

A DMSL for System Capability Diagram

MODEL DRIVEN SOFTWARE ENGINEERING PROJECT

Arthur Milani Giovanini
Gustavo Palma dos Santos
27 November, 2024

Group 3 

S U M M A R Y

- The System Capability Diagram
- Metamodel
- Abstract & concrete syntax
- Relations
- Viewpoint specification (.odesign)
- Demo time!

THE SYSTEM CAPABILITY DIAGRAM

- Define system capabilities
- Connects actors, entities, and capabilities
- Analyzes relationships between capabilities

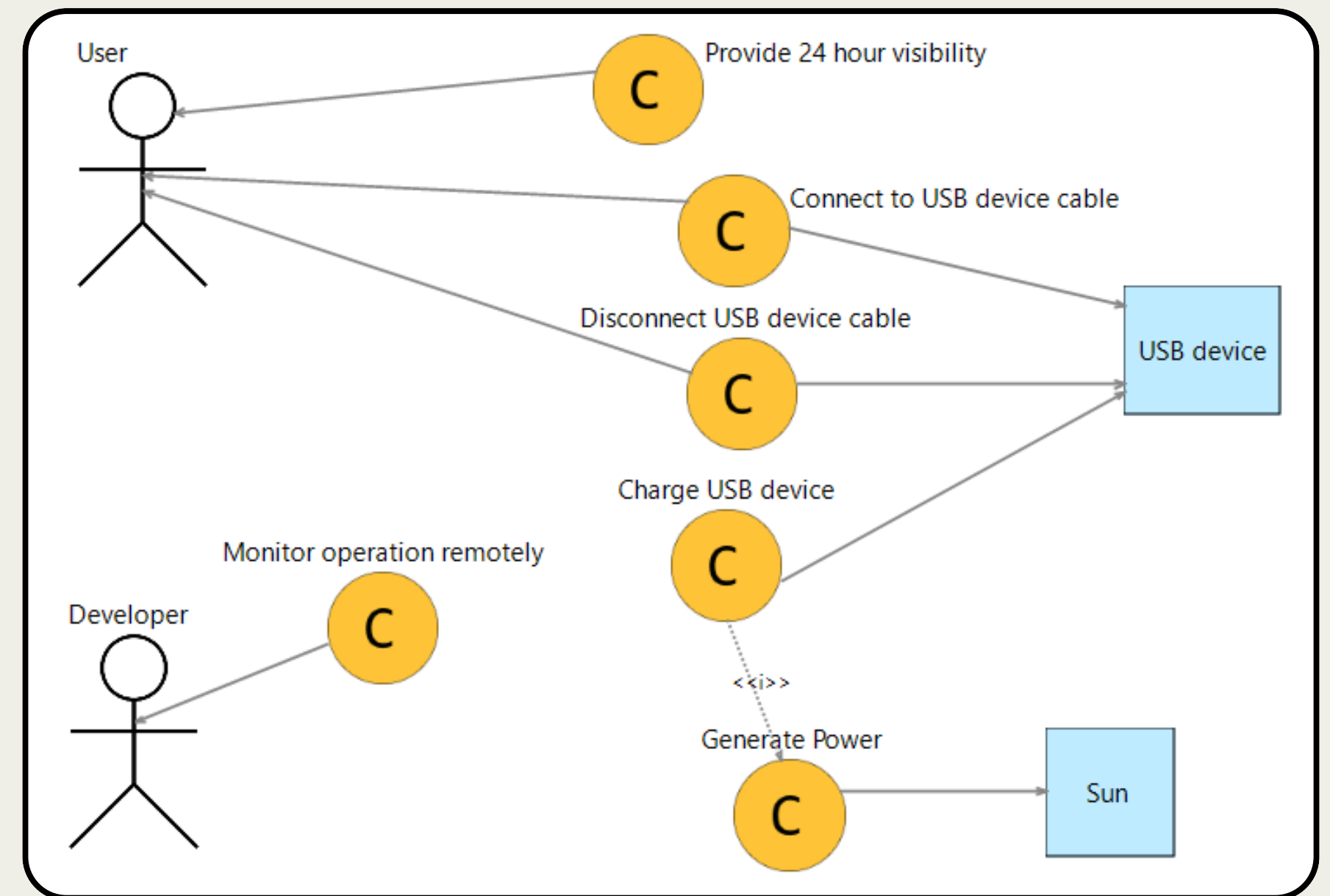


Fig. 1: Trail Power model

METAMODEL

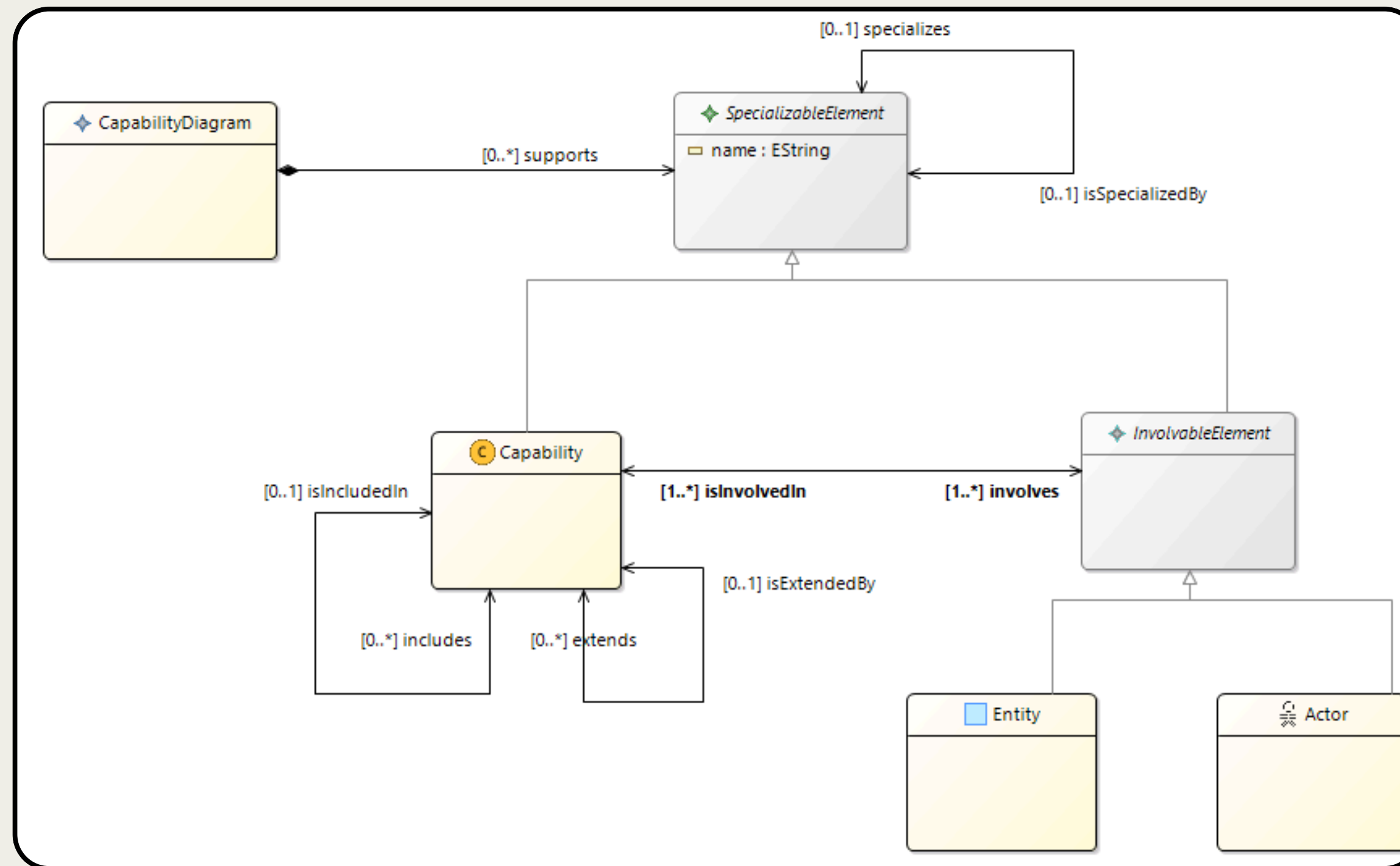
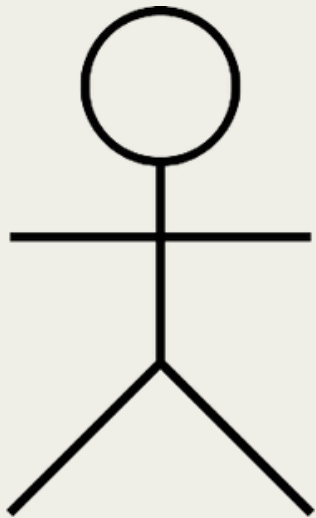
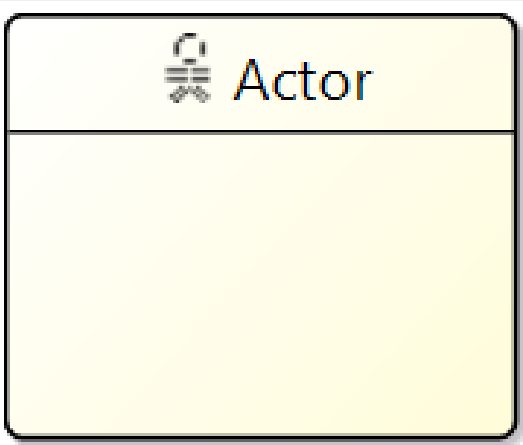
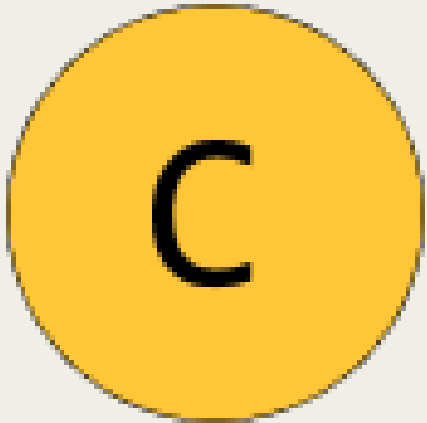
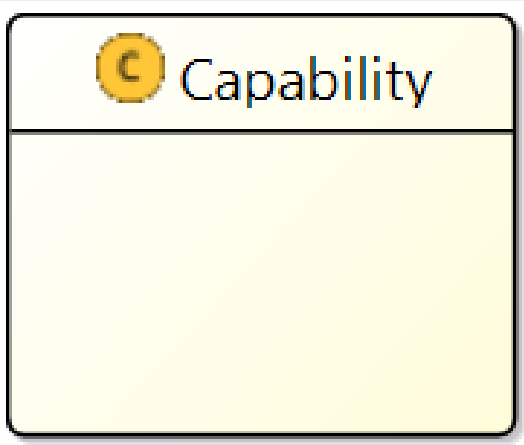


Fig. 2: System Capability Diagram metamodel

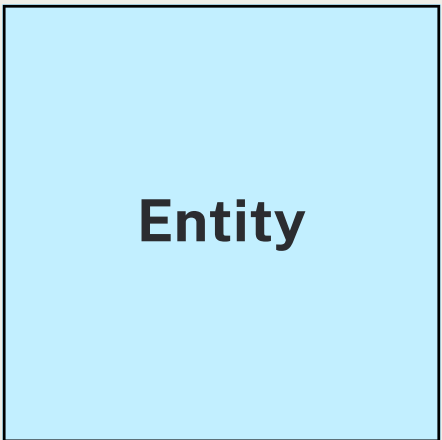
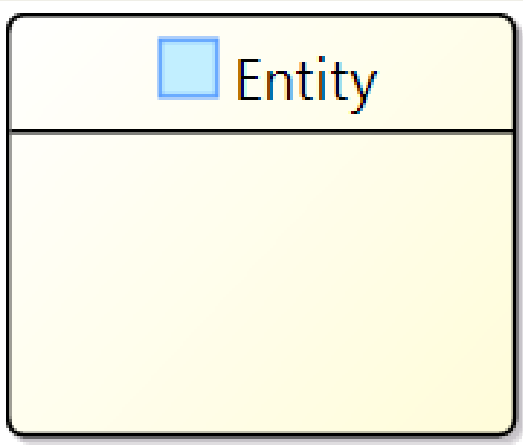
ABSTRACT & CONCRETE SYNTAX



Actor










Capability



Entity

RELATIONS

Relation	Origin	Target
Involves		 
Specializes	All elements	The same type as origin
Includes		
Extends		

Tab. 1: System Capability Diagram relations

VIEWPOINT SPECIFICATION (.ODESIGN)

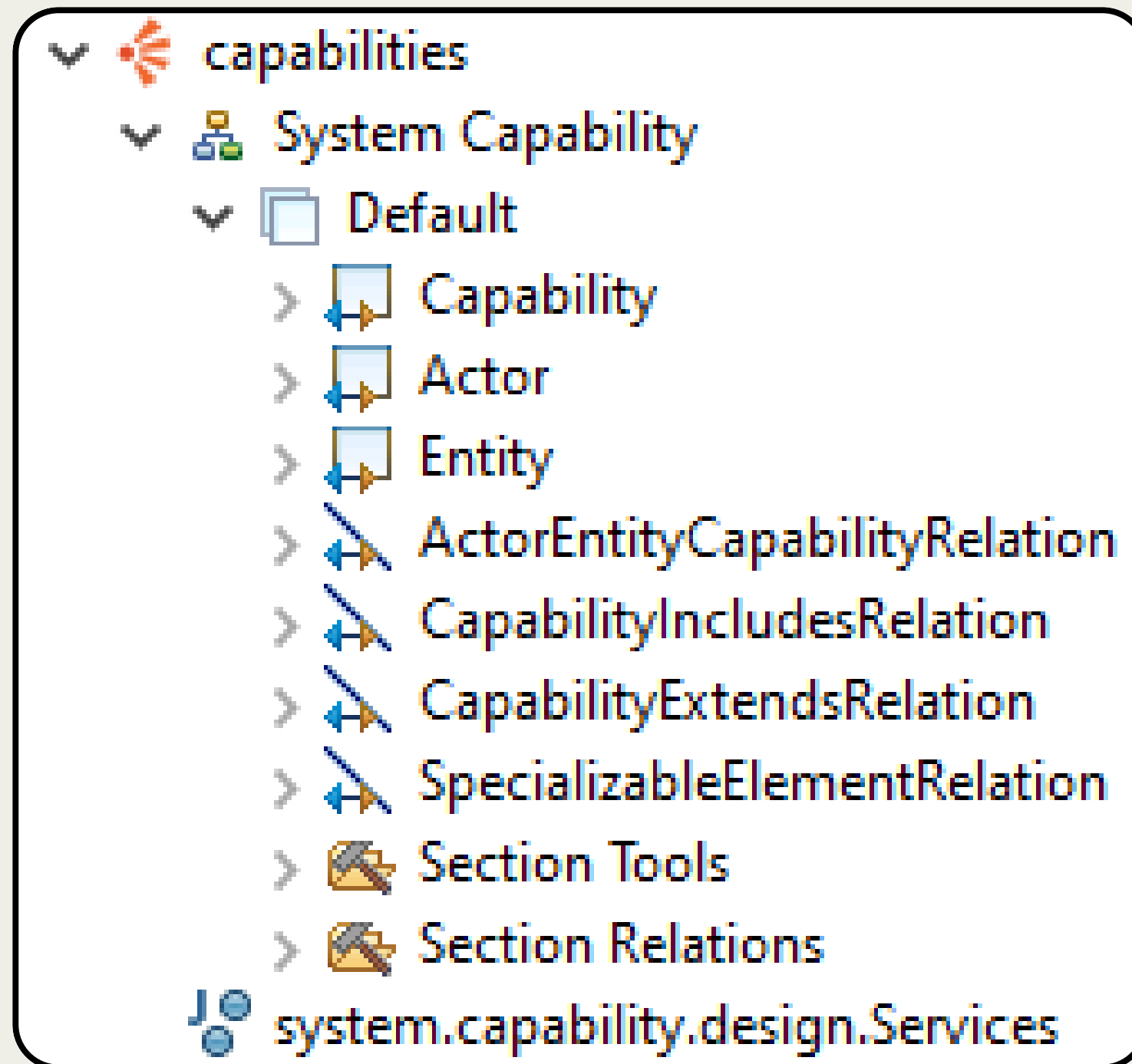


Fig. 3: Mapping implementation of the concrete syntax

VIEWPOINT SPECIFICATION (.ODESIGN)

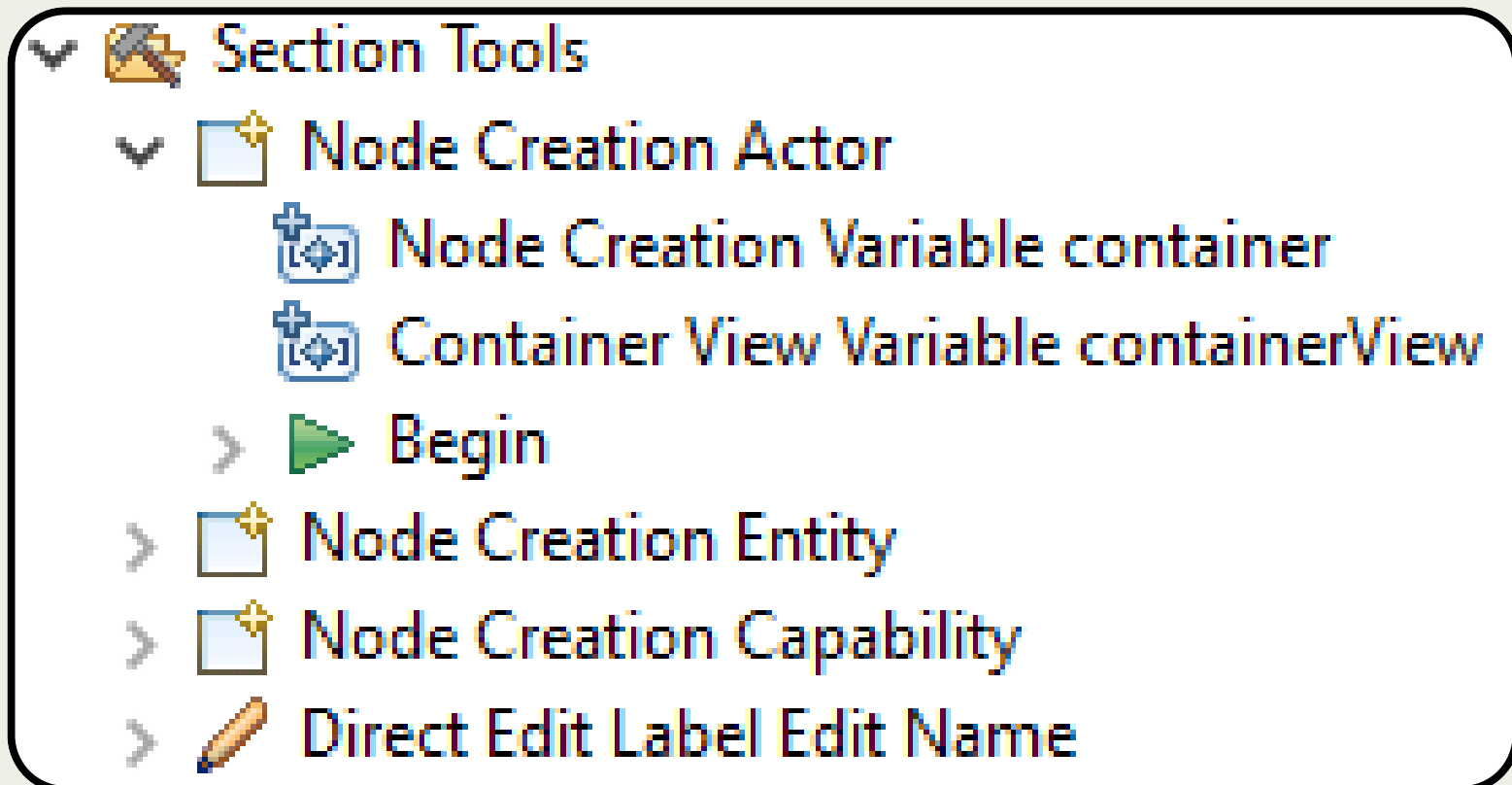


Fig. 4: Tools to create nodes

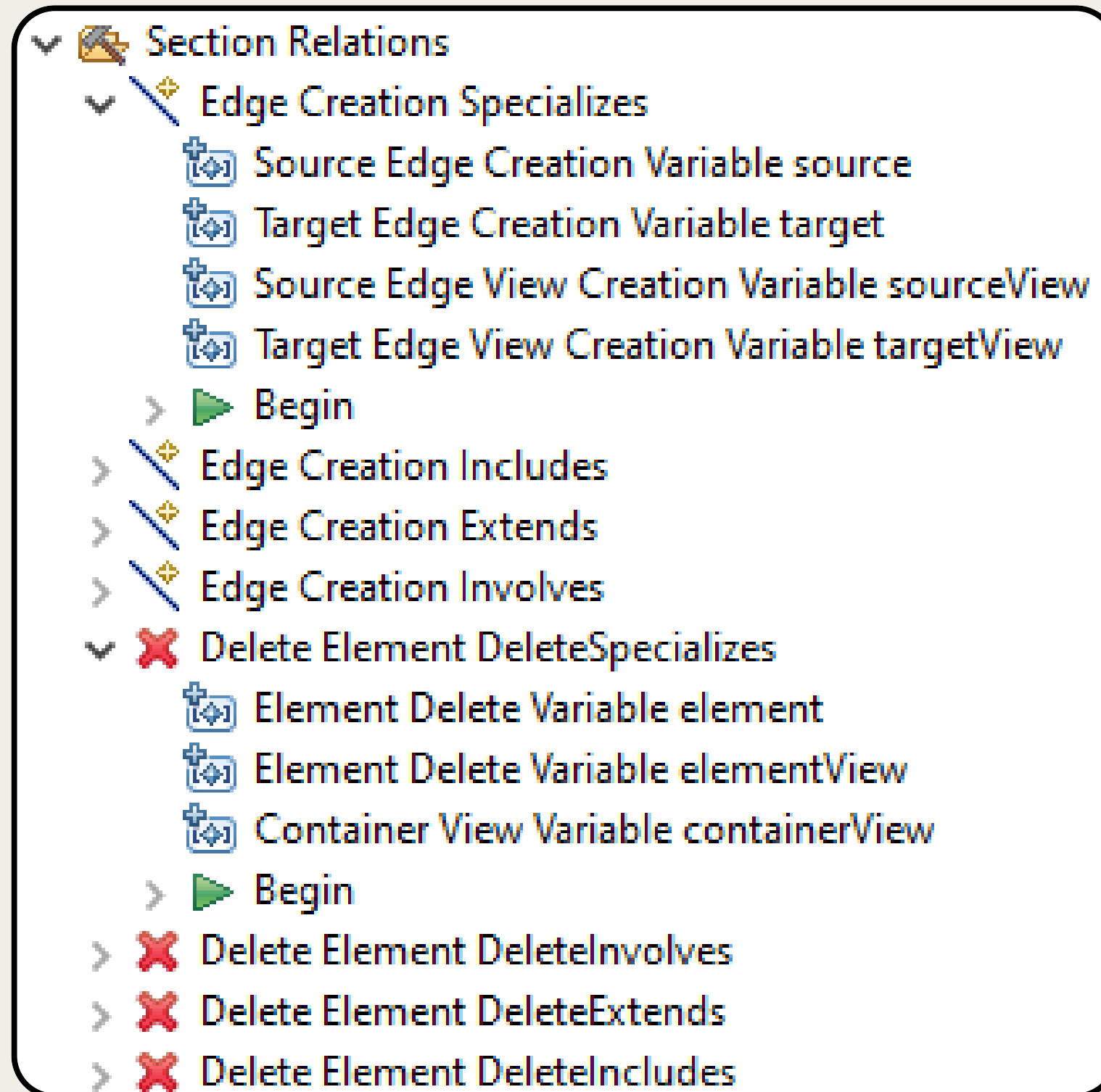


Fig. 5: Tools to create and delete edges

VIEWPOINT SPECIFICATION (.ODESIGN)



- ▼  System Capability
 - ▼  Validation
 - > ☒ Semantic Validation Rule Actor Specialization
 - > ☒ Semantic Validation Rule Entity Specialization
 - > ☒ Semantic Validation Rule Capability Specialization
 - > ☒ Semantic Validation Rule Irreflexive Capability Includes
 - > ☒ Semantic Validation Rule Irreflexive Capability Extends
 - > ☒ Semantic Validation Rule Asymetric Capability Extends
 - > ☒ Semantic Validation Rule Asymetric Capability Includes

Fig. 6: Validations to cover the OCL constraints

Demo time!

