

Traceability & Mega-Modeling

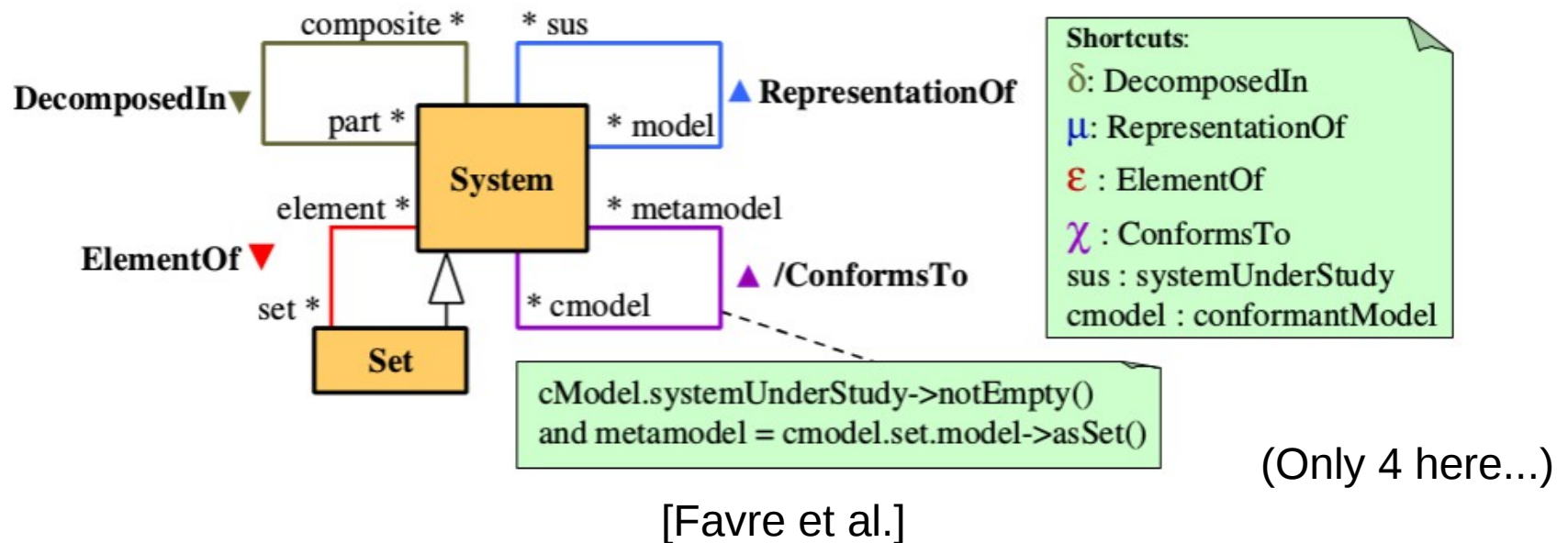
How to combine?

Traceability Relationships

Traceability information is usually expected in form of **relationships** or meta-data
(IEEE and others)

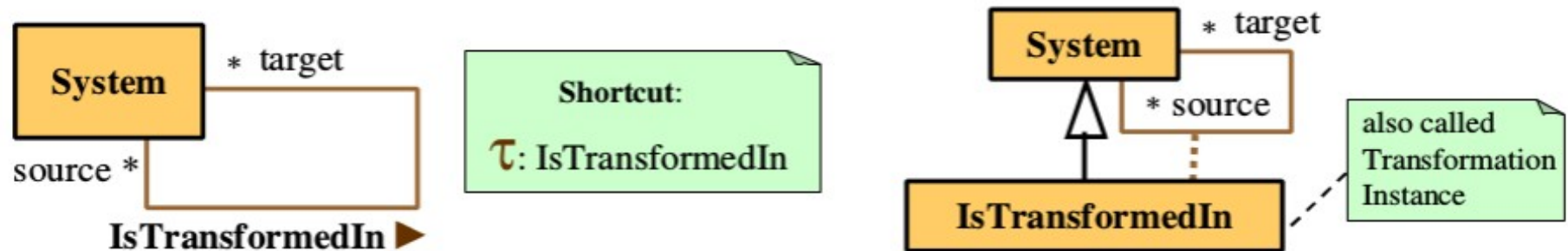
Researching Mega-Model
Background Info finding the...

...MDE Mega-Model



- Identifies 5 relations important to MDE.

MDE Mega-Model Transformations



[Favre et al.]

(... the 5th goes here.)

Transformation: $x \tau x'$
 Transformation Instance : (x, x')

Wait a minute...

Traceability Link == Relations && Transformation Instance == Relations

**Does
Transformation Instance == Traceability Link ???**

How come?

MegaL Traceability Recovery

- Recovered traceability links resemble transformation instances of τ

3.4 Traceability recovery

Traceability links may be recovered, for example, for conformance relationships and function application relationships (i.e., ‘transformations’). This is illustrated for the application of *ANTLRGenerator*. The input, *aGrammar*, is essentially a list of ANTLR rules with unique nonterminals on the left-hand sides. The output, *aParser*, is essentially a Java file exercising certain code patterns. In particular, for each nonterminal *n*, there is a corresponding method that implements the rule:

```
public final nContext n() throws RecognitionException { ... }
```

Thus, a suitable approach to traceability recovery is to retrieve nonterminals from the grammar and all relevant methods from the generated Java source and to check for a 1-1 correspondence; see [Figure 4](#) for illustration. For brevity, we show simplified evaluator code that only checks for correspondence, while the actual evaluator collects traceability links (i.e., pairs of URIs) of the following form:

```
< "http://.../MegaLParser.java/class/MegaLParser/method/megamodel/1" ,  
  "http://.../MegaL.g4/grammar/megal/rule/megamodel/1" >
```

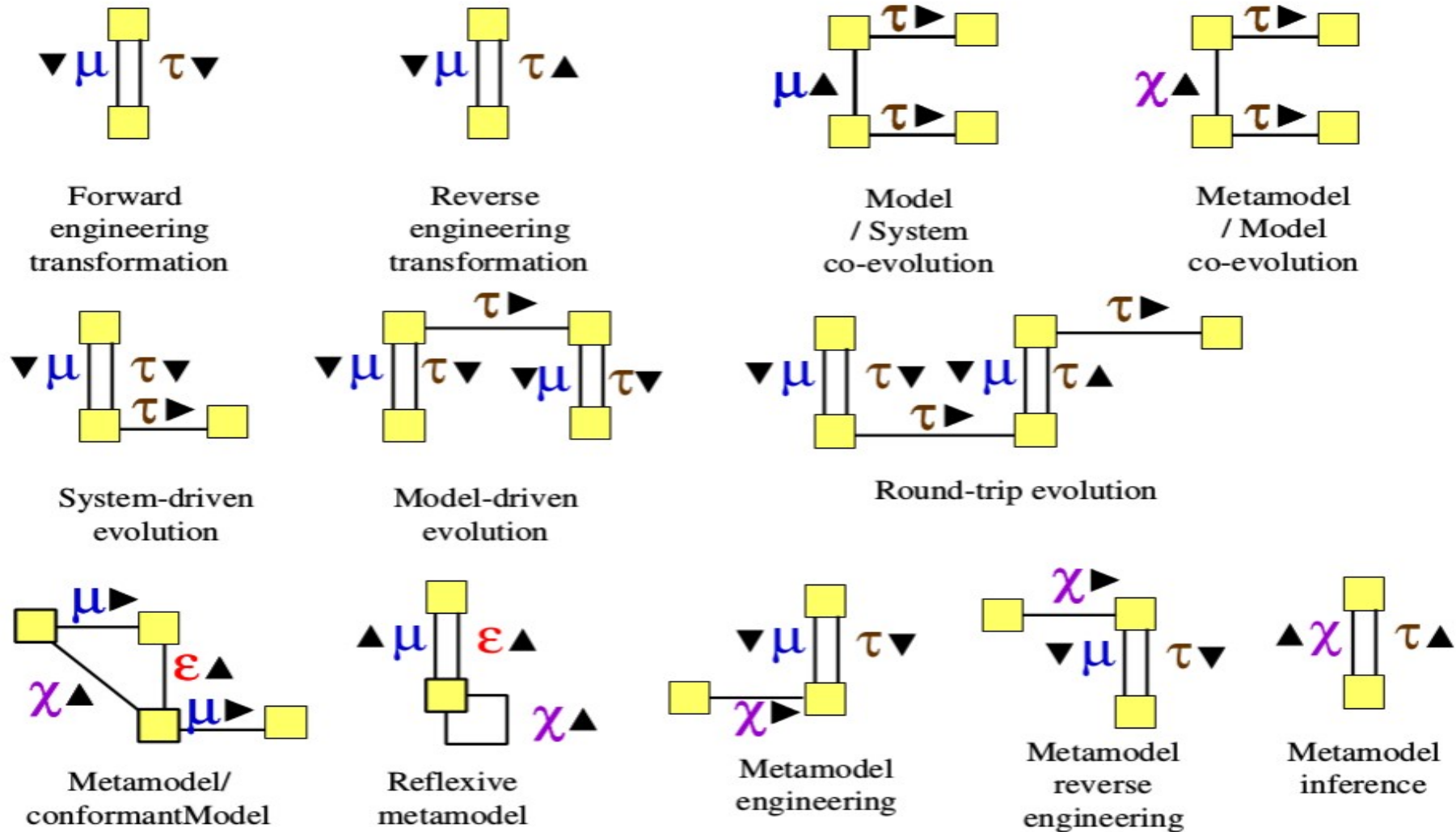
The URIs describe the relevant fragments in a language-parametric manner. That is, the URIs start with the actual resource URI for the underlying artifact. The rest of the URI, which is underlined for clarity, describes the access path to the relevant fragment. To this end, syntactical categories of the artifact’s language (see ‘class’ and ‘method’ versus ‘rule’) and names of abstractions (see ‘megamodel’) are used. (We note that ‘megamodel’ is the first nonterminal, in fact, the startsymbol of the grammar for *MegaL*.)

Hidden Traceability Links?

Relation	Instance
$x \varepsilon y$	(x,y)
$x \mu y$	(x,y)
$x \delta y$	(x,y)
$x \chi y$	(x,y)
$x \tau x'$	(x,x')

(In a set theoretical sense.)

Hidden Mega- (Traceability-) Patterns?



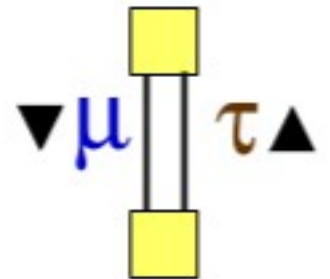
[Favre et al.]

Evidence?

MegaL Traceability Recovery

- Function Application: *ANTLR.Generator*
- Recovered Traceability Link: **(aParser, aGrammar)**
- Would imply:

- $aParser \mu aGrammar \&\& aGrammar \tau aParser$
- \Rightarrow Reverse engineering transformation Pattern



Reverse
engineering
transformation