

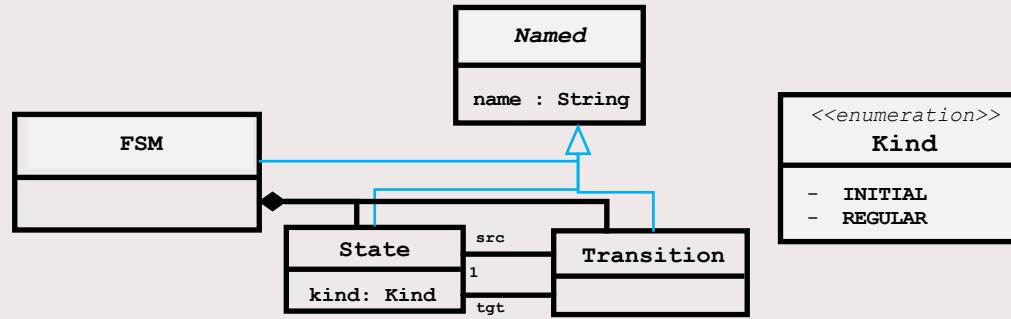


# Co-Evolving Meta-Models and View Types in View-Based Development

3 OCTOBER 2023

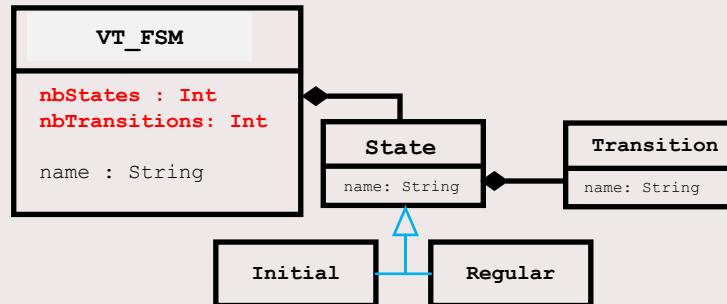
- ▶ **Hossain Muhammad Muctadir**, Eindhoven University of Technology, Netherlands
- Lars König**, Karlsruhe Institute of Technology, Germany
- Thomas Weber**, Karlsruhe Institute of Technology, Germany
- Moussa Amrani**, University of Namur, Belgium
- Loek Cleophas**, Eindhoven University of Technology, Netherlands & Stellenbosch University, South Africa

# Define view type and view based on MM



## Metamodel

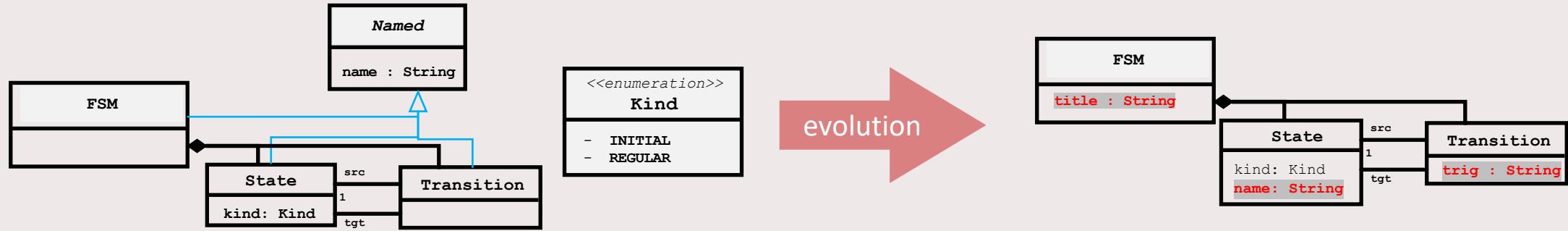
### View Type



fsm "Simple FSM"  
number of states = 4  
number of transitions = 6  
state "1" initial  
transition "a" to "2"  
endstate  
state "2"  
transition "b" to "1"  
transition "c" to "3"  
endstate  
...  
endfsm

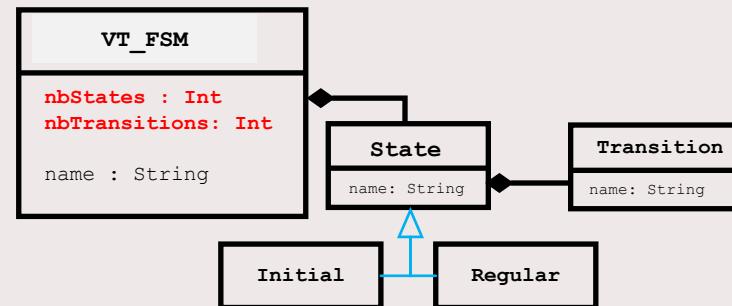
### View

# How to co-evolve view types when MMs evolve?



## Metamodel

### View Type



```
fsm "Simple FSM"
number of states      = 4
number of transitions = 6
state "1" initial
    transition "a" to "2"
endstate
state "2"
    transition "b" to "1"
    transition "c" to "3"
endstate
...
endfsm
```

### View

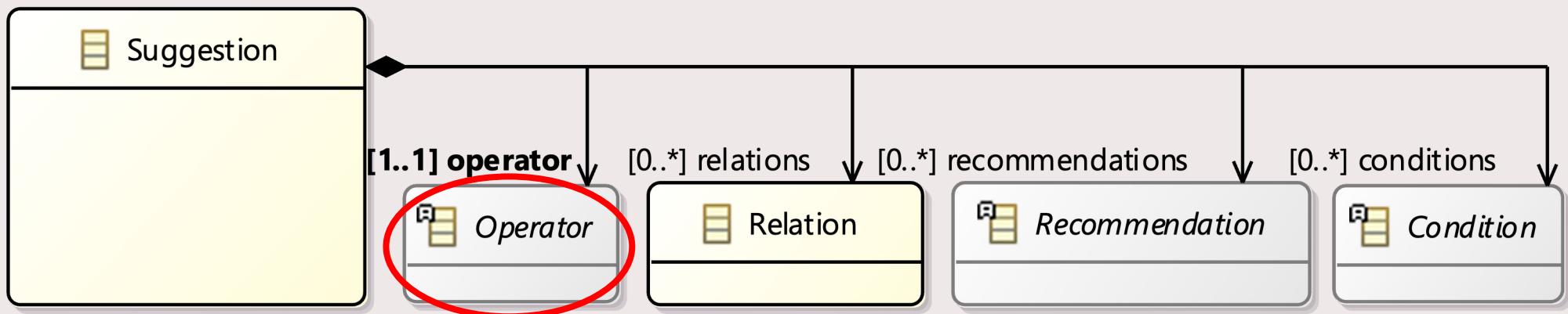
# Contribution

1. A domain-independent **approach** and **conceptual model** for meta-model/view type co-evolution
2. A **catalog of suggestions** for the most important evolution operators

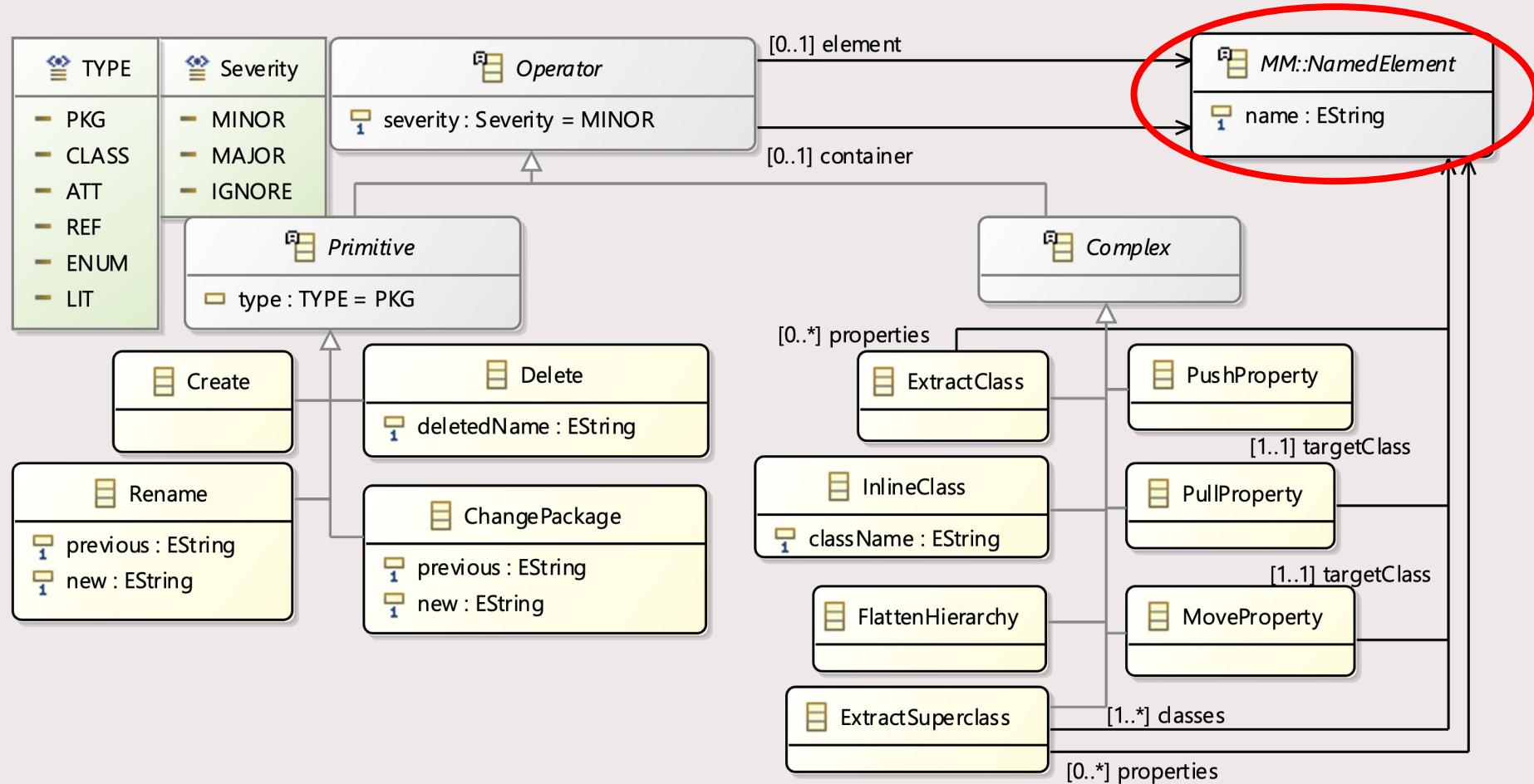
We illustrate this approach on a simple but **representative example**.



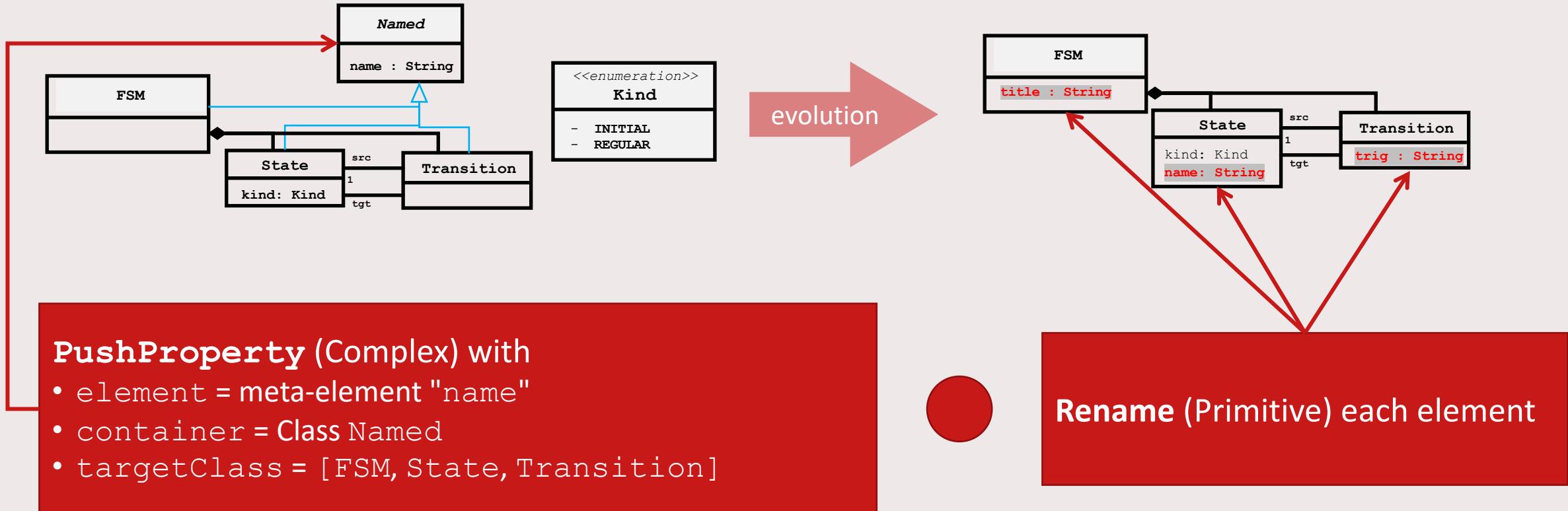
# A metamodel for Suggestions



# Capturing Operators for evolution



# Adding (rudimentary) guards/accepting State to FSMs



# Collecting evolution operators from literature

## An Extensive Catalog of Operators for the Coupled Evolution of Metamodels and Models

Markus Herrmannsdoerfer<sup>1</sup>, Sander D. Vermolen<sup>2</sup>, and Guido Wachsmuth<sup>2</sup>

<sup>1</sup> Institut für Informatik,  
Technische Universität München  
Boltzmannstr. 3, 85748 Garching b. München, Germany  
[herrmama@in.tum.de](mailto:herrmama@in.tum.de)

<sup>2</sup> Software Engineering Research Group,  
Delft University of Technology  
The Netherlands  
[{S.D.Vermolen,G.H.Wachsmuth}@tudelft.nl](mailto:{S.D.Vermolen,G.H.Wachsmuth}@tudelft.nl)

## Detecting Complex Changes During Metamodel Evolution

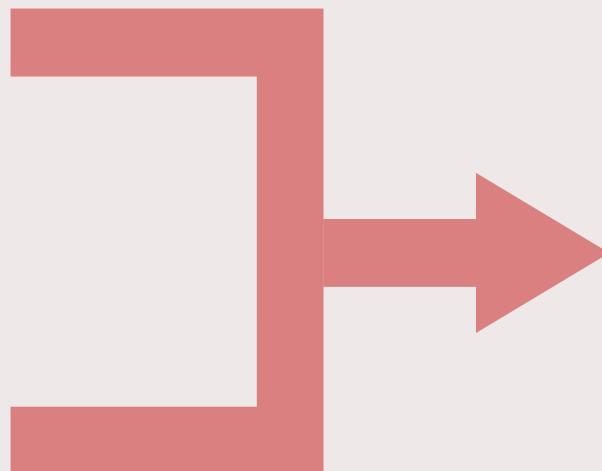
Djamel Eddine Khelladi<sup>1(✉)</sup>, Regina Hebig<sup>1</sup>, Reda Bendraou<sup>1</sup>,  
Jacques Robin<sup>1</sup>, and Marie-Pierre Gervais<sup>1,2</sup>

<sup>1</sup> Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, F-75005 Paris, France  
[djamel.khelladi@lip6.fr](mailto:djamel.khelladi@lip6.fr)

<sup>2</sup> Université Paris Ouest Nanterre La Défense, F-92001 Nanterre, France

## 61 evolution Operators

- 27 primitive**
- 34 complex



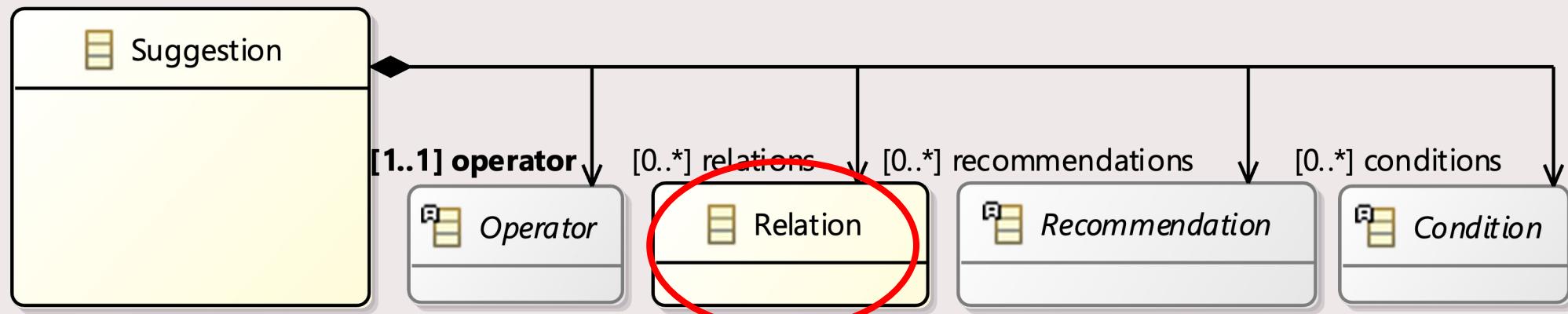
## 34 evolution Operators

- 27 primitive
- 7 complex

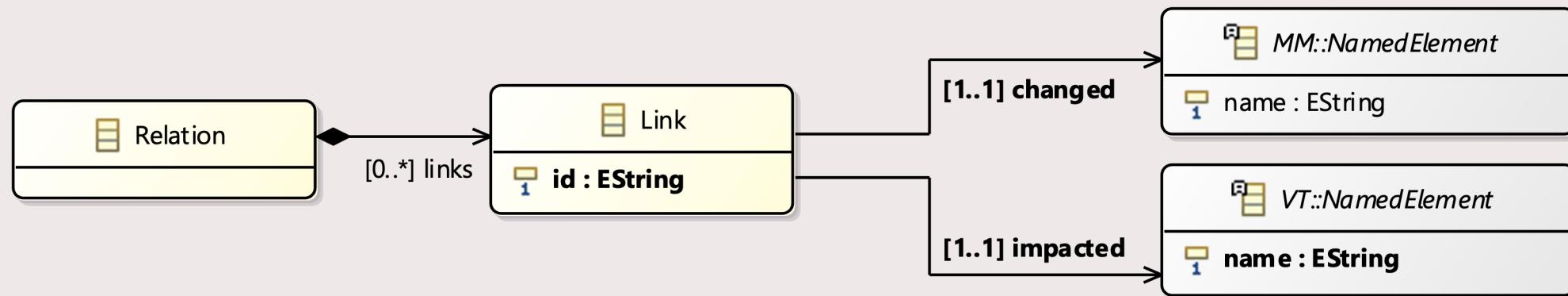
## 7 Complex operators

- Covering 72% of complex changes

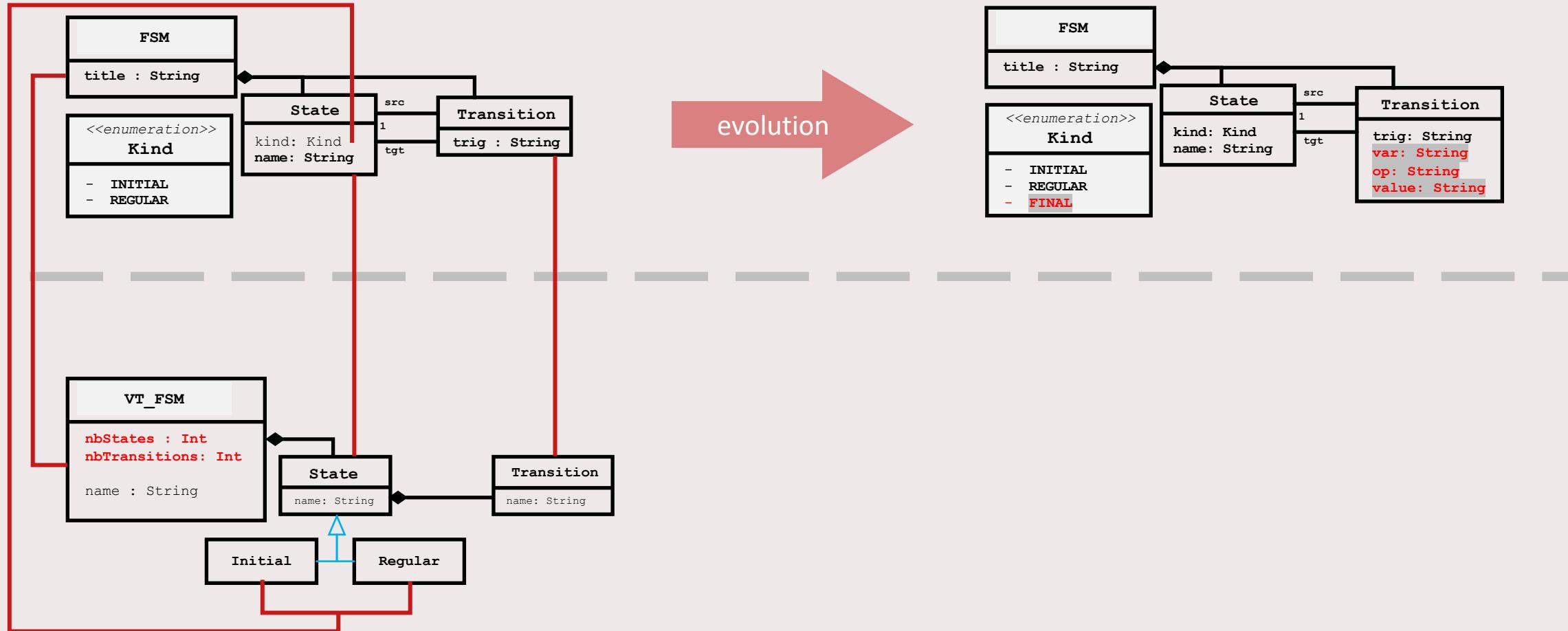
# A Metamodel for Suggestions



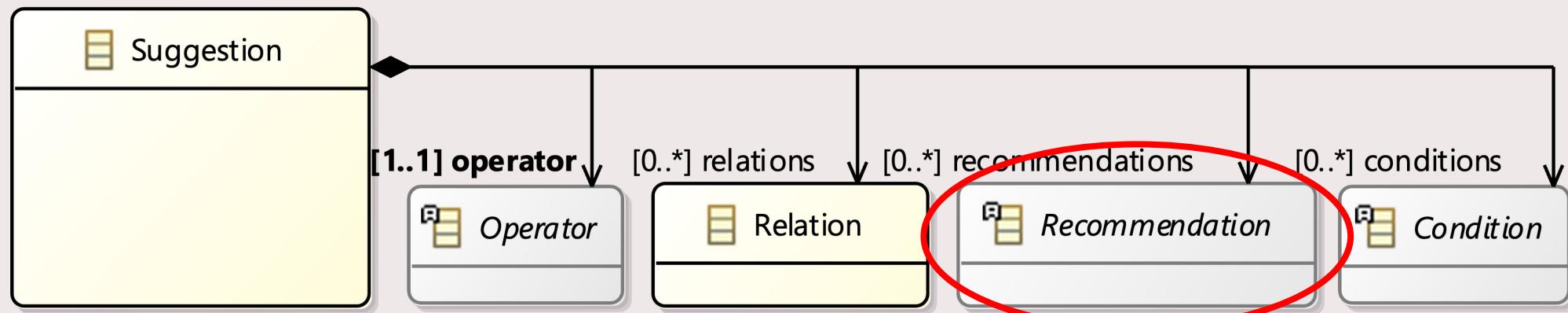
# Linking MM elements with impacted VT elements



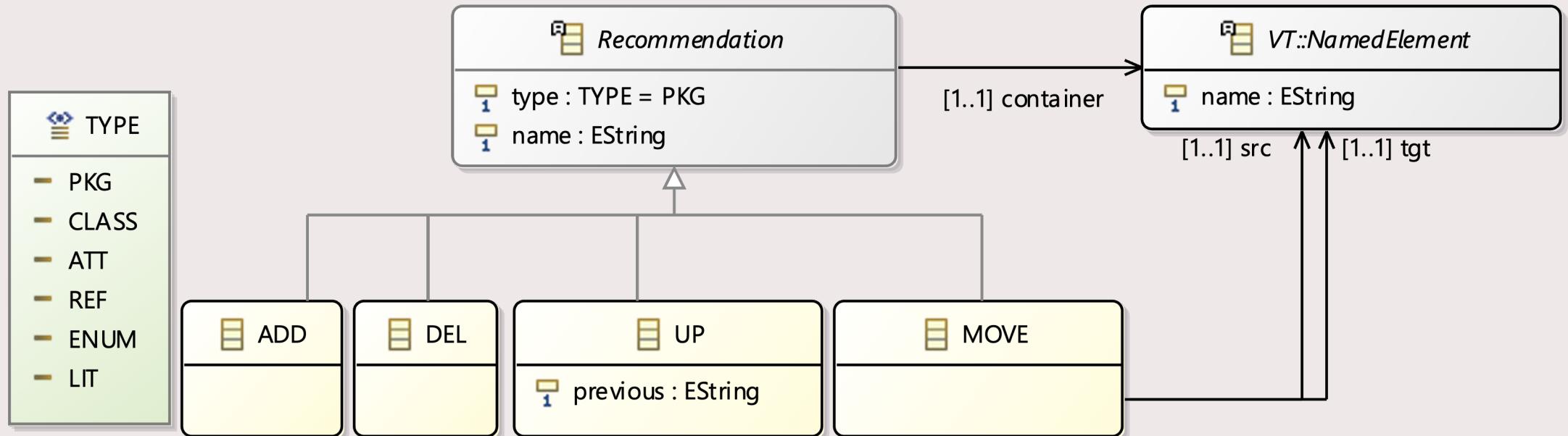
# Finding impacted elements of Create



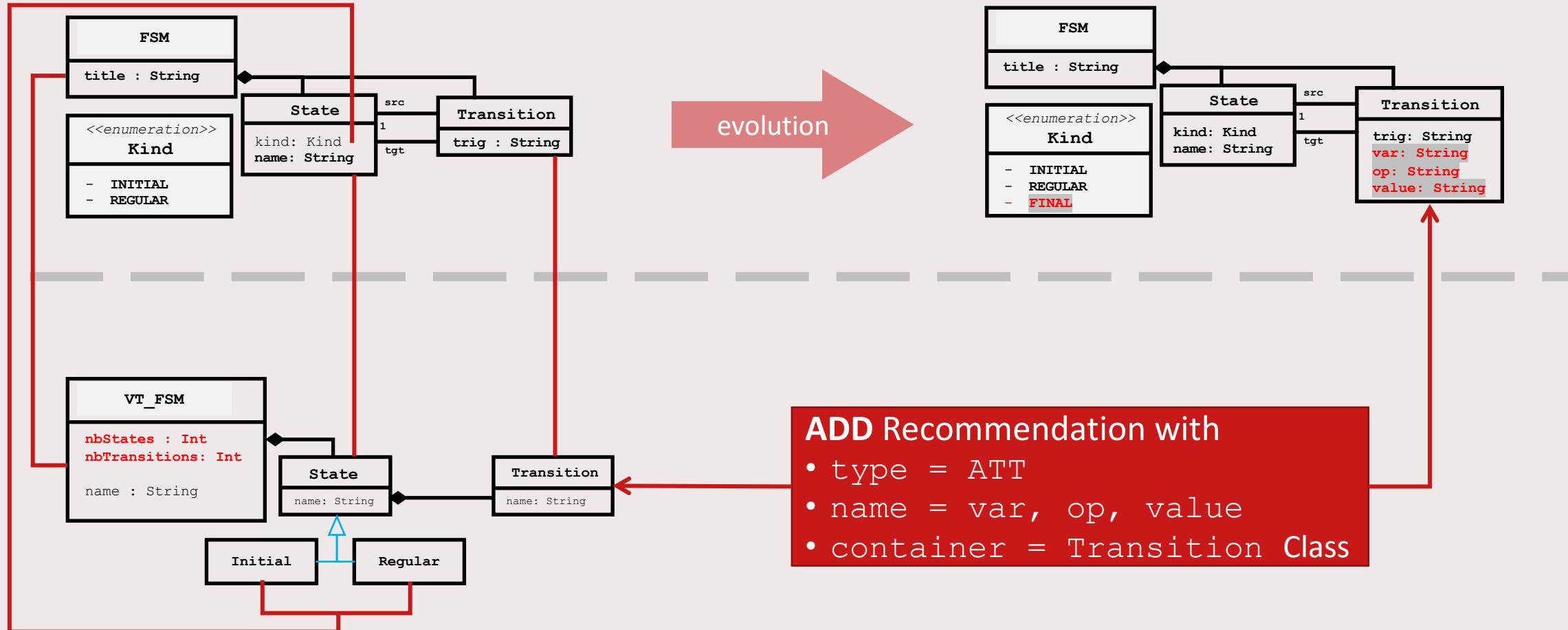
# A Metamodel for Suggestions



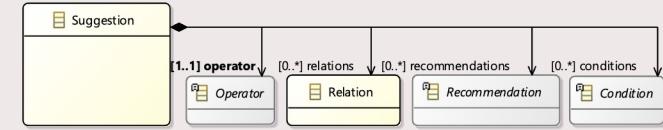
# Recommending actions for VT co-evolution



# Finding impacted elements of Create



# Catalog of suggestions



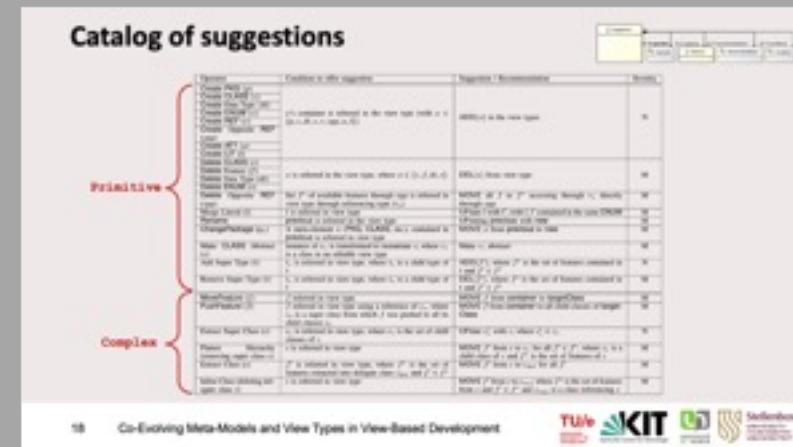
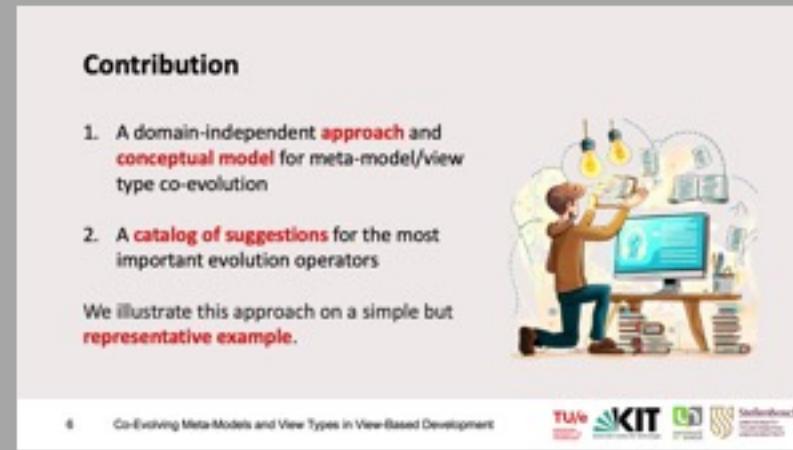
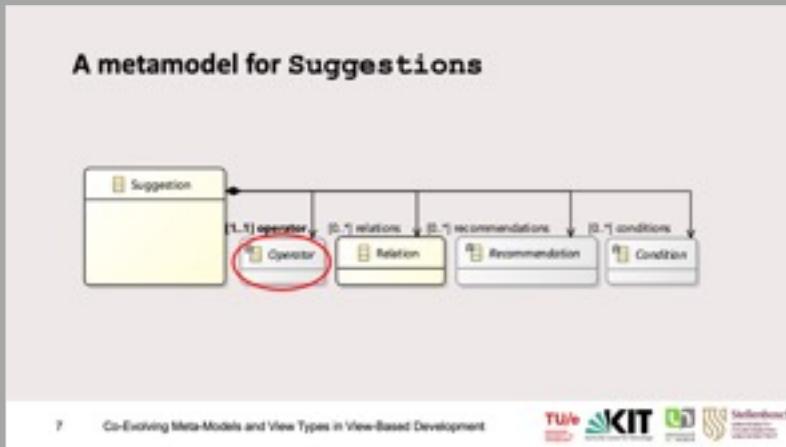
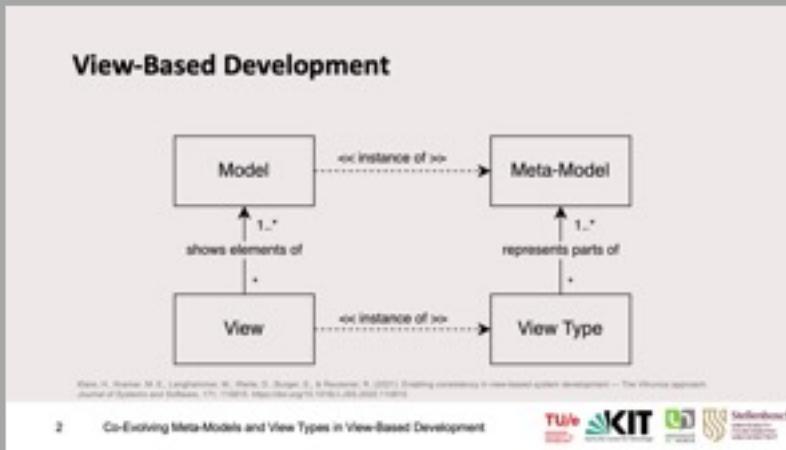
Operator	Condition to offer suggestion	Suggestion / Recommendation	Severity
Create PKG ( <i>p</i> )			
Create CLASS ( <i>c</i> )			
Create Data Type ( <i>dt</i> )			
Create ENUM ( <i>e</i> )			
Create REF ( <i>r</i> )			
Create Opposite REF ( <i>opp</i> )			
Create ATT ( <i>a</i> )			
Create LIT ( <i>l</i> )			
Delete CLASS ( <i>c</i> )			
Delete Feature ( <i>f</i> )			
Delete Data Type ( <i>dt</i> )			
Delete ENUM ( <i>e</i> )			
Delete Opposite REF ( <i>opp</i> )	Set $f^*$ of available features through <i>opp</i> is referred in view type through referencing type ( $r_s$ )	MOVE all $f$ in $f^*$ accessing through $r_s$ directly through <i>opp</i>	M
Merge Literal ( <i>l</i> )	$l$ is referred in view type	UPdate $l$ with $l'$ , with $l, l'$ contained in the same ENUM	M
Rename	previous is referred in view type	UPdating previous with new	M
ChangePackage ( <i>po</i> )	A meta-element $x$ (PKG, CLASS, etc.), contained in previous is referred in view type	MOVE $x$ from previous to new	M
Make CLASS Abstract ( <i>c</i> )	instance of $c_v$ is transformed to instantiate $c$ , where $c_v$ is a class in an editable view type	Make $c_v$ abstract	M
Add Super Type ( <i>t</i> )	$t_c$ is referred in view type, where $t_c$ is a child type of <i>t</i>	ADD( $f'$ ), where $f^*$ is the set of features contained in <i>t</i> and $f' \in f^*$	N
Remove Super Type ( <i>t</i> )	$t_c$ is referred in view type, where $t_c$ is a child type of <i>t</i>	DEL( $f'$ ), where $f^*$ is the set of features contained in <i>t</i> and $f' \in f^*$	M
MoveFeature ( <i>f</i> )	<i>f</i> referred in view type	MOVE <i>f</i> from container to targetClass	M
PushFeature ( <i>f</i> )	<i>f</i> referred in view type using a reference of $c_s$ , where $c_s$ is a super class from which <i>f</i> was pushed to all its child classes $c_c$	MOVE <i>f</i> from container to all child classes of target-Class	M
Extract Super Class ( <i>c</i> )	$c_c$ is referred in view type, where $c_c$ is the set of child classes of <i>c</i>	UPdate $c'_c$ with <i>c</i> , where $c'_c \in c_c$	N
Flatten Hierarchy (removing super class <i>c</i> )	<i>c</i> is referred in view type	MOVE $f'$ from <i>c</i> to $c_c$ for all $f' \in f^*$ , where $c_c$ is a child class of <i>c</i> and $f^*$ is the set of features of <i>c</i>	M
Extract Class ( <i>c</i> )	$f'$ is referred in view type, where $f^*$ is the set of features extracted into delegate class $c_{del}$ and $f' \in f^*$	MOVE $f'$ from <i>c</i> to $c_{del}$ for all $f'$	M
Inline Class (deleting delegate class <i>c</i> )	<i>c</i> is referred in view type	MOVE $f'$ from <i>c</i> to $c_{ref}$ where $f^*$ is the set of features from <i>c</i> and $f' \in f^*$ and $c_{ref}$ is a class referencing <i>c</i>	M

# Discussion

- Suggestions closely follow the operators
- Suggestions can be extended for view generation transformations
- FSM example too simplistic



**Thanks for your attention**



**✉ h.m.muctadir@tue.nl**

 muctadir.github.io