MultEcore

Combining The Best of Fixed-Level and Multilevel Metamodelling

Fernando Macías Adrian Rutle Volker Stolz

October 4, 2016

MOF-based approaches

Pros

- ✓ High reliability
- ✓ Mature (meta)modelling ecosystems
- ✓ Good tool coverage

MOF-based approaches

Pros

- ✓ High reliability
- ✓ Mature (meta)modelling ecosystems
- ✓ Good tool coverage

Cons

- × Mixed abstraction levels
- × Synthetic typing relation
- × Convoluted

Multilevel approaches

Pros

- ✓ Unbounded number of levels
- ✓ Deep hierarchies (potency)
- ✓ Linguistic extensions

Multilevel approaches

Pros

- ✓ Unbounded number of levels
- ✓ Deep hierarchies (potency)
- ✓ Linguistic extensions

Cons

- × Lack of clear consensus on the foundations
- × No common focus in current multilevel tools
- × Technology lock-in

Multilevel approaches

Pros

- ✓ Unbounded number of levels
- ✓ Deep hierarchies (potency)
- ✓ Linguistic extensions

Cons

- × Lack of clear consensus on the foundations
- × No common focus in current multilevel tools
- × Technology lock-in

"There is still no clear consensus on what the paradigm actually entails and how it should be applied"

Multilevel approaches

Pros

- ✓ Unbounded number of levels
- ✓ Deep hierarchies (potency)
- ✓ Linguistic extensions

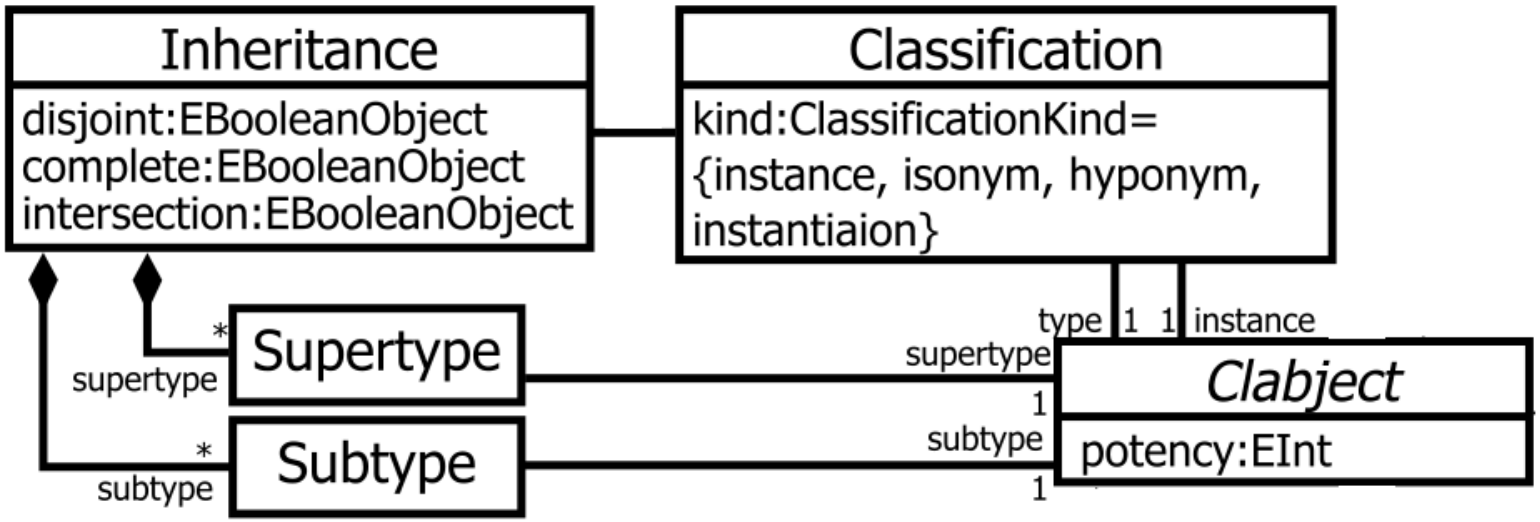
Cons

- × Lack of clear consensus on the foundations
- × No common focus in current multilevel tools
- × Technology lock-in

"There is still no clear consensus on what the paradigm actually entails and how it should be applied"

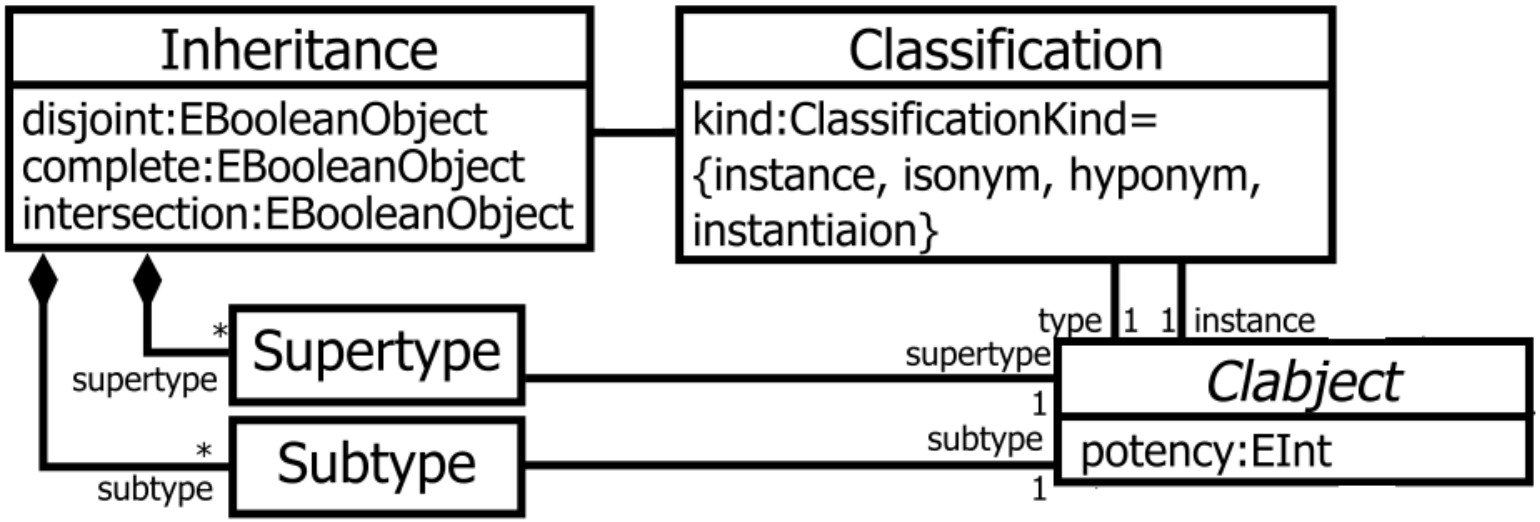
MULTI 2016 CfP

Common solution: Clabject



Adapted from: Melanee Project – `https://melanee2.informatik.uni-mannheim.de/confluence/`

Common solution: Clabject



Adapted from: Melanee Project – `https://melanee2.informatik.uni-mannheim.de/confluence/`

Issues

- × Requires a linguistic metamodel
- × Every element needs a linguistic type
- × Synthetic typing and flattening of the ontological stack
- × Custom tools and representations

Our solution: MultEcore

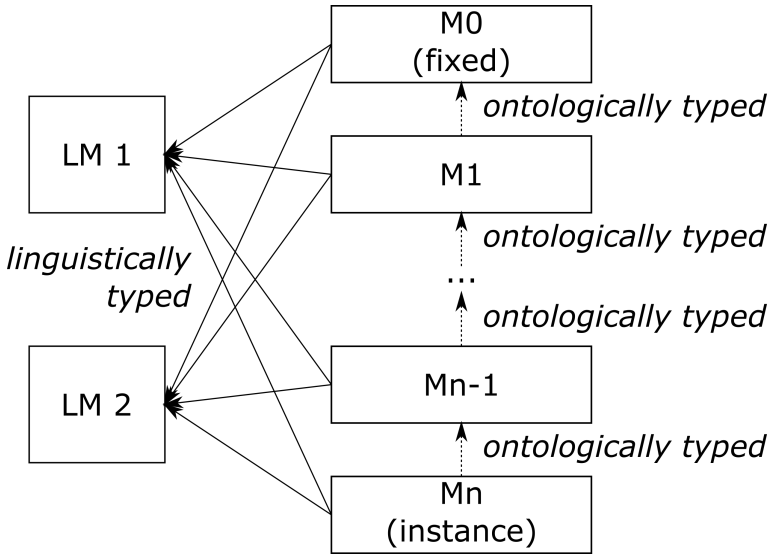
Ontological stack Does not require linguistic metamodels, synthetic typing relations or flattening

Linguistic metamodels Multiple and independent metamodels orthogonal to the ontological stack

Linguistic typing Less strict. An element may have none, one or several linguistic types

Linguistic extension Still allowed, but models with linguistic extension become dependent on the linguistic metamodel

Sliding window Reuse of the two-level cascading technique



Our solution: MultEcore

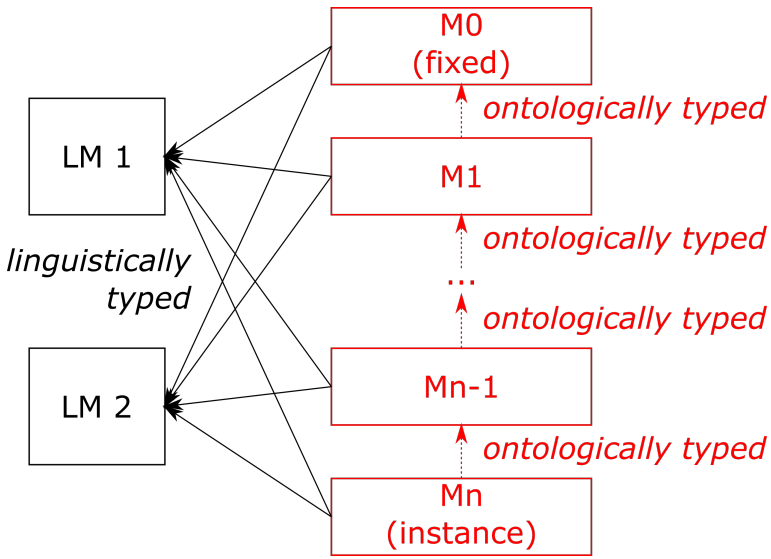
Ontological stack Does not require linguistic metamodels, synthetic typing relations or flattening

Linguistic metamodels Multiple and independent metamodels orthogonal to the ontological stack

Linguistic typing Less strict. An element may have none, one or several linguistic types

Linguistic extension Still allowed, but models with linguistic extension become dependent on the linguistic metamodel

Sliding window Reuse of the two-level cascading technique



Our solution: MultEcore

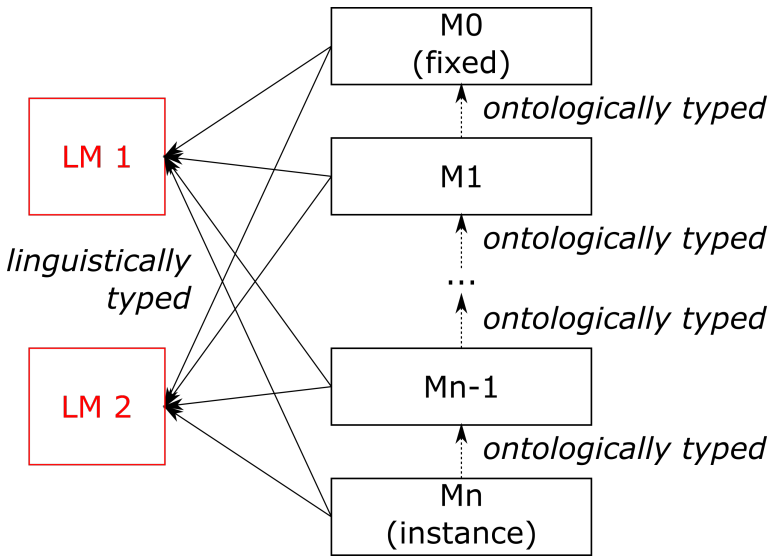
Ontological stack Does not require linguistic metamodels, synthetic typing relations or flattening

Linguistic metamodels Multiple and independent metamodels orthogonal to the ontological stack

Linguistic typing Less strict. An element may have none, one or several linguistic types

Linguistic extension Still allowed, but models with linguistic extension become dependent on the linguistic metamodel

Sliding window Reuse of the two-level cascading technique



Our solution: MultEcore

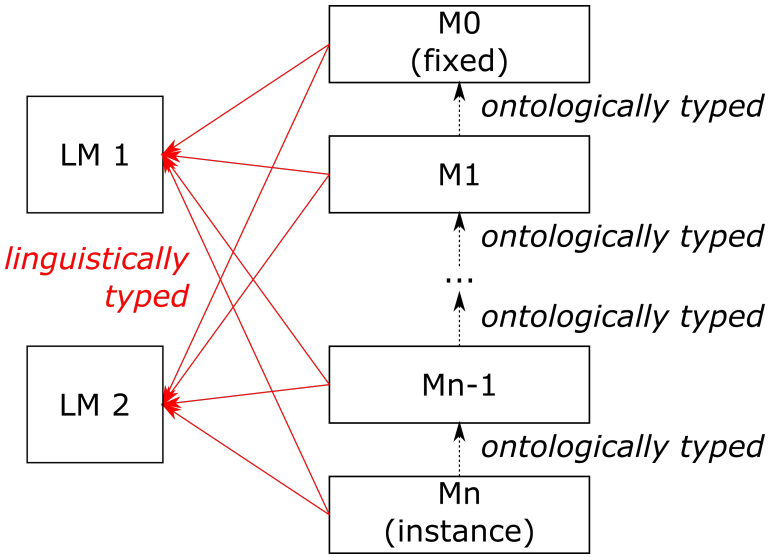
Ontological stack Does not require linguistic metamodels, synthetic typing relations or flattening

Linguistic metamodels Multiple and independent metamodels orthogonal to the ontological stack

Linguistic typing Less strict. An element may have none, one or several linguistic types

Linguistic extension Still allowed, but models with linguistic extension become dependent on the linguistic metamodel

Sliding window Reuse of the two-level cascading technique



Our solution: MultEcore

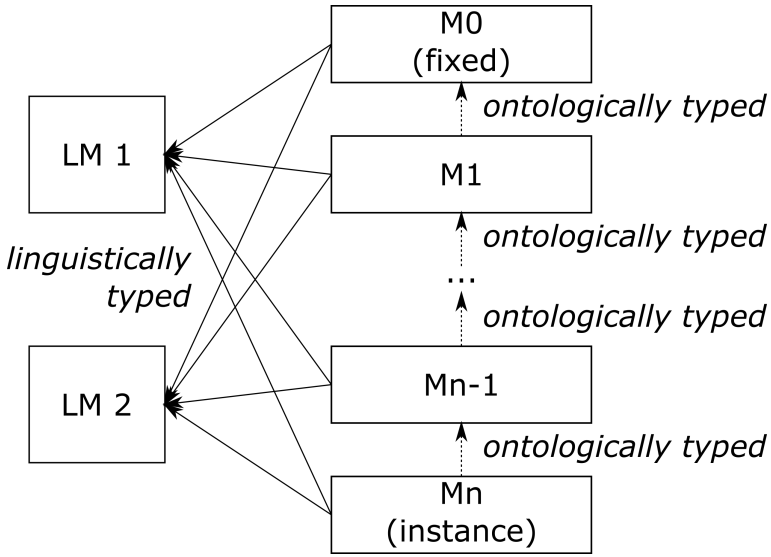
Ontological stack Does not require linguistic metamodels, synthetic typing relations or flattening

Linguistic metamodels Multiple and independent metamodels orthogonal to the ontological stack

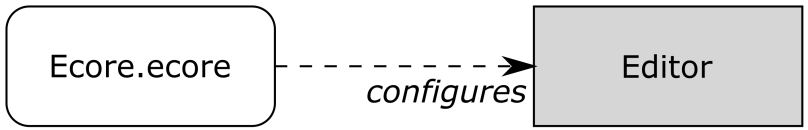
Linguistic typing Less strict. An element may have none, one or several linguistic types

Linguistic extension Still allowed, but models with linguistic extension become dependent on the linguistic metamodel

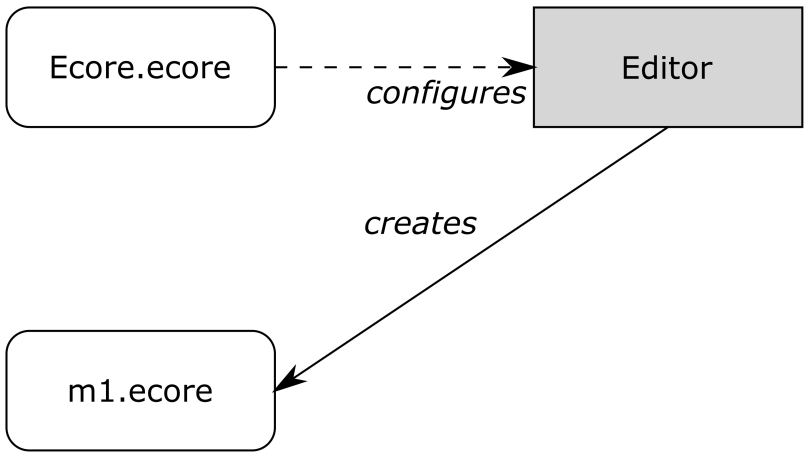
Sliding window Reuse of the two-level cascading technique



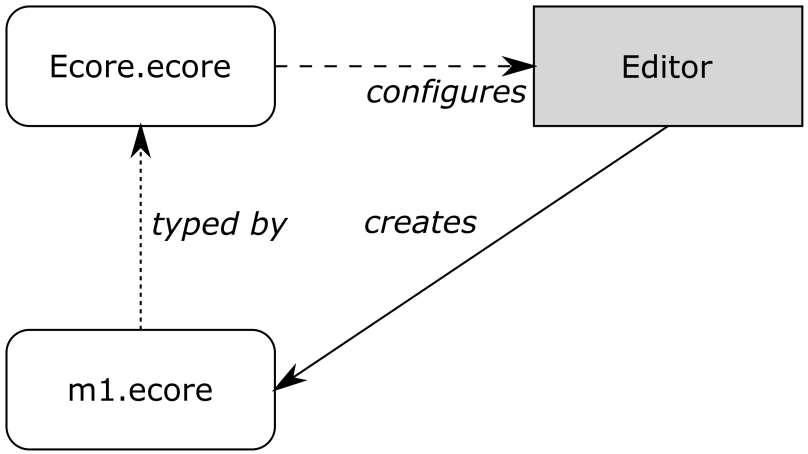
Realization on EMF – Sliding window



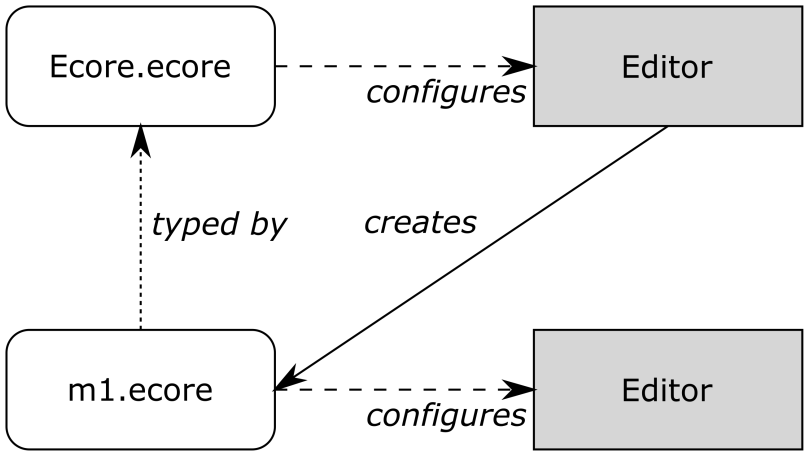
Realization on EMF – Sliding window



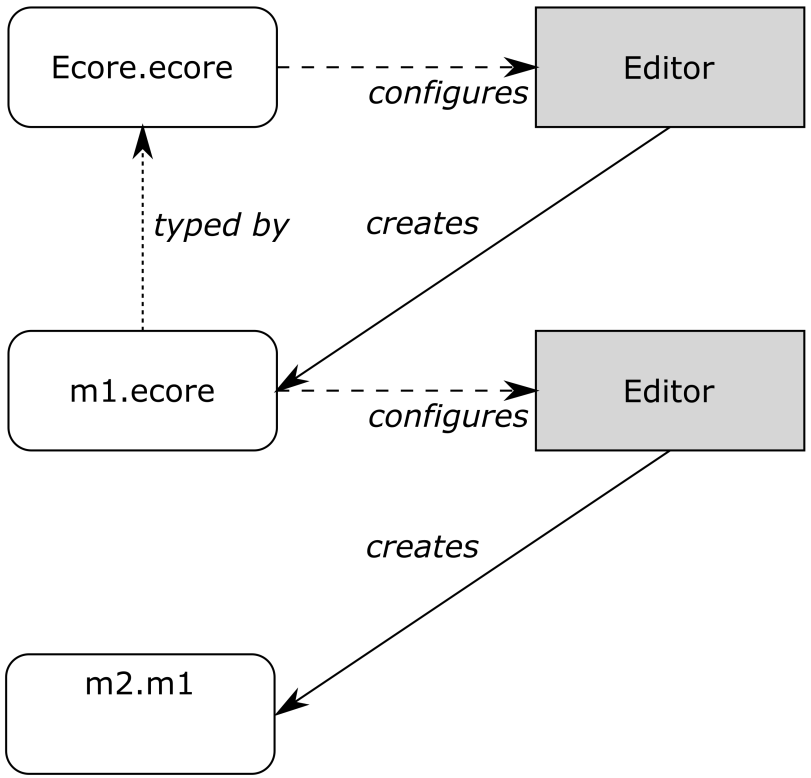
Realization on EMF – Sliding window



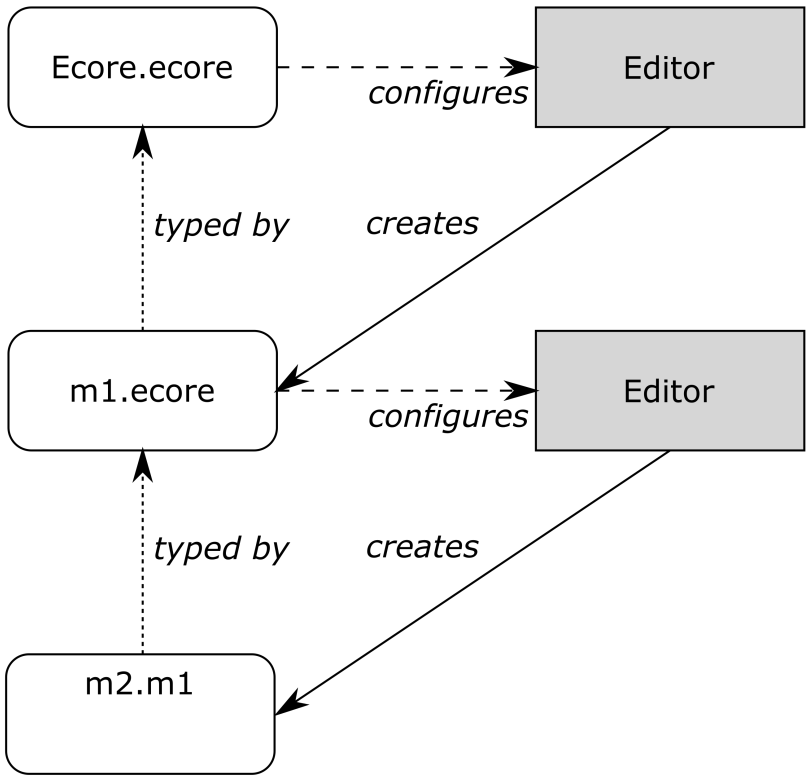
Realization on EMF – Sliding window



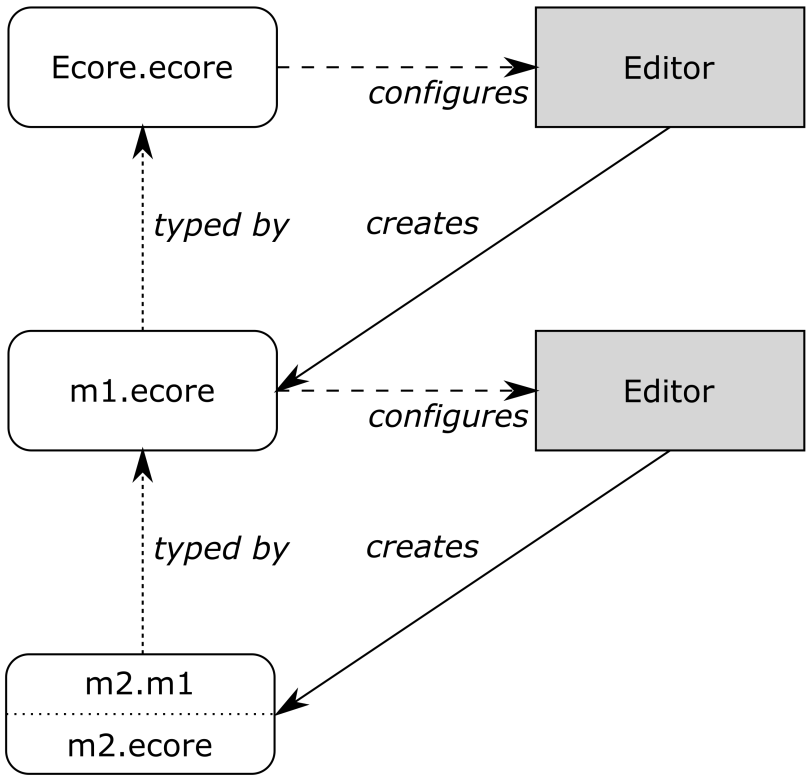
Realization on EMF – Sliding window



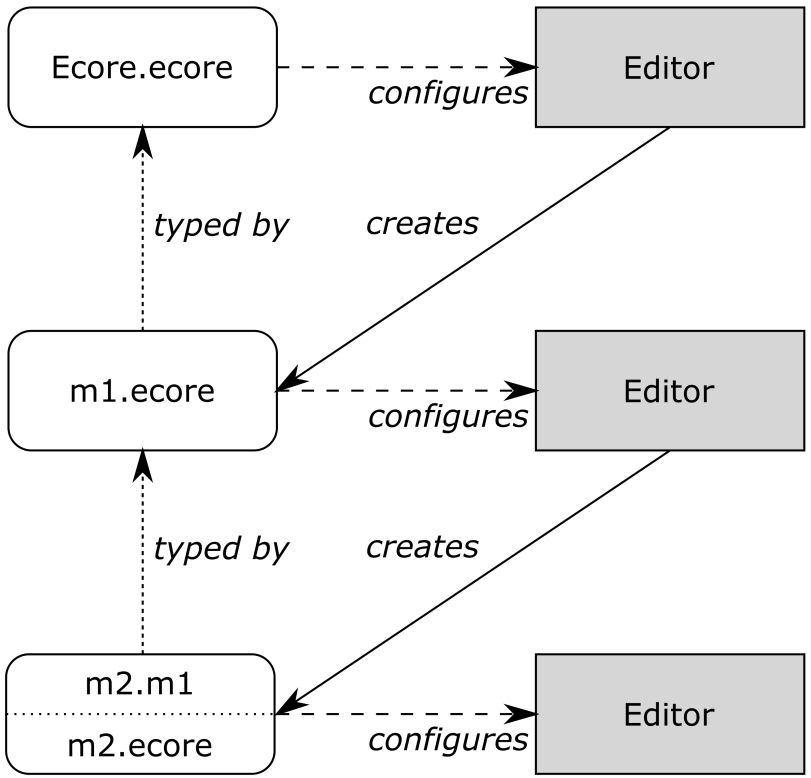
Realization on EMF – Sliding window



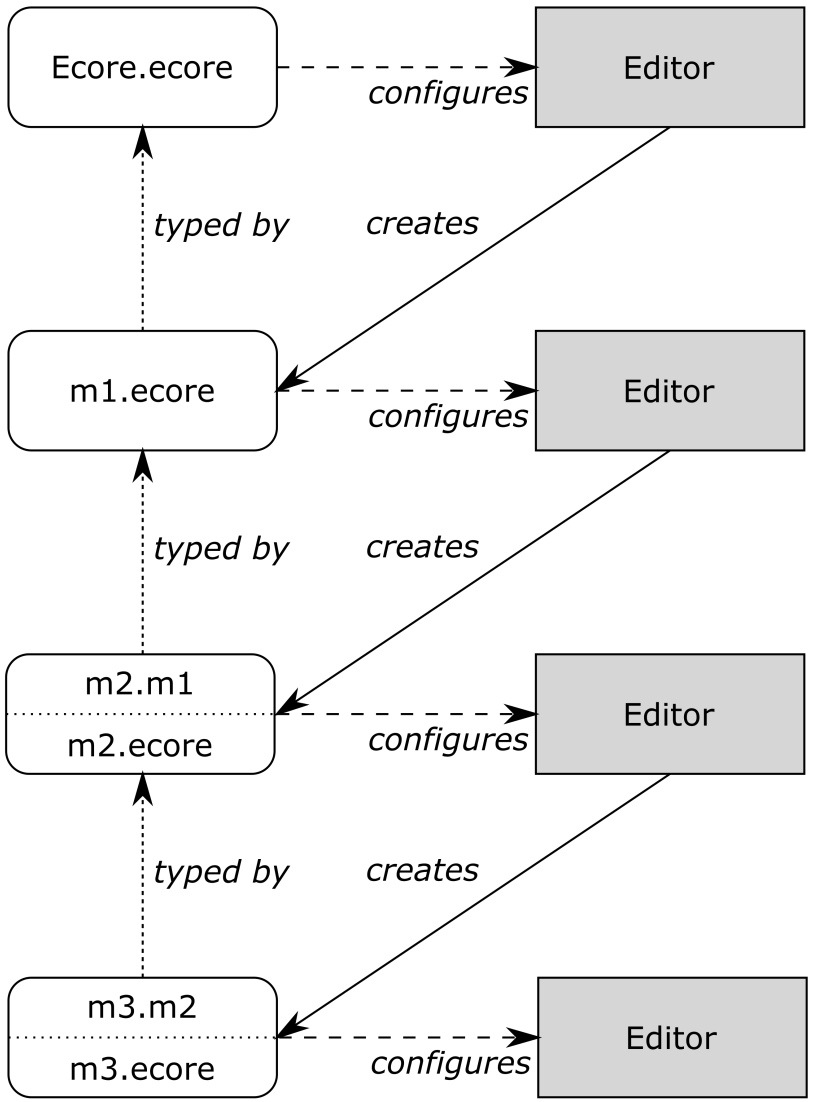
Realization on EMF – Sliding window



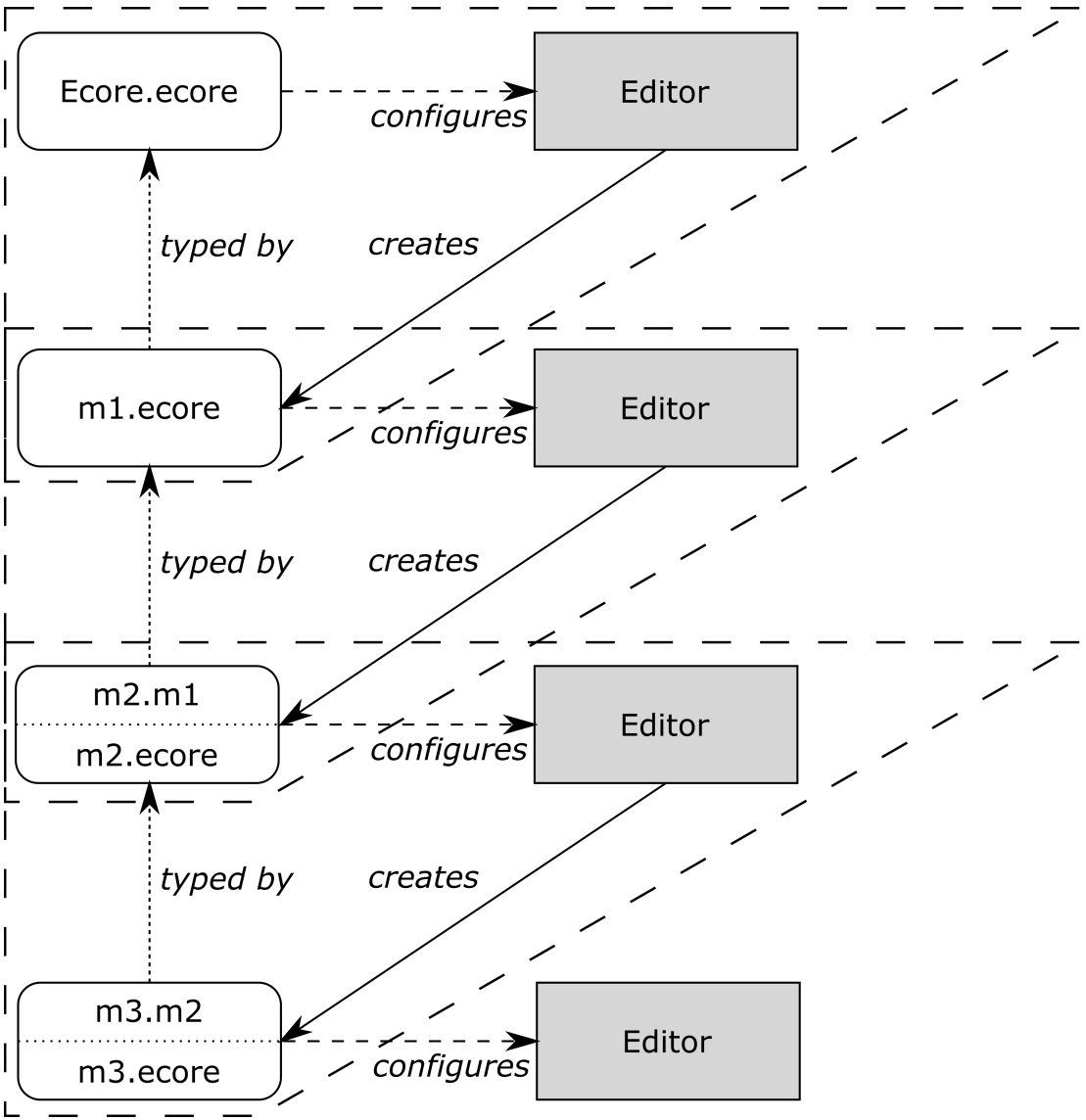
Realization on EMF – Sliding window



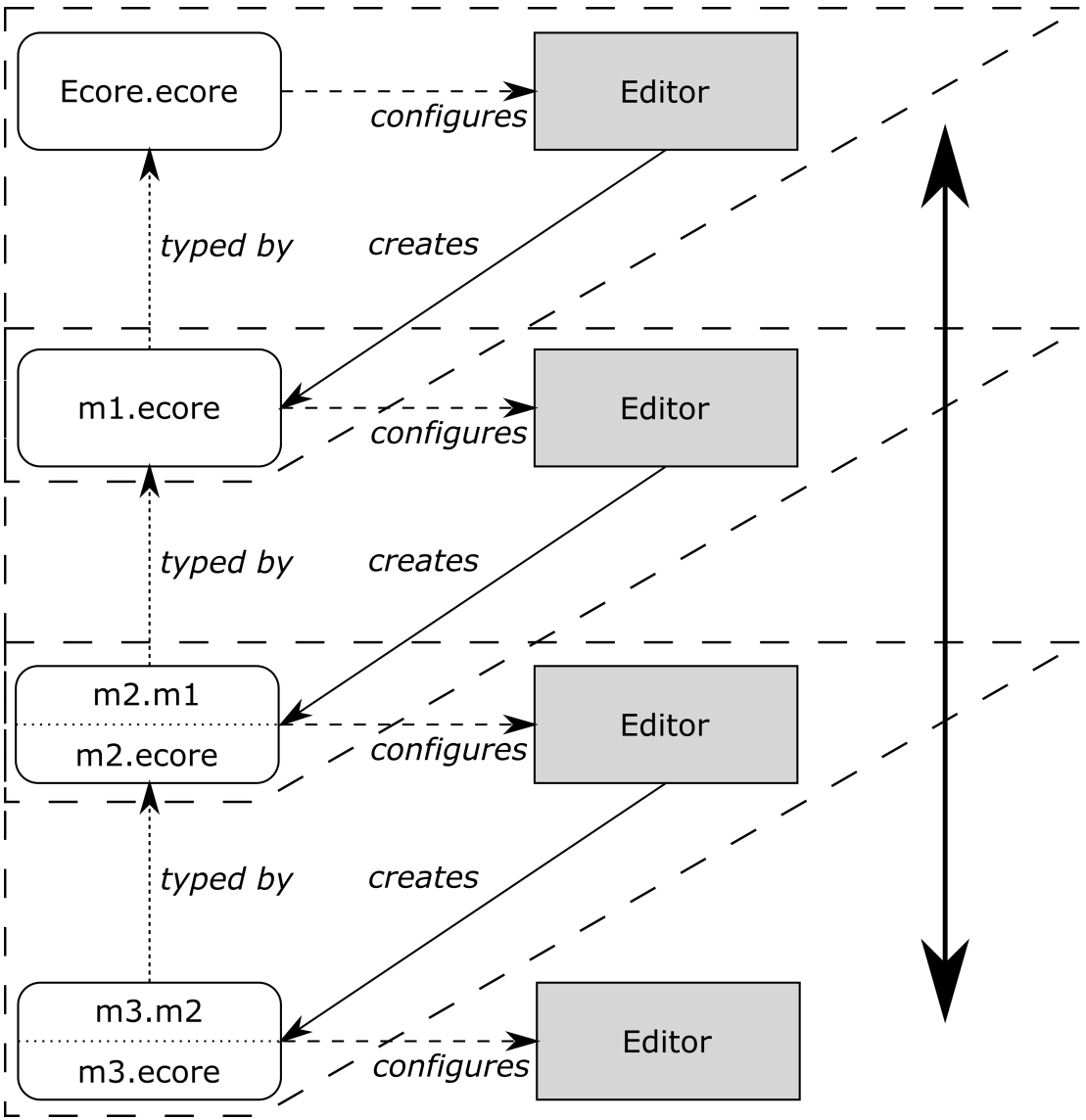
Realization on EMF – Sliding window



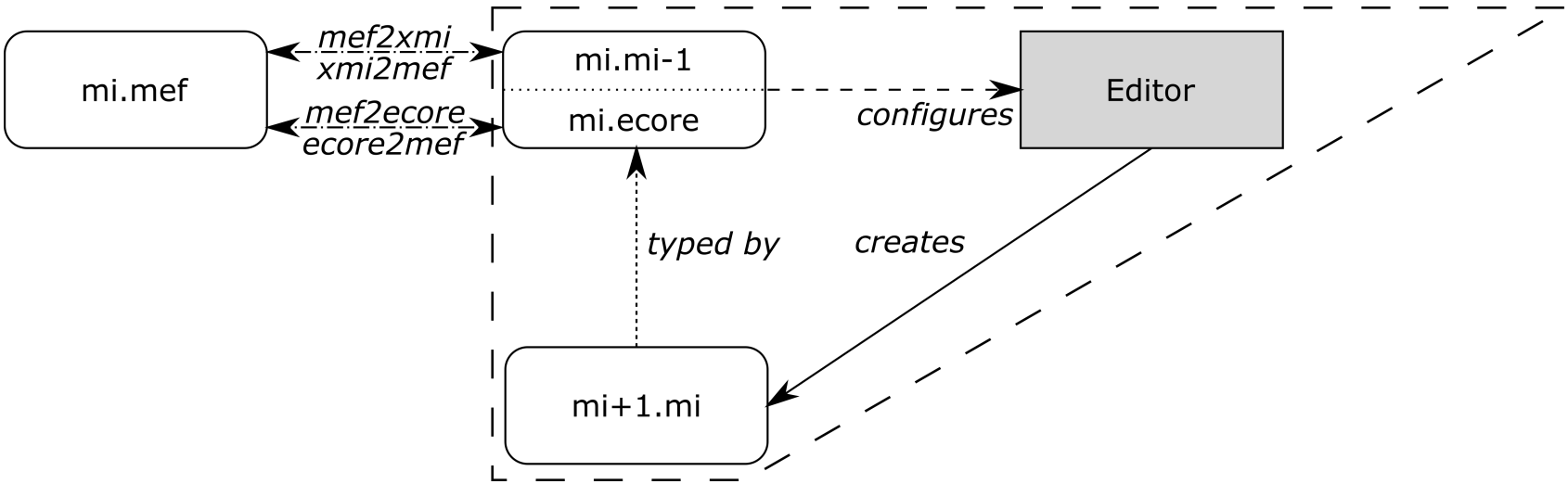
Realization on EMF – Sliding window



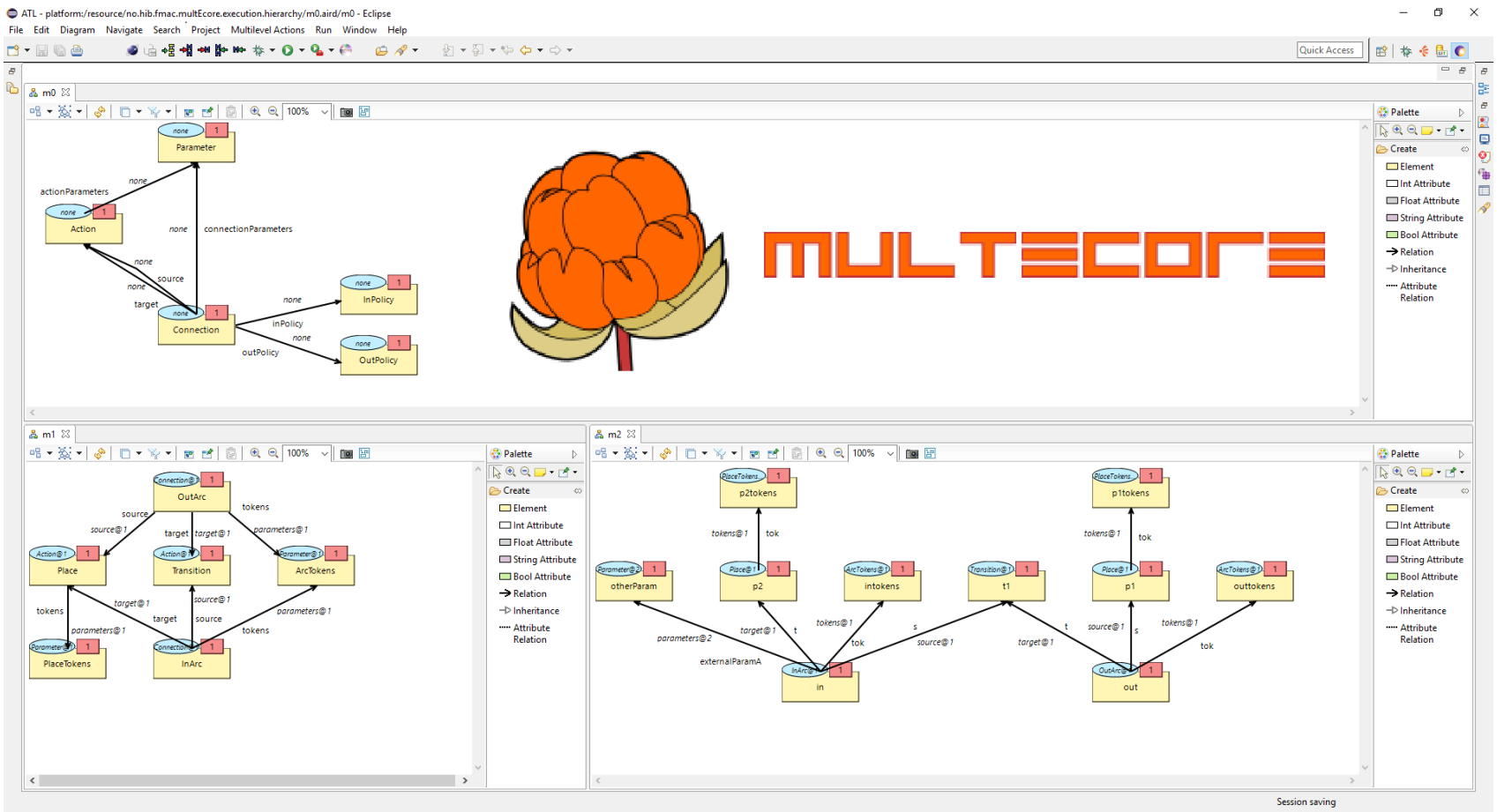
Realization on EMF – Sliding window



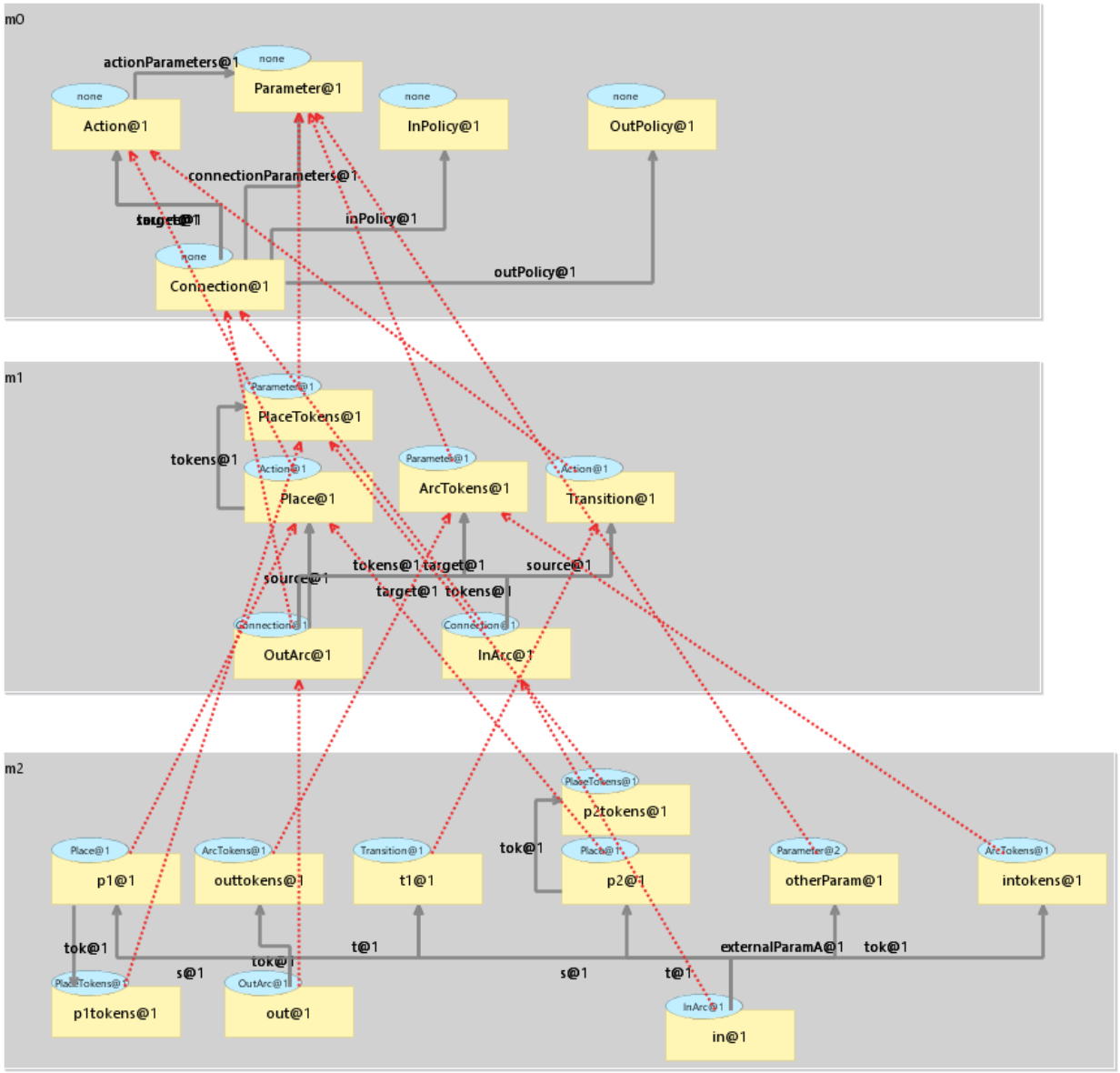
Realization on EMF – MEF representation



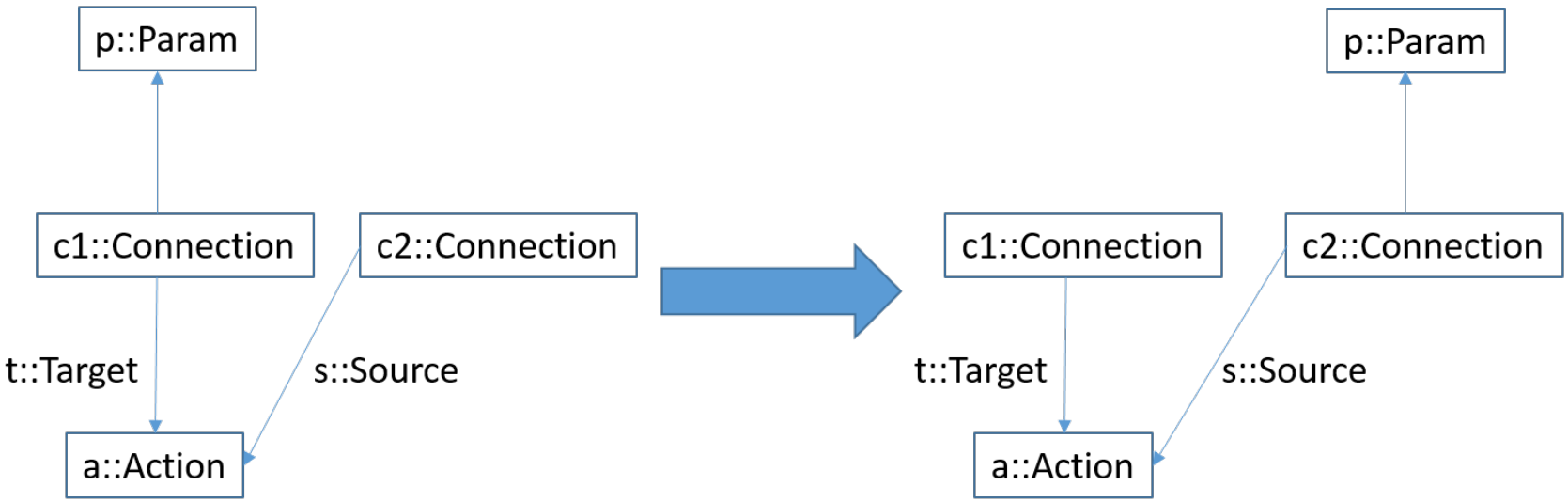
MultEcore tool



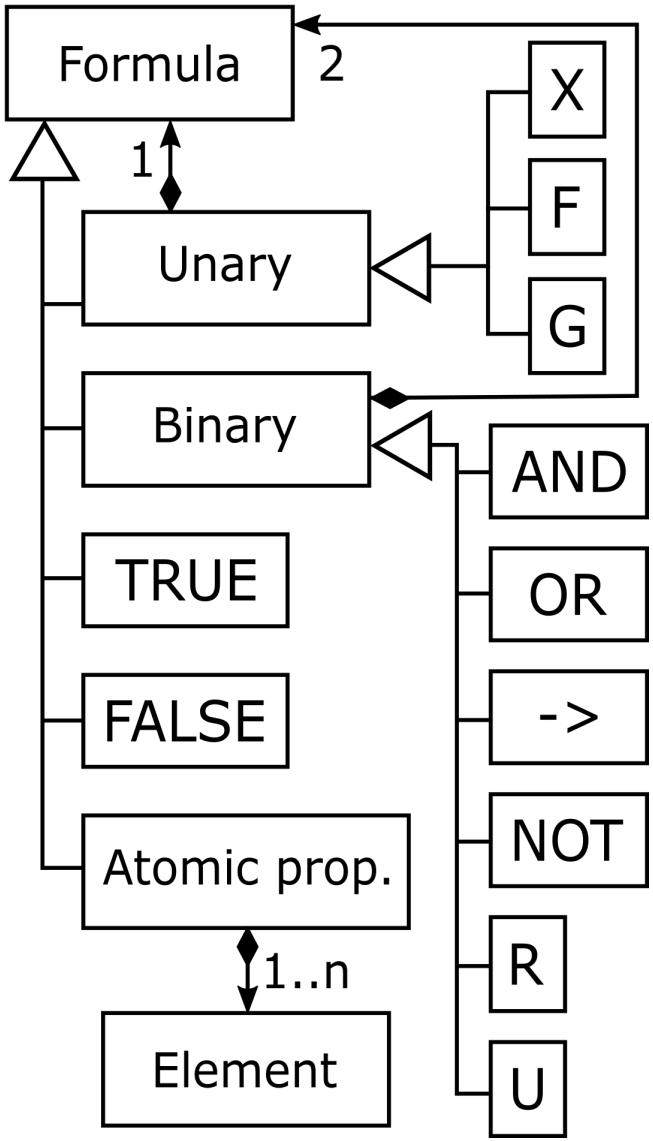
Applications – Behavioural metamodelling



Multilevel Coupled Model Transformations

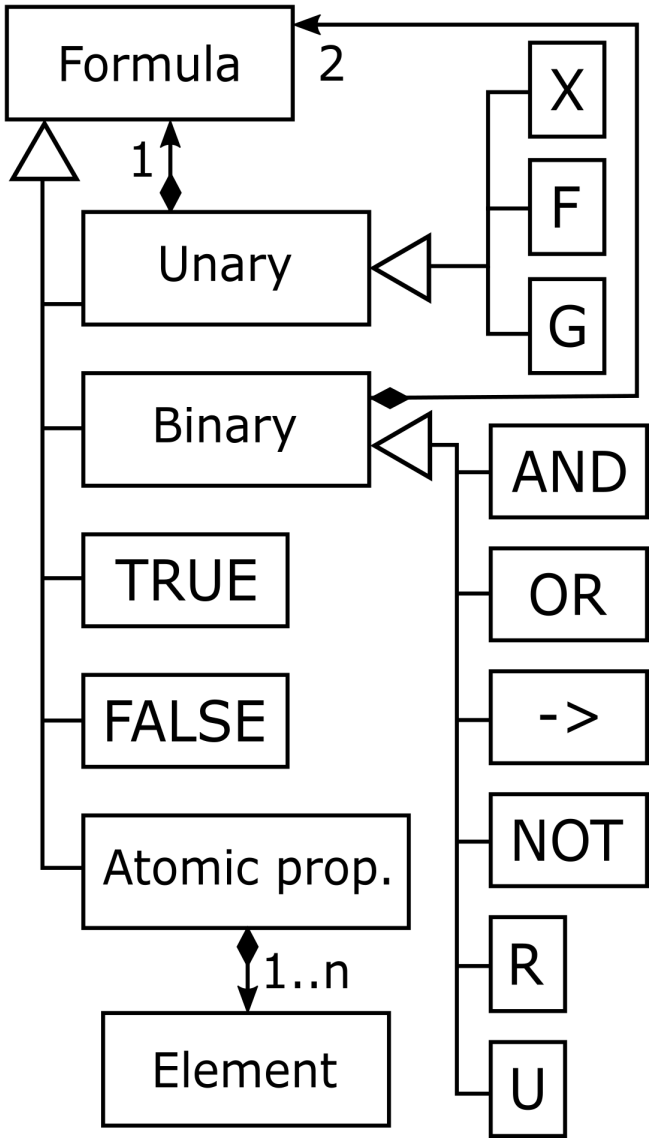


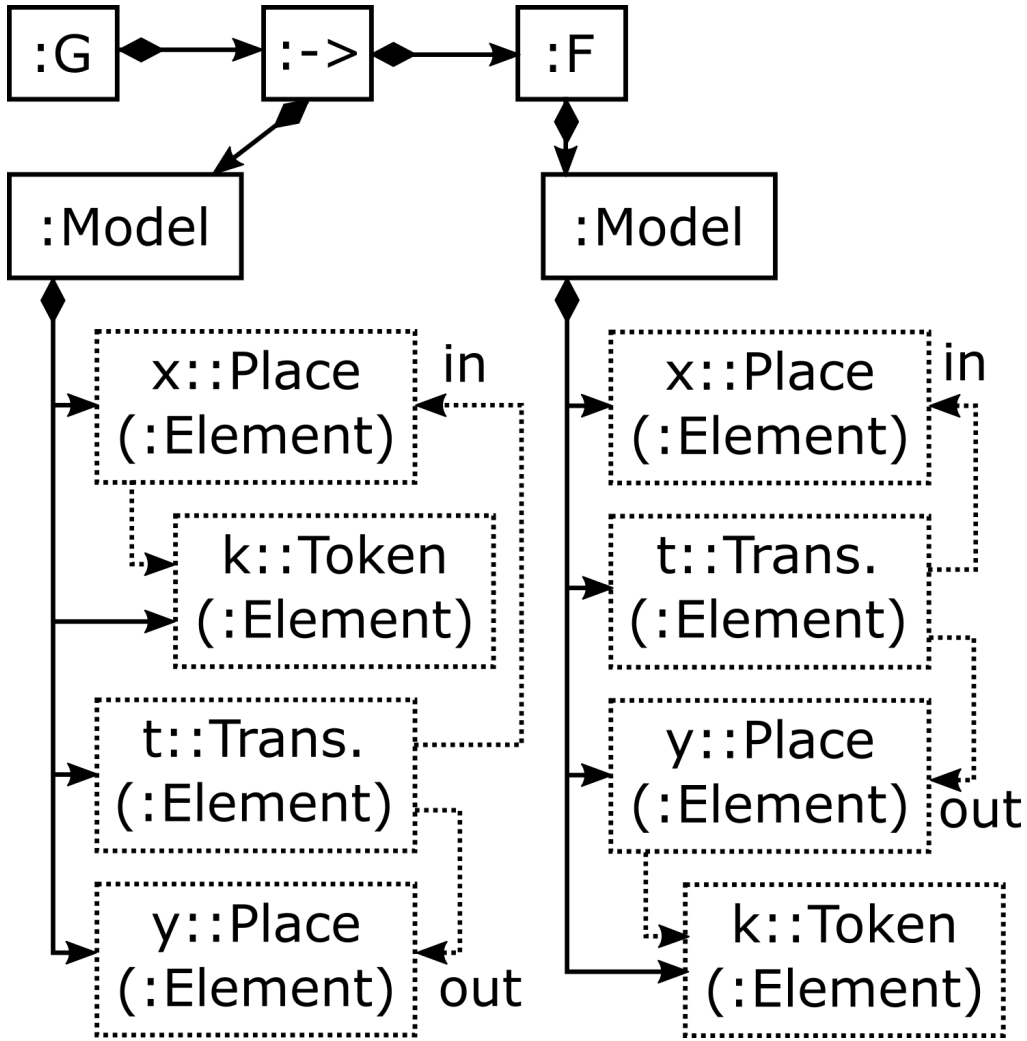
Applications – Runtime Verification



Macias et al. *Integration of Runtime Verification into Metamodeling for Simulation and Code Generation.* RV 2016

Applications – Runtime Verification





Macias et al. *Integration of Runtime Verification into Metamodeling for Simulation and Code Generation.* RV 2016

Conclusions

- MultEcore, an alternative framework for multilevel modelling
- Applied to behavioural metamodelling and RV
- Tool as an Eclipse plugin, bypassing EMF's two-level limitation
  - Small learning curve
  - Mature ecosystem and toolset

Future Work

- New multilevel functionalities: navigation of typing relations
- Creation of a hierarchy of behavioural models
- Implementation of Multilevel Coupled Model Transformations
- MEF formalization: metamodel and model transformations

`http://prosjekt.hib.no/ict/multecore/`