# SyVOLT: Full Model Transformation Verification Using Contracts

Levi Lúcio, **Bentley James Oakes**, Claudio Gomes, Gehan Selim, Juergen Dingel, James R. Cordy, Hans Vangheluwe

McGill University, Canada
University of Antwerp, Belgium
Queen's University, Canada

September 30, 2015

# Motivation and Overview

- Model transformations are at the heart of model-driven engineering
- Want to verify correctness for transformation specifications
  - Verify visual contracts
  - Identify combinations of rules where contracts hold or not
- Objective: Verification for all input models
  - Input independence

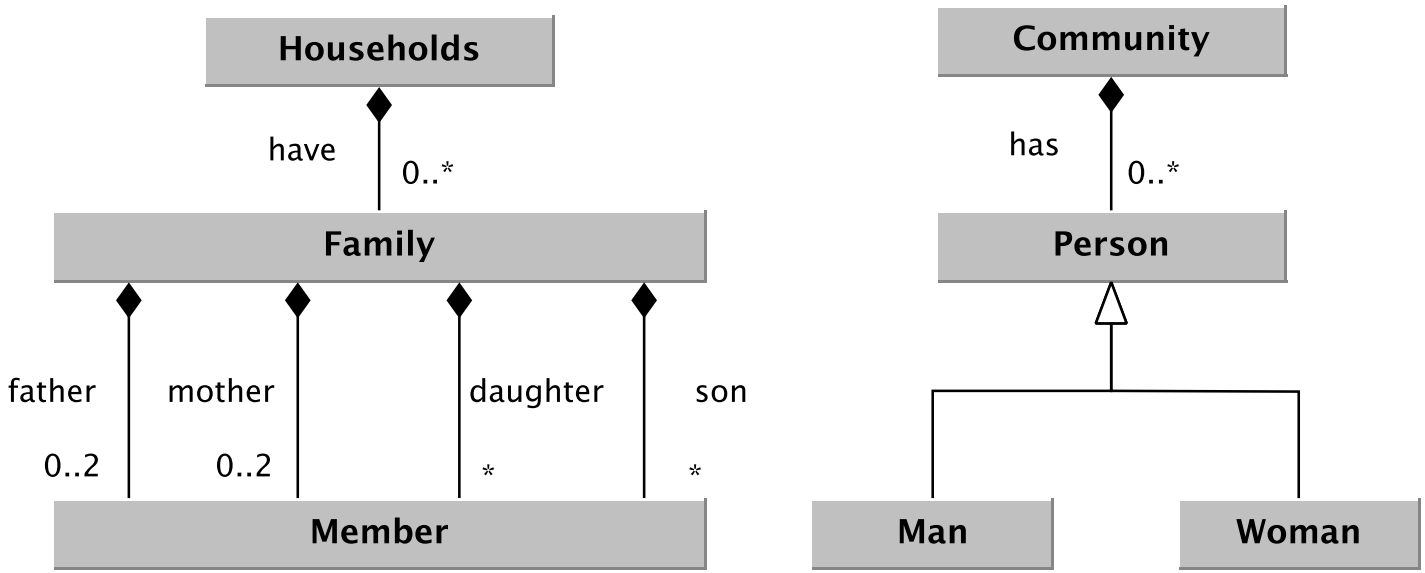
# DSLTrans Transformation

- Visual language for model transformations
- Graph-based, rule-based
- Out-place so no rewriting performed
  - Suited for 'translation' transformations
- All its computations are terminating and confluent
- Unbounded loops during execution are not allowed
- Rules are grouped in sequential layers

---

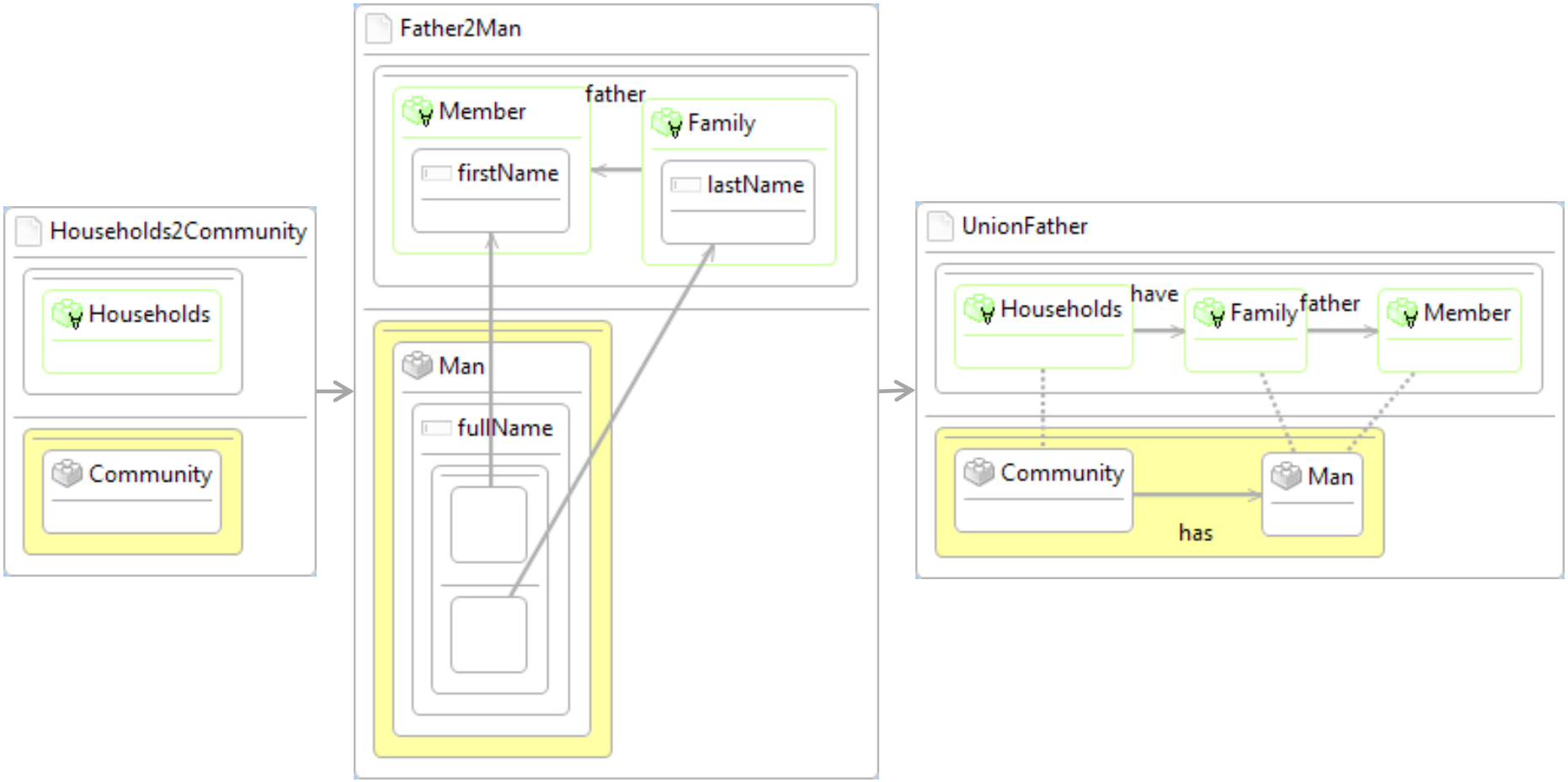
Selim, Gehan et al. "Model transformations for migrating legacy deployment models in the automotive industry." Software and Systems Modeling 14, no. 1 (2013): 365-381.

# Transformation Metamodels



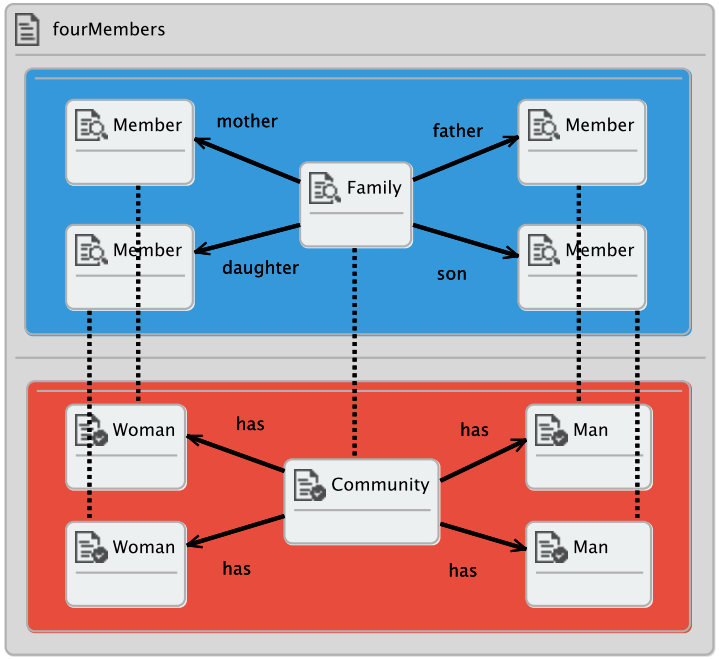

- Transform *Members* to *Men* and *Women*

# DSLTrans



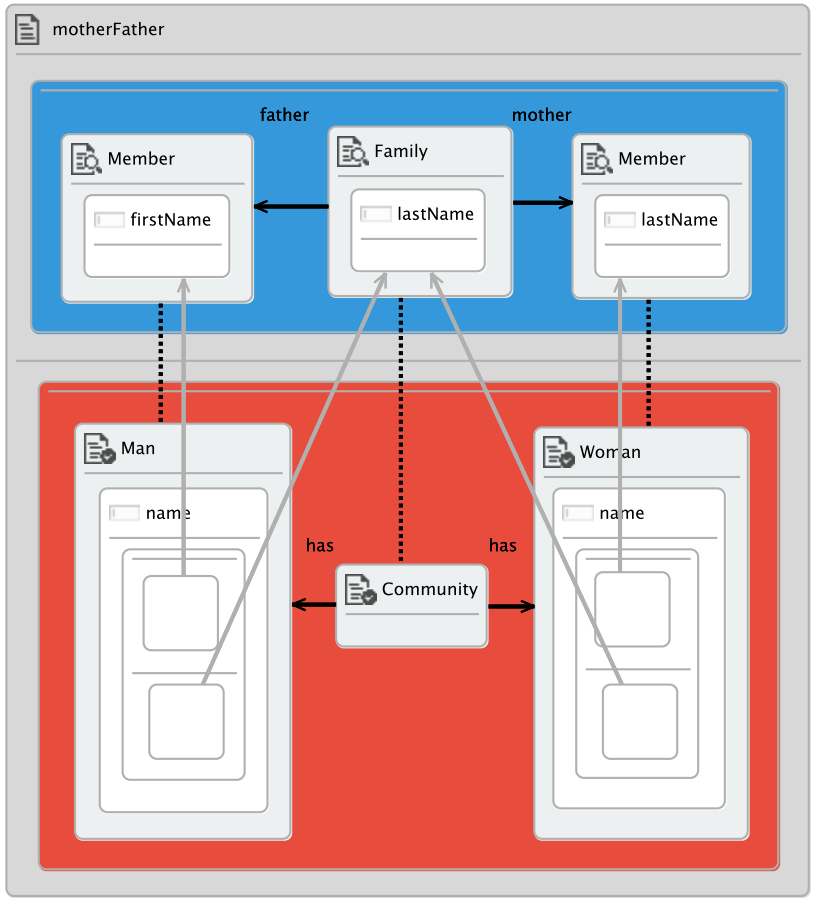

- Rules arranged in layers
- Match graph on top of rules
- Apply graph on bottom
  - Produced when match graph is found

# Contracts



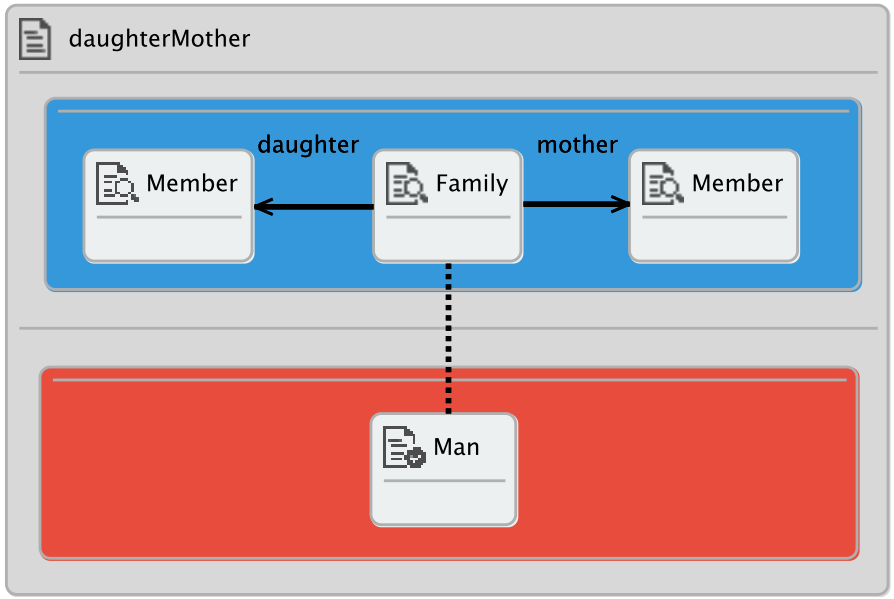

- Pre- / post- visual contracts
- If blue graph is found in input model, then red graph is found in output model
- Objective: Prove for all input models/transformation executions - input independence
- *A family with a father, mother, son, daughter should always produce two males and two females in the target community*

# Contracts




- Reasoning about attributes of elements
- *Is the full name of the produced Person correctly created from the last name of the Family and the first name of the Member?*

# Contracts




- A contract that will not hold
- *A family with a mother and a daughter will always produce a community with a man*

# Path Conditions

- All possible executions of the transformation are symbolically constructed
  - Built as sets of rules called path conditions
    - No rules execute, only rule 1 executes, rule 1 and rule 2 both execute
    - Rule dependencies/combinations resolved
  - Final set of path conditions represents all possible transformation executions
- A contract holds for a transformation if it holds for all generated path conditions

---

L. Lúcio, B. Oakes, and H. Vangheluwe. A technique for symbolically verifying properties of graph-based model transformations. Technical report, Technical Report SOCS-TR-2014.1, McGill U, 2014.

# Other Work at MODELS 2015

- Poster available
- Fully Verifying Transformation Contracts for Declarative ATL (MODELS)
  Bentley James Oakes, Javier Troya, Levi Lucio, and Manuel Wimmer
  (McGill University, Canada; Vienna University of Technology, Austria)
  Thursday at 3:30 PM
- Finding and fixing bugs in model transformations with formal verification: An experience report (AMT)
  Gehan M. K. Selim, James R. Cordy, Juergen Dingel, Levi Lucio and Bentley James Oakes
  (Queen's University, Canada; McGill University, Canada)

# Conclusion

- Can verify visual contracts on DSLTrans transformations in feasible time
- Contracts verified on all transformation executions
- Eclipse plugin to build transformation and contracts
- Future work is to focus on contract-driven development of model transformations

- Thank you for your time!

**SyVOLT: Full Model Transformation Verification Using Contracts**

**Levi Lúcio, Bentley James Oakes, Claudio Gomes, Gehan Selim, Juergen Dingel, James R. Cordy, Hans Vangheluwe**

`http://msdl.cs.mcgill.ca/people/levi/files/MODELS2015`