A Comparison of Traceability in MDE Mega-Models

Maximilian Meffert

1.
What is Traceability ?

Origins

Requirement Engineering:

How can I validate/prove that all requirements are met?

On what information can I base such validation/prove?

Idea

Engineering activities leave `traces.`

E.g. Version Control records:

- Modification (Change)
- Date
- User
- Comment

Traces

”[...] a ***relationship*** *between two or more products of the development process [...]”*
– IEEE

”[...] (possibly) non-material indication or evidence what has existed or happened [...]”
– OED

Answers to:

- What ?
- Who ?
- Where ?
- How ?
- When ?
- Why ?

Traceability

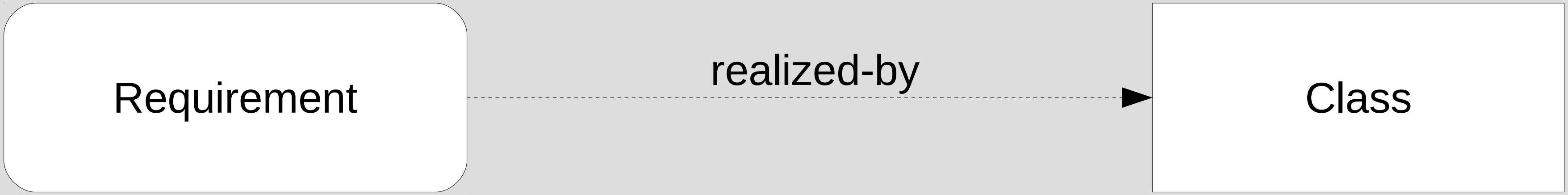
”[...] ***the ability to chronologically interrelate uniquely identifiable entities in an a way that matters.*** *[...] [It] refers to the capability for tracing artifacts along a set of chained [manual or automated] operations.”*
– Paige et al.

“The degree to which each element in a software development product establishes its reason for existing.”
– IEEE

The subject is expanded to general artifact/model traceability.

Traceability Links

> *”[...] a **relationship** between two*
> *or more products of the*
> *development process [...]”*
> *– IEEE*



Traceability Link Types

Type	Description
Dependency Link	Change Impact
Refinement Link	Abstraction/Composition/Aggregation Hierarchies
Evolution Link	Downstream Transformation
Satisfiability Link	Compliance/Conformance
Overlap Link	Share of Information, Difference in Representation
Conflict Link	Conflict between artifacts
Rationalization Link	Justification/Rationale for transformations
Contribution Link	Stakeholder association

accordng to Spanoudakis an Zisman

Traceability Categories

Pre Model	Intra Model	Post Model
Traceability between early non-model artifacts and first models	Traceability between models during model-refinement	Traceability between final models and artifacts generated by them

accordng to Paige et al.

Traceability Objectives

Technical Comprehension	System Management
• Supporting special audits • Improving changeability • **Proving system adequateness** • Extracting metrics • Tracking rationale • Finding reusable elements • Documenting re-engineering • **Supporting design decisions** • Change impact analysis • Synchronizing models • Extracting best practices • Understanding the system • **Extracting best practices**	• Prioritizing requirements • Proving system adequateness • Extracting metrics • **Monitoring progress** • Assessing the development process • **Establishing accountability** • Documenting re-engineering • Extracting best practices • **Change impact analysis** • Driving product-line development • Understanding the system

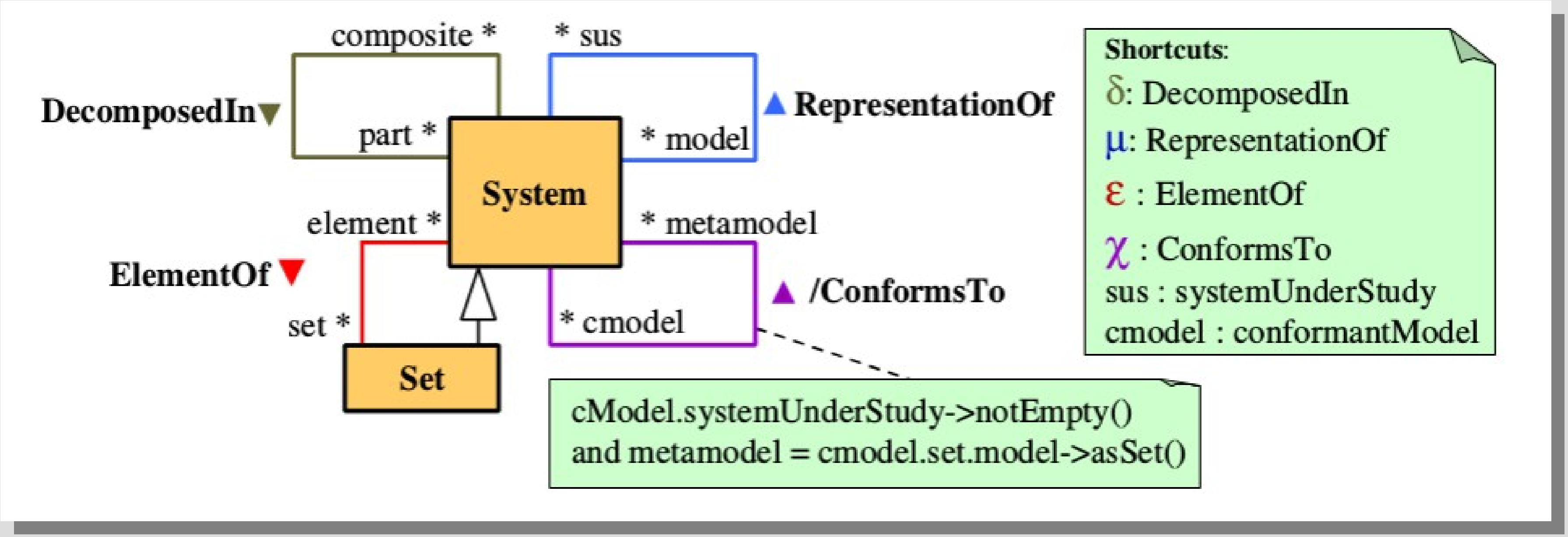
Traceability Objectives accordng to
Winkler and von Pilgrim
(but re-categorized)

2.
Comparison of MDE Mega-Models

Comparison Scheme

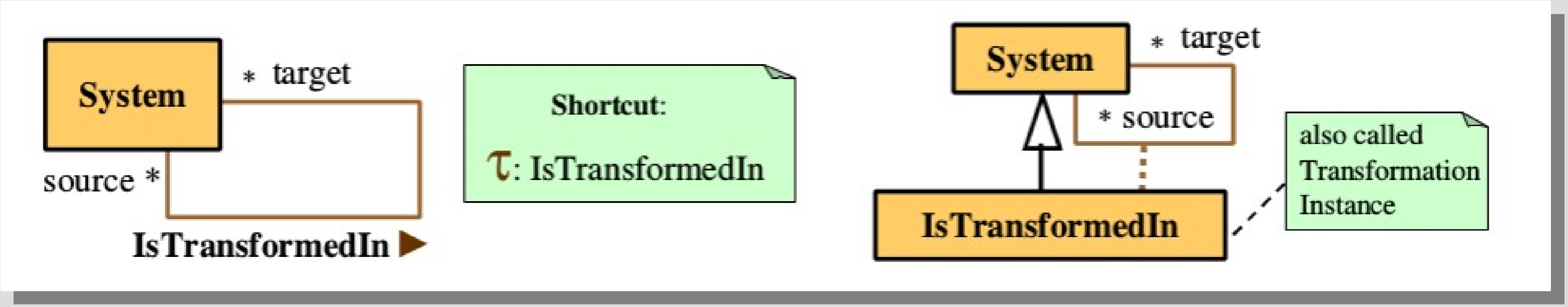
Model	Traceability Link Types	Traceability Categories	Traceability Objectives

The MDE Mega-Model



Shortcuts:
δ: DecomposedIn
μ: RepresentationOf
ε : ElementOf
χ : ConformsTo
sus : systemUnderStudy
cmodel : conformantModel

```
cModel.systemUnderStudy->notEmpty()
and metamodel = cmodel.set.model->asSet()
```



Shortcut:
τ: IsTransformedIn

also called Transformation Instance

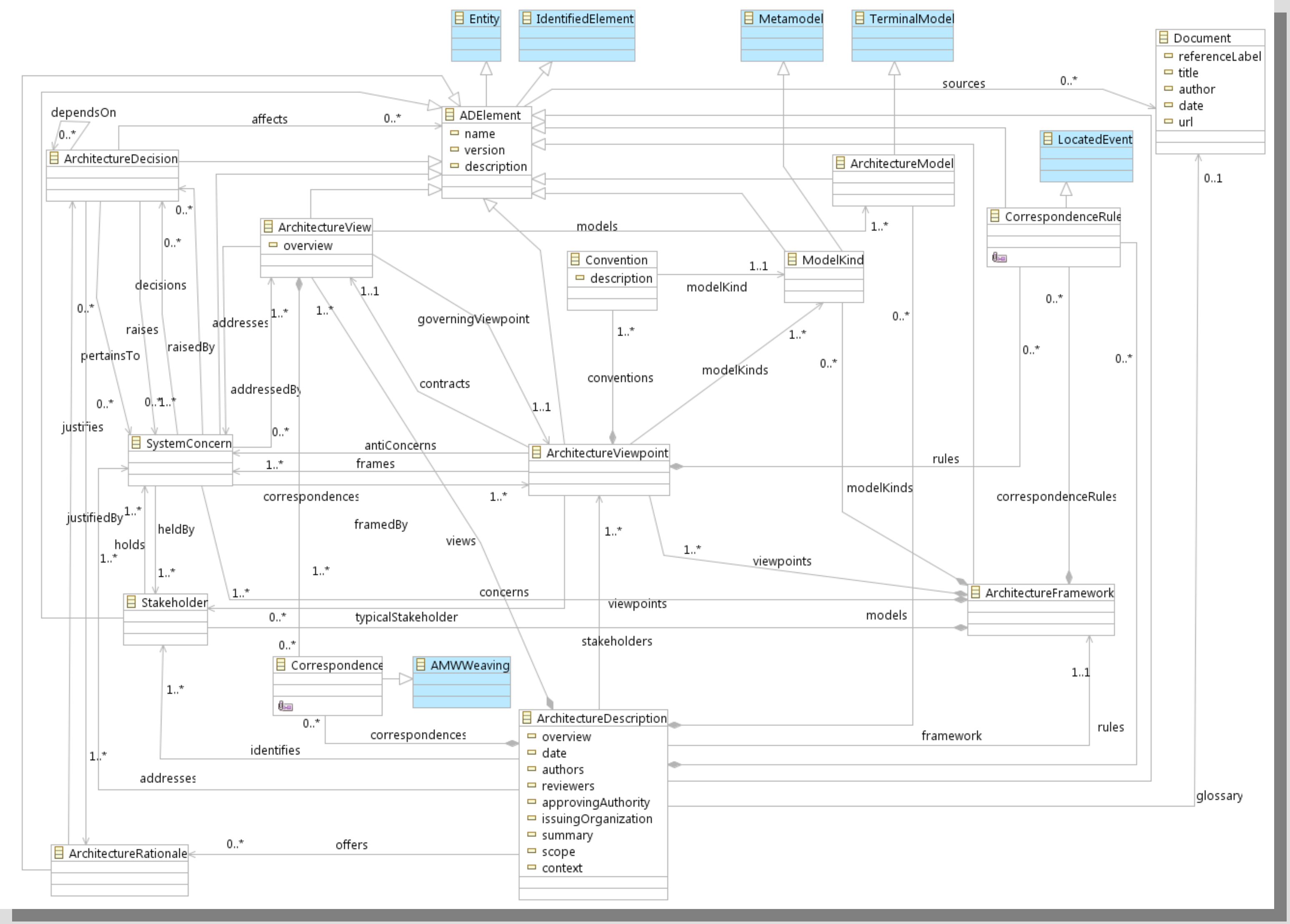
by Favre et al.

The MDE Mega-Model

Relation	Traceability Link Type
DecomposedIn	Refinement
ElementOf	Refinement
RepresentationOf	Overlap (implicit)
ConformsTo	Satisfiability, Dependency
IsTransformedIn	Evolution

The MDE Mega-Model	Dependency, Refinement, Evolution, Satisfiability, Overlap (implicit)	Pre Model, Intra Model, Post Model	Technical Comprehension

MEGAF



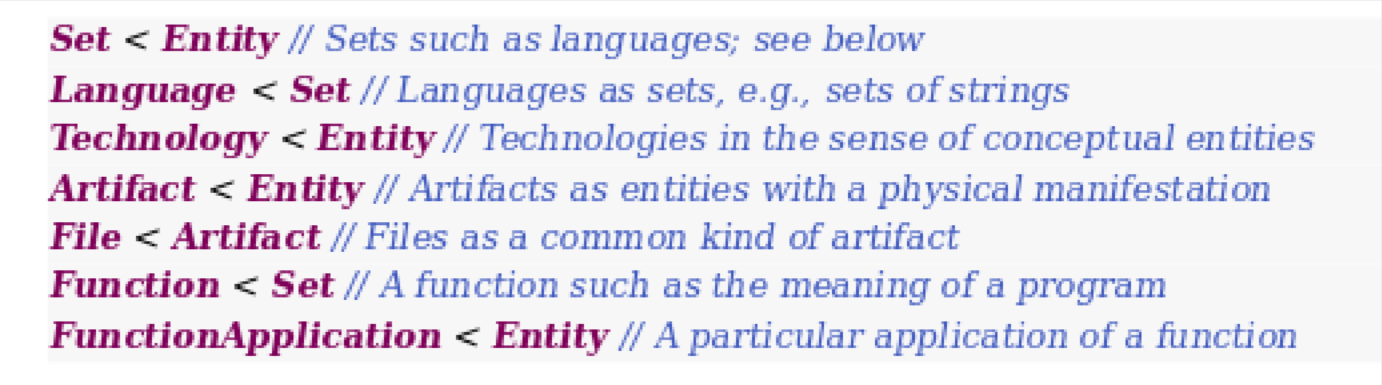
by Hillard et al.

MEGAF

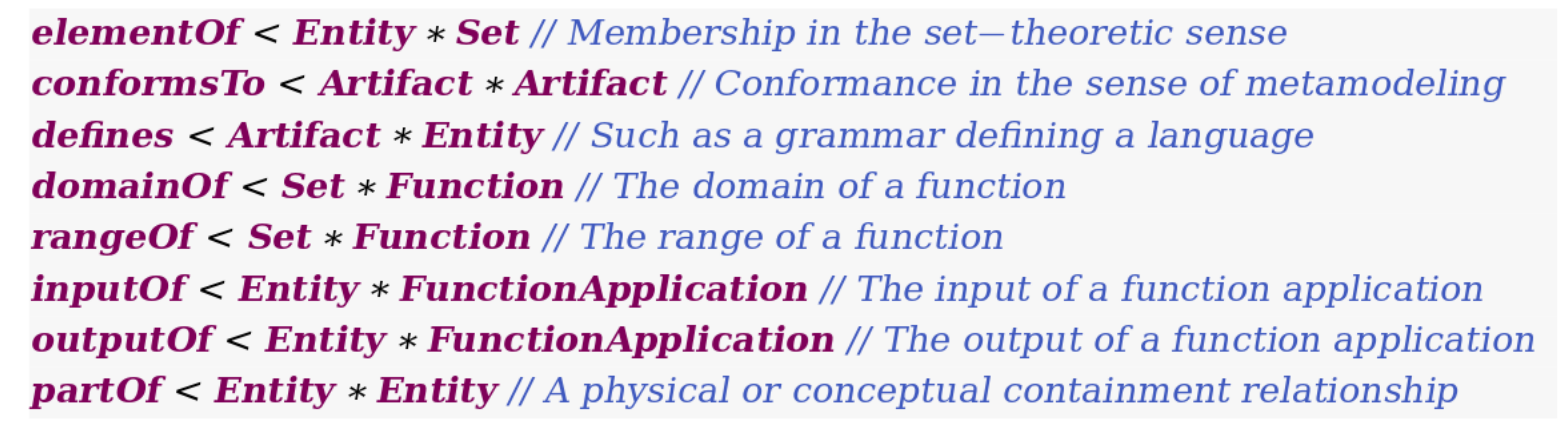
Relation	Traceability Link Type
ArchitectureFramework **modelKinds** ModelKind	Refinement
ArchitectureViewpoint **contracts** ArchitectureView	Satisfiability, Dependency
ArchitectureRationale **justifies** ArchitectureDecision, ArchitectureDescription **identifies** Stakeholder	Rationalization, Dependency
Stakeholder **holds** SystemConcern	Contribution

MEGAF	Dependency, Refinement, Satisfiability, Rationalization, Contribution	Pre Model (framed) Intra Model (framed) Post Model (framed)	Technical Comprehension, System Management

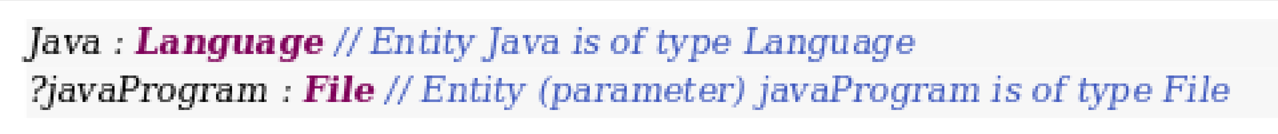
MegaL



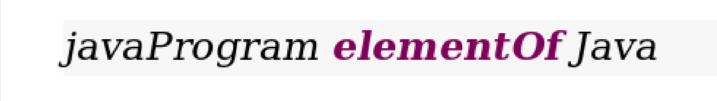
```
Set < Entity // Sets such as languages; see below
Language < Set // Languages as sets, e.g., sets of strings
Technology < Entity // Technologies in the sense of conceptual entities
Artifact < Entity // Artifacts as entities with a physical manifestation
File < Artifact // Files as a common kind of artifact
Function < Set // A function such as the meaning of a program
FunctionApplication < Entity // A particular application of a function
```



```
elementOf < Entity * Set // Membership in the set—theoretic sense
conformsTo < Artifact * Artifact // Conformance in the sense of metamodeling
defines < Artifact * Entity // Such as a grammar defining a language
domainOf < Set * Function // The domain of a function
rangeOf < Set * Function // The range of a function
inputOf < Entity * FunctionApplication // The input of a function application
outputOf < Entity * FunctionApplication // The output of a function application
partOf < Entity * Entity // A physical or conceptual containment relationship
```



```
Java : Language // Entity Java is of type Language
?javaProgram : File // Entity (parameter) javaProgram is of type File
```



```
javaProgram elementOf Java
```

by Lämmel & Varanovich

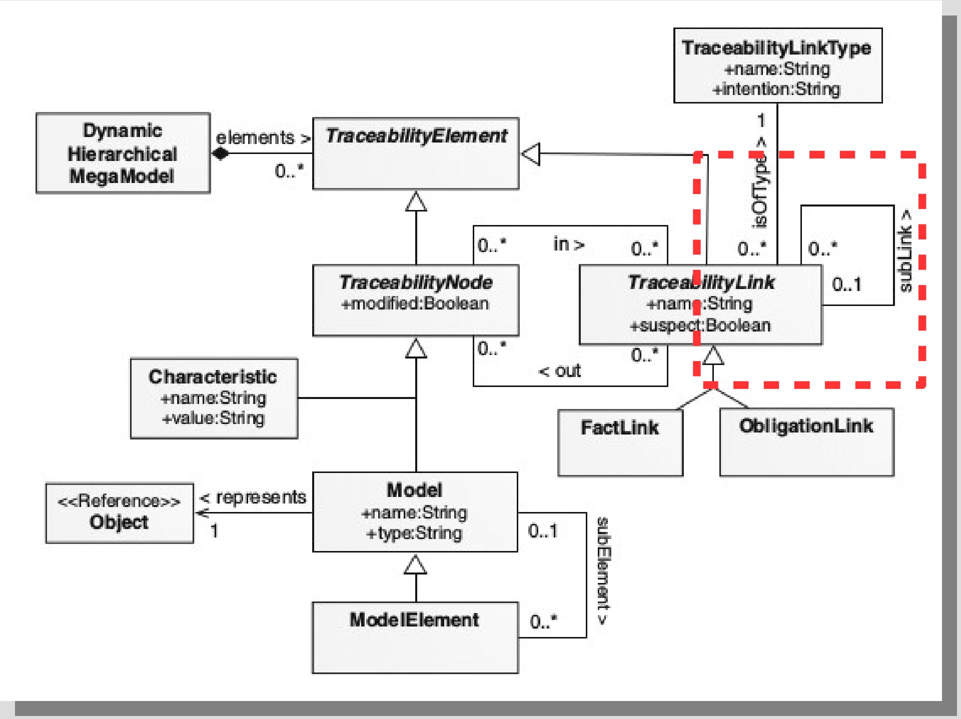
MegaL

Relation	Traceability Link Type
elementOf, partOf, domainOf, rangeOf	Refinement
conformsTo, defines, domainOf, rangeOf	Satisfiability, Dependency
inputOf, outputOf	Evolution (Overlap)

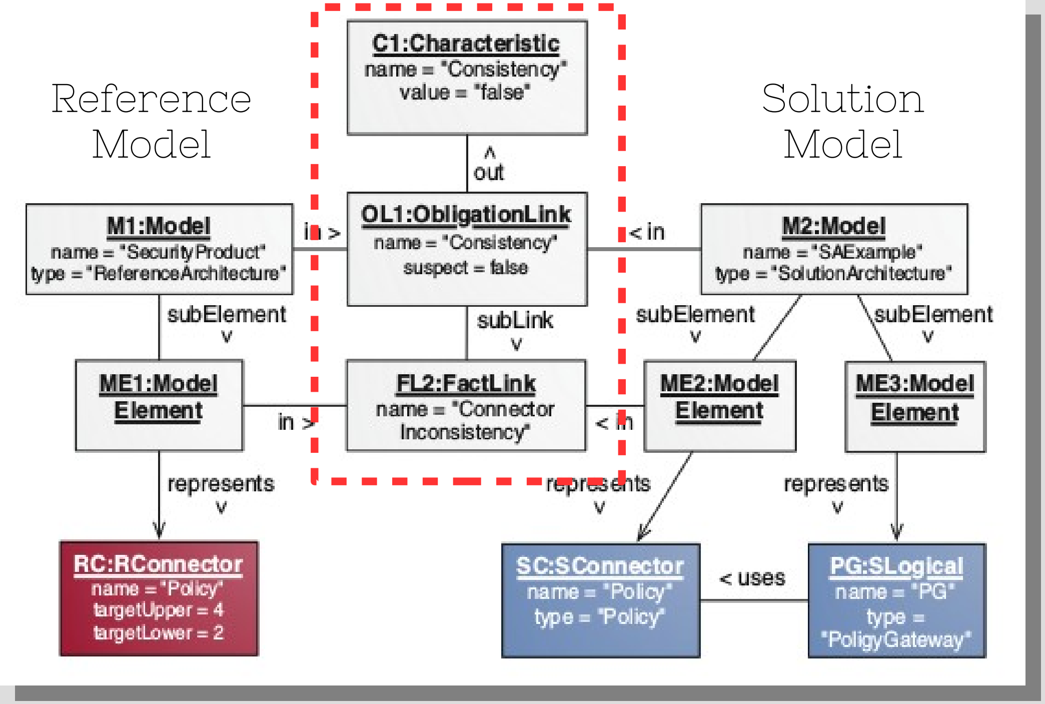
MegaL	Refinement, Satisfiability Dependency, Evolution (Overlap)	(Intra Model) Post Model	Technical Comprehension

Dynamic Hierarchical Mega-Models

(Explicit traceability model)



Top-Down Hierarchy



Inconsistency because RConnector requests at least 2 users, but only SConnector exists.

by Seibel et al.

Dynamic Hierarchical Mega-Models

- Explicit **Traceability Model** for MDE Tool Support
- Special interest in hierarchical between traceability links of different abstraction levels
- e.g. links between models may imply, that corresponding links between their elements exist

Dynamic Hierarchical Mega-Models	Conflict *(Custom Types)*	Intra Model	Technical Comprehension, System Management

Conclusion

Model	Traceability Link Types	Traceability Categories	Traceability Objectives
The MDE Mega-Model	Dependency Refinement Evolution Satisfiability Overlap	Pre Model Intra Model Post Model	Technical Comprehension
MEGAF (GMM4SA)	Dependency Refinement Satisfiability Rationalization Contribution	Pre Model (framed) Intra Model (framed) Post Model (framed)	Technical Comprehension System Management
Dynamic Hierarchical Mega-Models	Conflict *(Custom Types)*	Intra Model	Technical Comprehension System Management
MegaL	Dependency Refinement Satisfiability Overlap *Evolution (implicit)*	Post Model	Technical Comprehension

Fin.