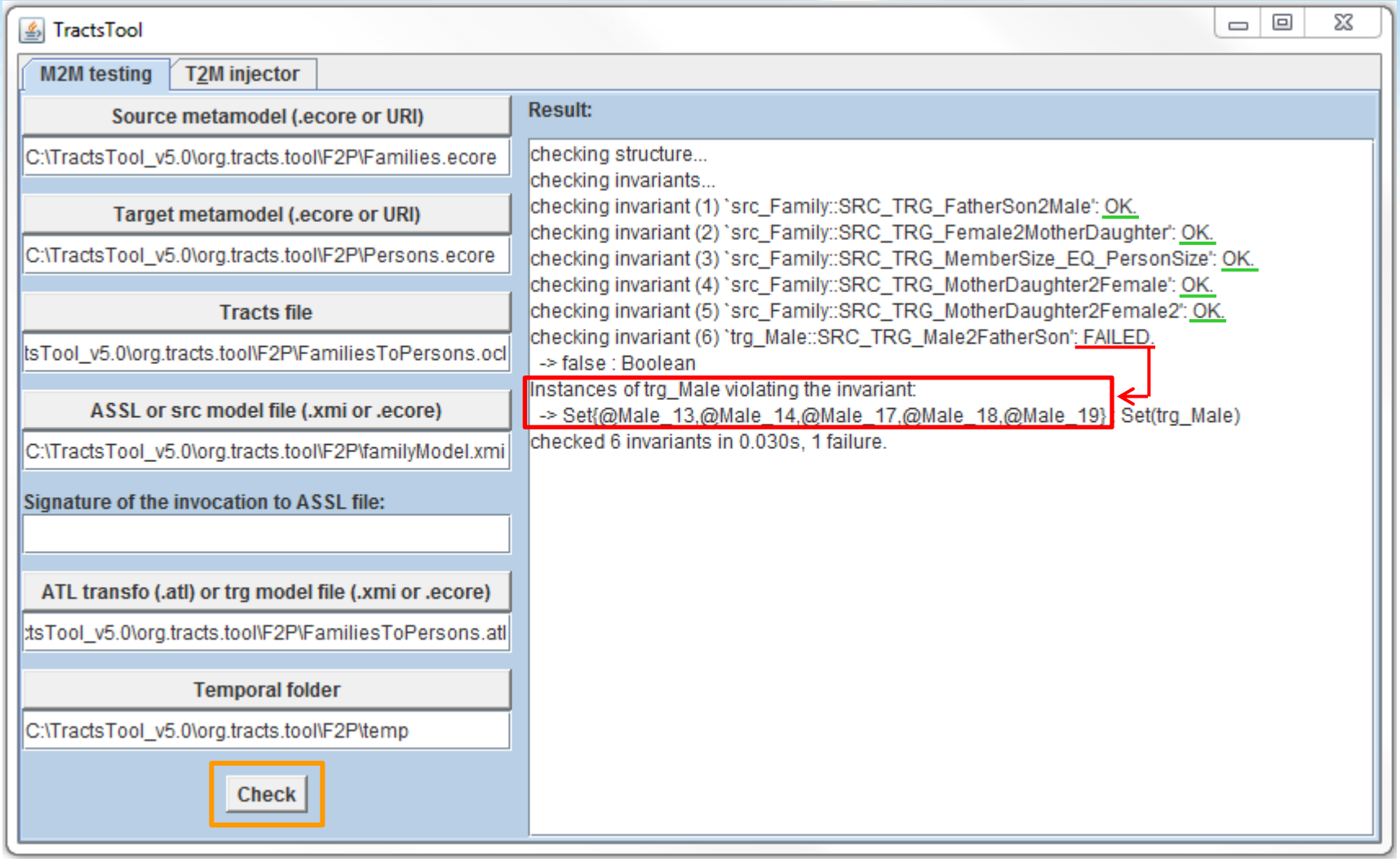
http://atenea.lcc.uma.es/index.php/Main_Page/Resources/Tracts

TractsTool



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

TractsTool

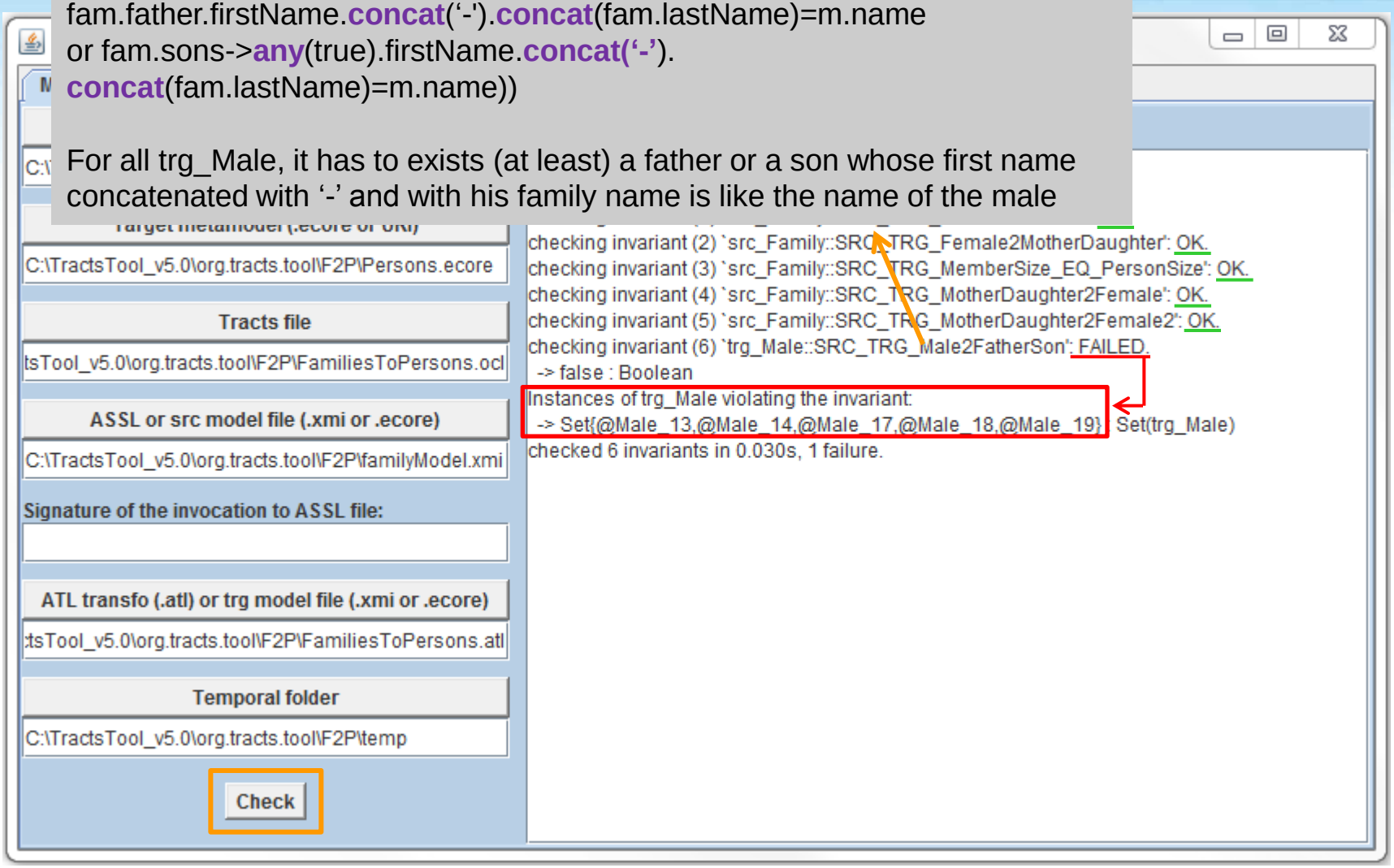


http://atenea.lcc.uma.es/index.php/Main_Page/Resources/Tracts

actsTool

```
context trg_Male inv SRC_TRG_Male2FatherSon:
trg_Male.allInstances->forAll(m|src_Family.allInstances->exists(fam |
fam.father.firstName.concat('-').concat(fam.lastName)=m.name
or fam.sons->any(true).firstName.concat('-').
concat(fam.lastName)=m.name))
```

For all trg_Male, it has to exists (at least) a father or a son whose first name concatenated with '-' and with his family name is like the name of the male

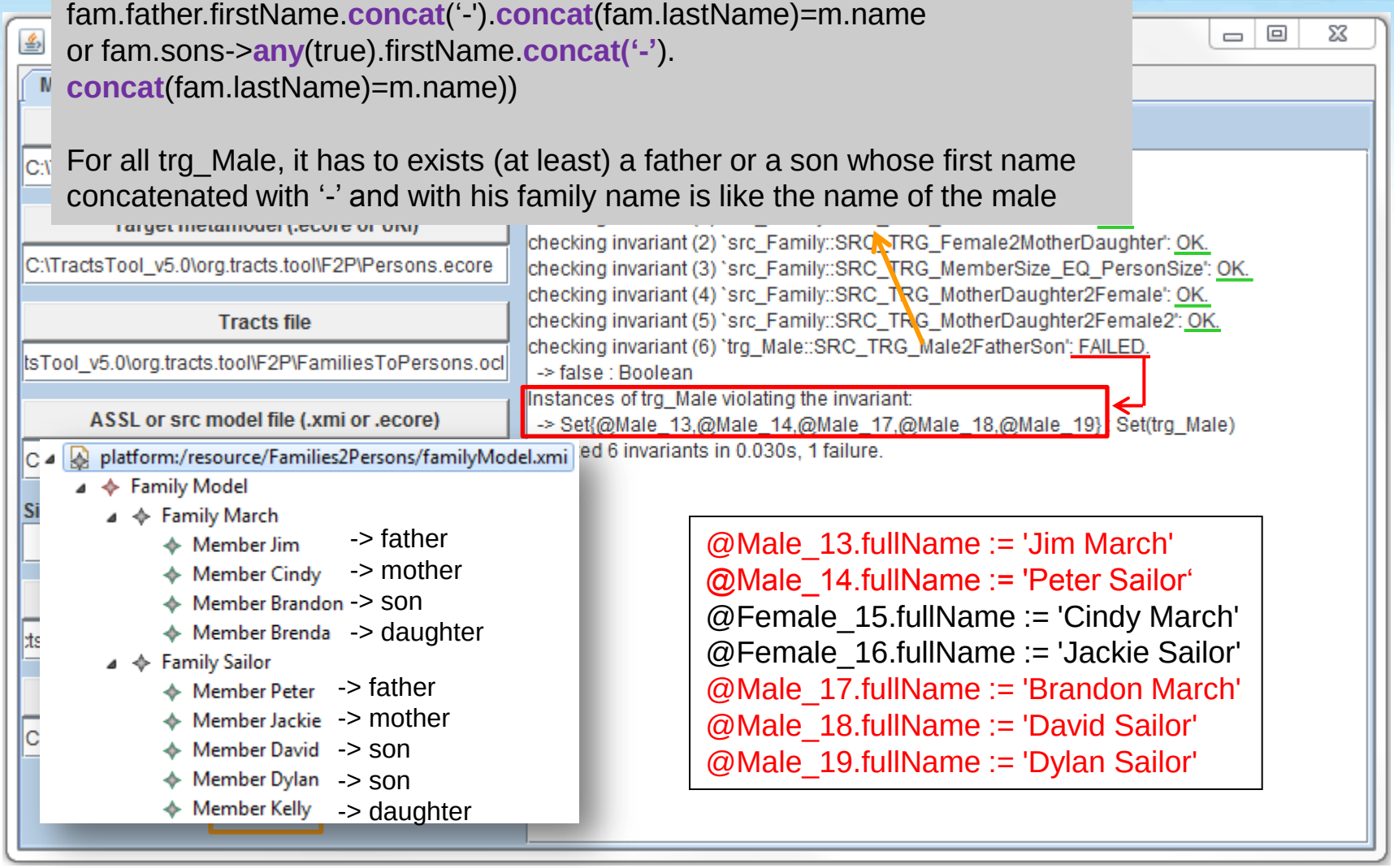


http://atenea.lcc.uma.es/index.php/Main_Page/Resources/Tracts

actsTool

```
context trg_Male inv SRC_TRG_Male2FatherSon:
trg_Male.allInstances->forAll(m|src_Family.allInstances->exists(fam |
fam.father.firstName.concat('-').concat(fam.lastName)=m.name
or fam.sons->any(true).firstName.concat('-').
concat(fam.lastName)=m.name))
```

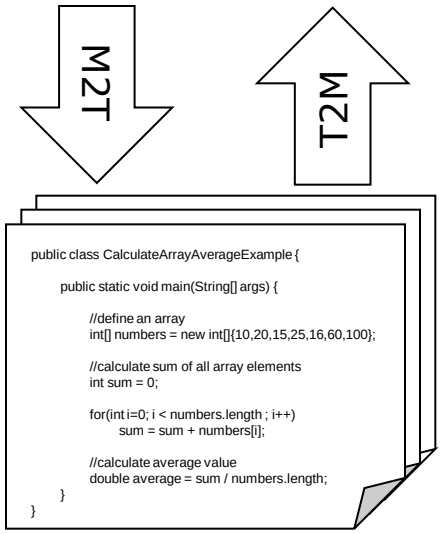
For all trg_Male, it has to exists (at least) a father or a son whose first name concatenated with '-' and with his family name is like the name of the male

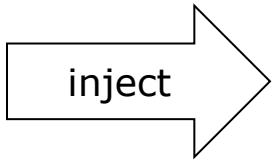


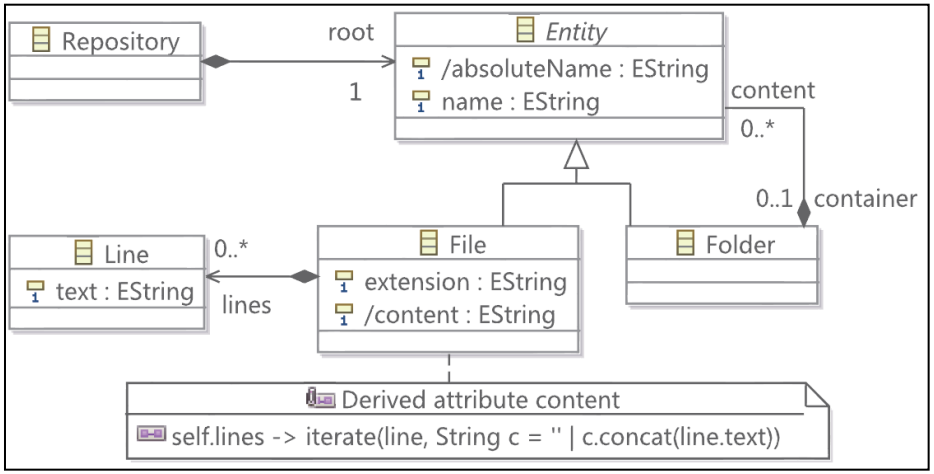
http://atenea.lcc.uma.es/index.php/Main_Page/Resources/Tracts

Dealing with M2T and T2M Transformations

- Testing problem is transformed to M2M testing problem
  - Text artefacts are represented as models
  - Then TractsTool can be simply reused







Next steps

- Generate ASSL code automatically
  - Metamodel Coverage/Transformation Coverage
- Enhance output of TractsTool
  - Text report → diagnostic model
- Allow transformation written in other languages to be tested

Thanks!

The Tracts Team:
Loli Burgueño, Lars Hamann, Martin Gogolla,
Antonio Vallecillo, Manuel Wimmer, and Javier Troya
Universidad de Málaga/Universität Bremen/Technische Universität Wien

The TractsTool:

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

More information on Tracts:

- Loli Burgueño, Manuel Wimmer, Antonio Vallecillo. "Towards Tracking Guilty Transformation Rules". In AMT@MODELS 2012: 27-32
- Antonio Vallecillo, Martin Gogolla, Loli Burgueño, Manuel Wimmer, Lars Hamann: Formal Specification and Testing of Model Transformations. In SFM 2012: 399-437
- Antonio Vallecillo, Martin Gogolla: Typing Model Transformations Using Tracts. In ICMT 2012: 56-71
- Martin Gogolla, Antonio Vallecillo: *Tract*able Model Transformation Testing. In ECMFA 2011: 221-235

Acknowledgements:

This work has been supported by Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech

Universität Bremen

TU WIEN TECHNISCHE UNIVERSITÄT WIEN Vienna University of Technology

UNIVERSIDAD DE MÁLAGA