# The SDL-UML Profile Revisited

Alexander Kraas
04. October 2010



Fraunhofer ESK

# OUTLINE

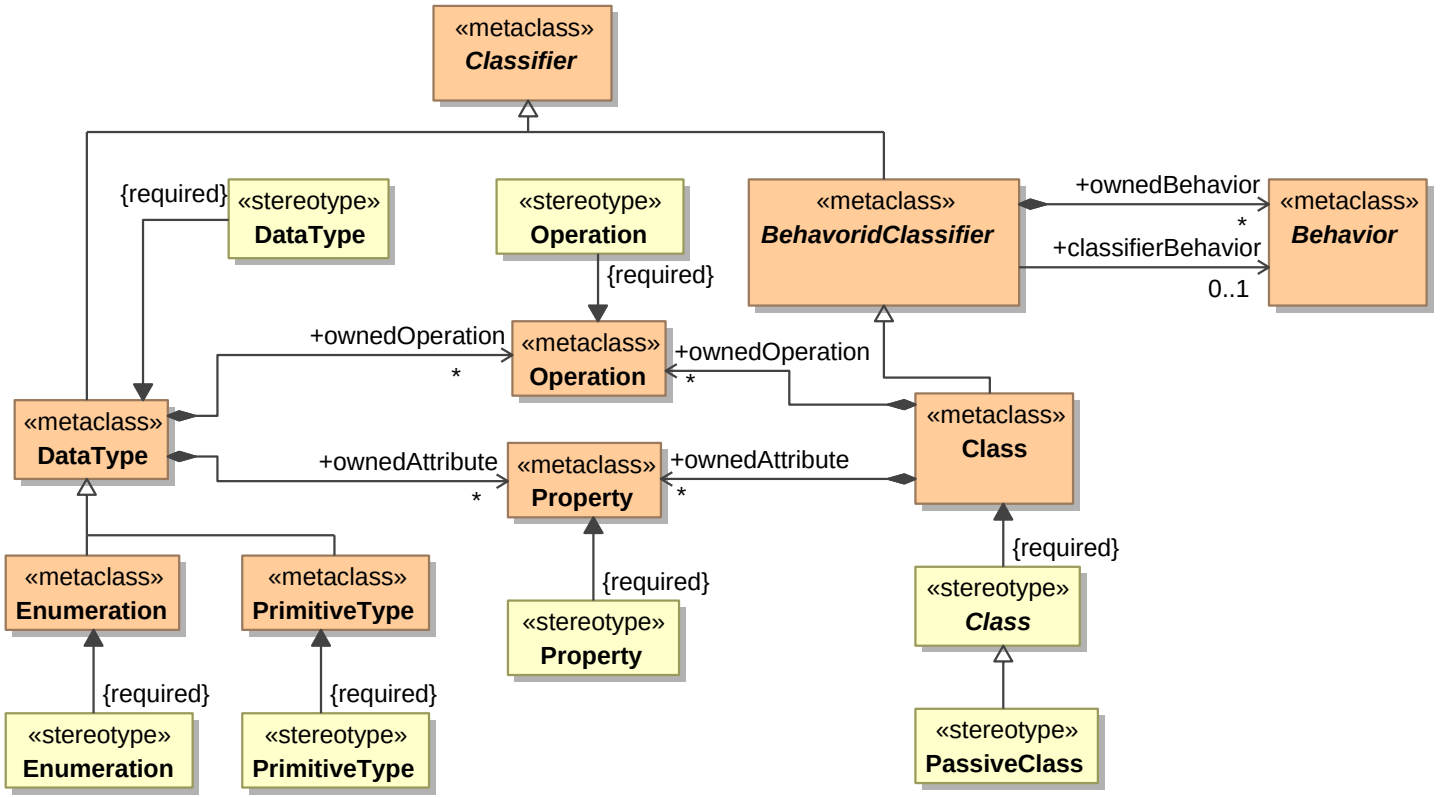
1. Motivation
2. Data Type Model
3. Value Specification Model
4. Conclusion & Future Work

# 1. Motivation

- The SDL-UML profile shall make it possible to specify SDL systems with UML.
- The most recent version of the SDL-UML profile was published by the ITU-T in June 2007.
- A case study in 2008 had shown that this profile version has some shortcomings and limitations.
- Some minor shortcomings was reported to the ITU-T SG17/Q.13 in order to remedy them in the current version of the Z.109 rec.
- Discovered limitations related to the current data type model and the value specification model of SDL-UML were not included in that report.
- It is planed to publish a new version of the SDL-UML Profile for SDL-2010.
- We propose new models for data type and value specification for the upcoming profile version.

# 2. Data Type Model
## Limitations of the actual model

1. Deletion of object data types in SDL-2010
2. Behavior of operations
3. No Support for choice data type
4. No support for optional fields in a structured data type

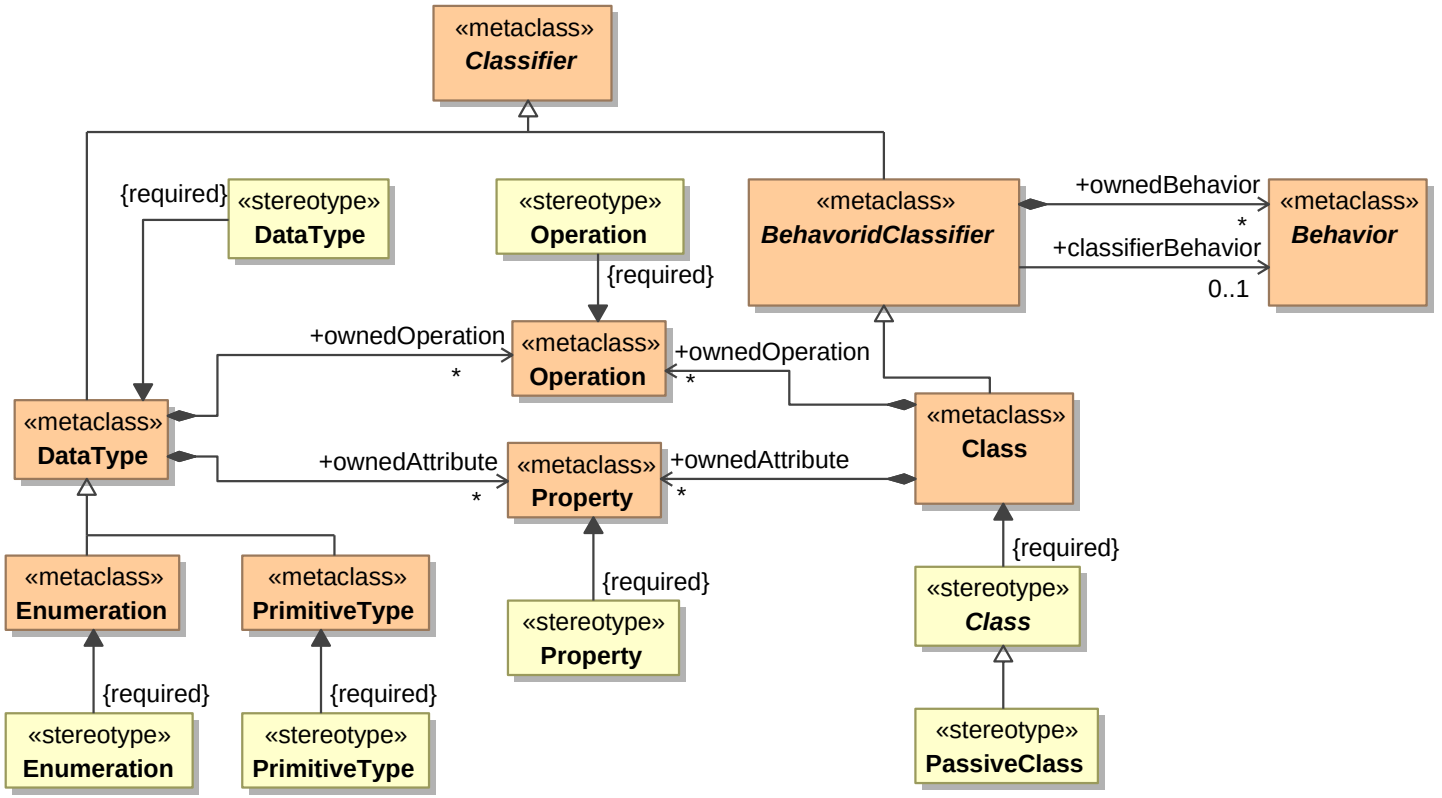





Fraunhofer ESK

# 2. Data Type Model
## Limitations of the actual model

5. No distinction between static and dynamic operations
6. Missing support for syntypes
7. Restricted possibility to specify literal signatures

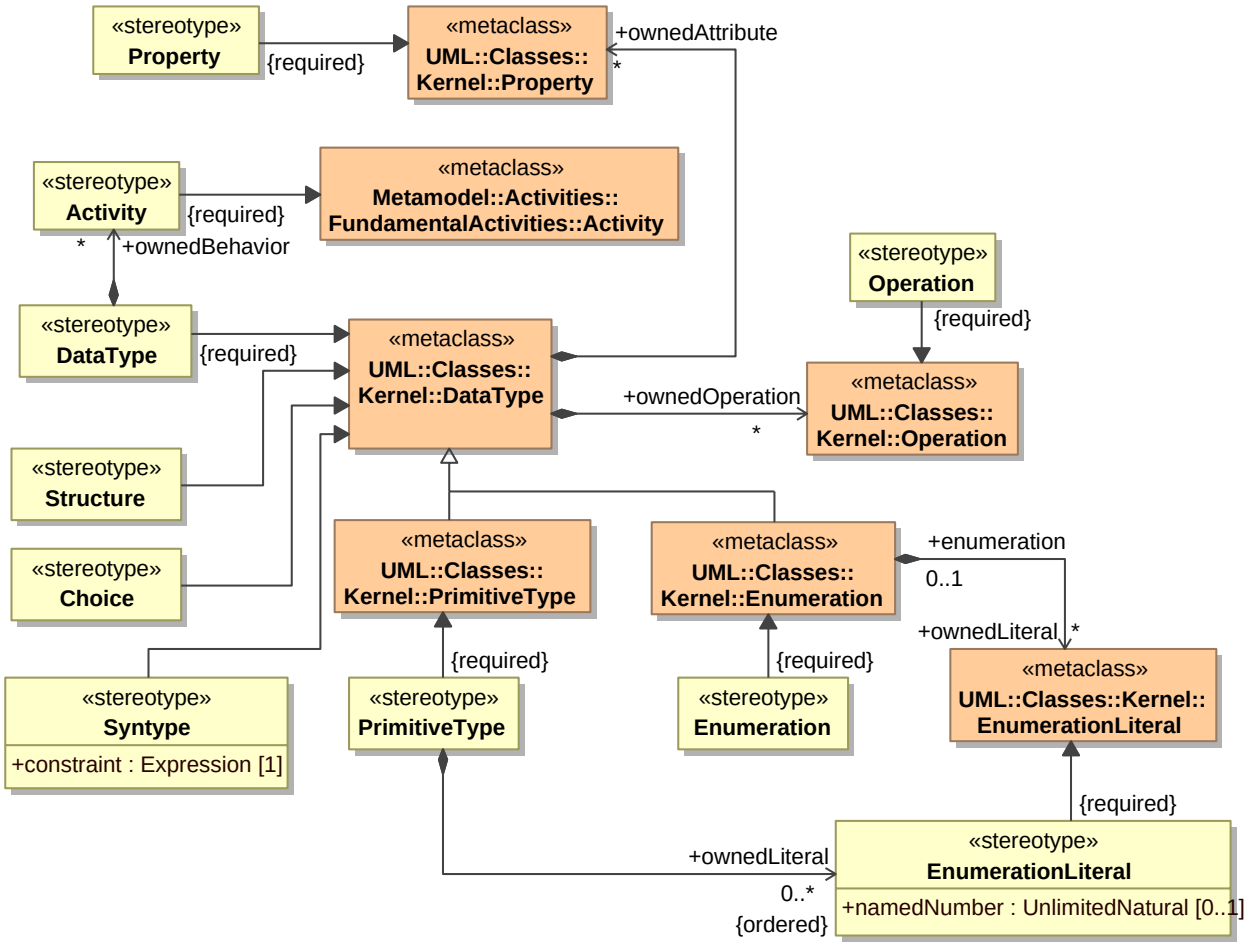





Fraunhofer ESK

# 2. Data Type Model
## Proposed model

1. <<PassiveClass>> Class stereotype is not any longer present.
2. The <<Choice>> DataType represents a SDL choice data type.
3. The <<Syntype>> DataType introduces the feature of syntype specification in SDL-UML.





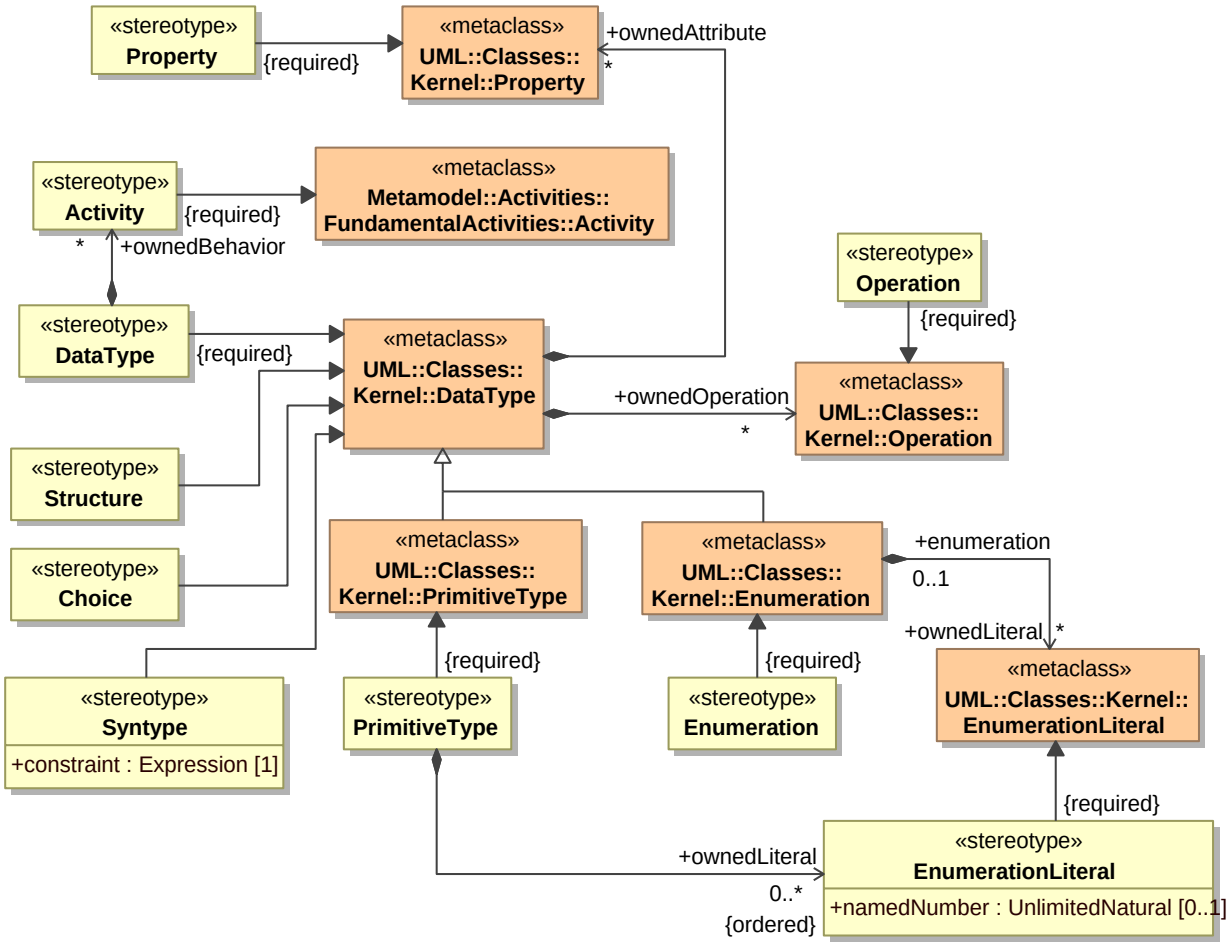

Fraunhofer ESK

# 2. Data Type Model
# Proposed model

4. The <<Structure>> stereotype represents a SDL structure type.

5. NamedNumber tagged value of the <<Enumer-ationLiteral>> stereotype represents SDL named numbers.

6. OwnedBehavior tagged value of the <<Data-Type>> stereotype shall reference an Activity used as the method of an <<Operation>>.

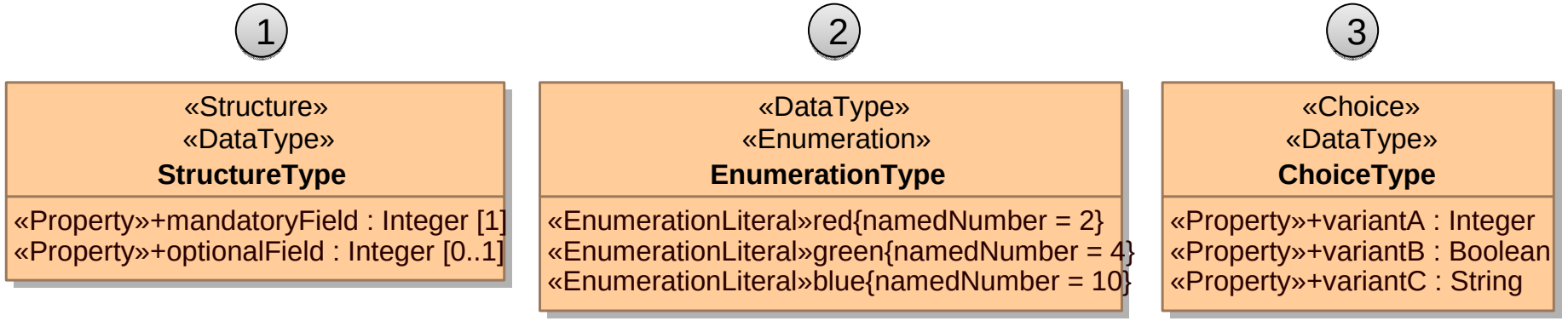





Fraunhofer ESK

# 2. Data Type Model
## Application examples

1. <<Structure>> DataType example:
   - A “mandatoryField” shall have a multiplicity of [1]
   - A multiplicity of [0..1] represents an “optionalField”.
2. <<Enumeration>> Enumeration example:
   - The namedNumber of <<EnumerationLiteral>> is used to associate a literal with a particular value.
   - E.g. the literal “red” is associated with the literal value 2

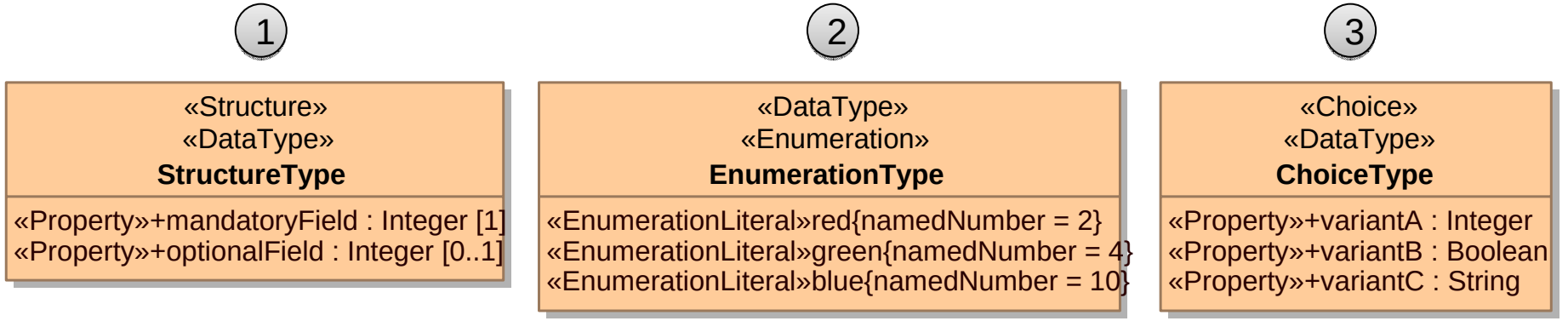





Fraunhofer ESK

# 2. Data Type Model
## Application examples

3. <<Choice>> DataType example:
   - Different variants of a choice data type are specified by the attributes of a <<Choice>> DataType.
   - E.g. The “ChoiceType” (example 3) consists of three different variants.

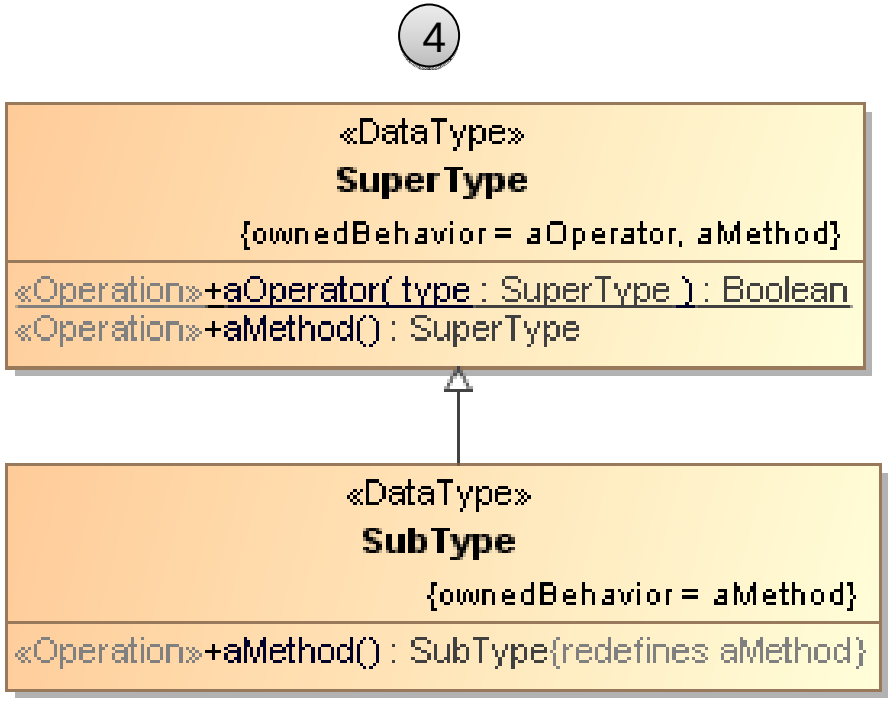





Fraunhofer ESK

# 2. Data Type Model
## Application examples

4. Operations of data types:
   - A static <<Operation>> Operation represent an SDL operator.
   - A SDL method is represented by a non-static <<Operation>> Operation.
   - The method of an Operation is stored in the scope of the ownedBehavior tagged value.
   - Only non-static <<Operation>> Operations can be redefined.

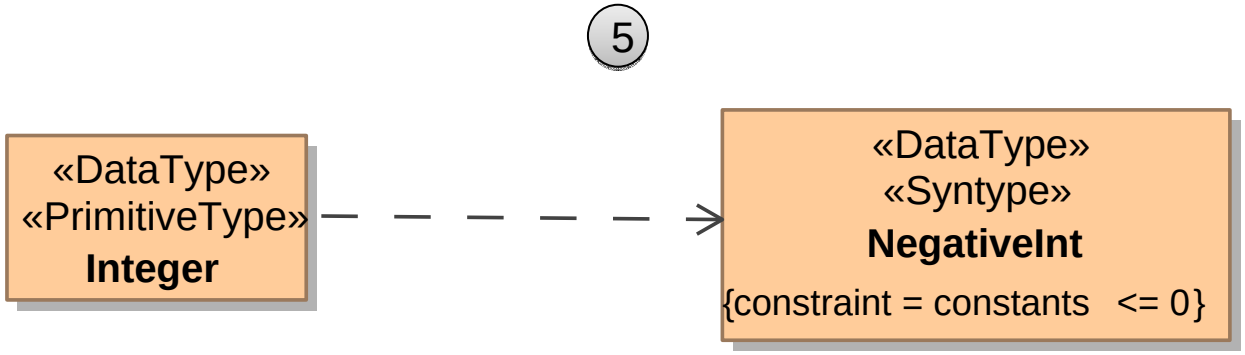





Fraunhofer ESK

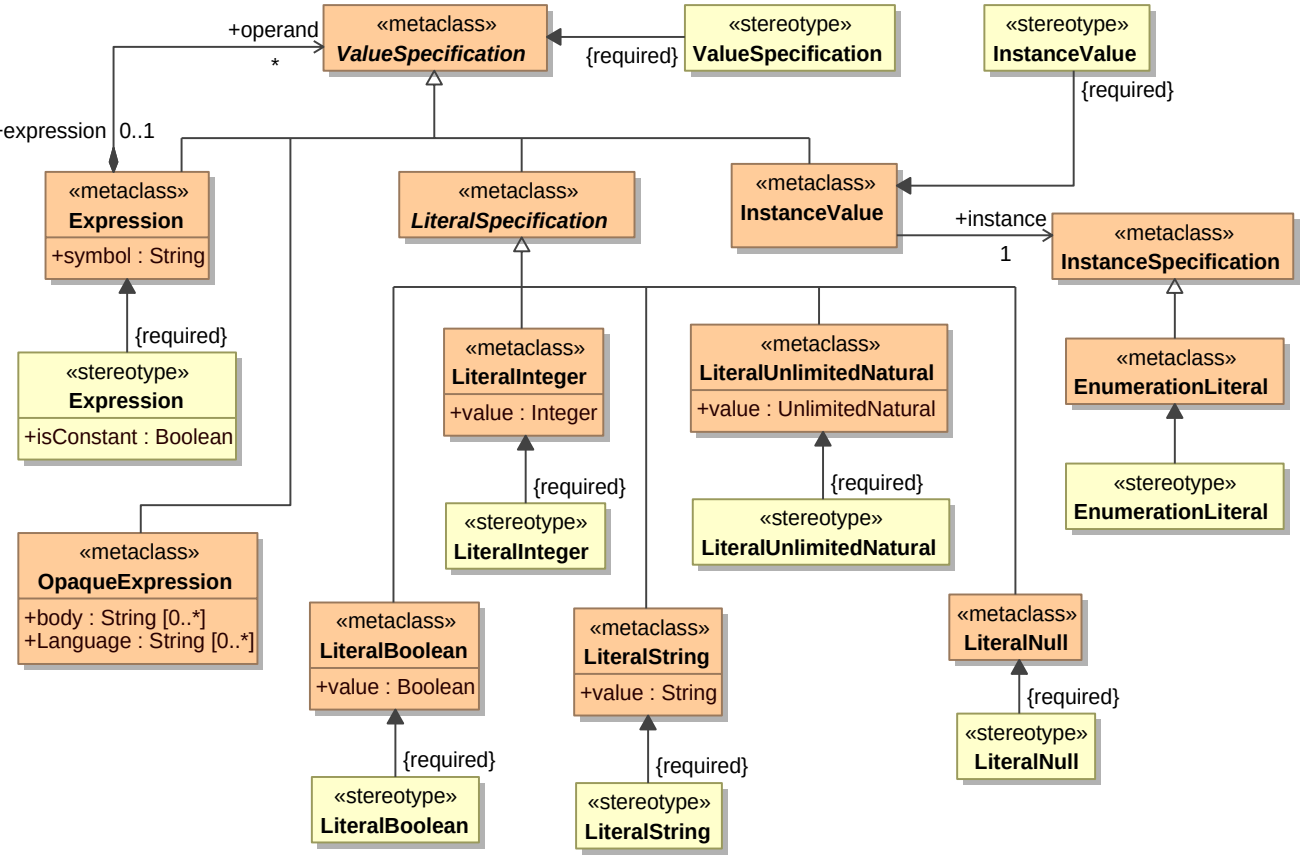
# 2. Data Type Model
## Application examples

5. <<Syntype>> DataType example:
   - The <<Syntype>> DataType can constraint an already existing data type.
   - For the purpose of specifying a <<Syntype>> DataType, a UML Dependency have to be used (example 5).
   - The <<Syntype>> DataType shall only consist of the tagged value constraint.



# 3. Value Specification Model
## Actual model

- SDL-UML supports 18 predefined data types
- Values of four pre-defined types are represented by LiteralSpecifications
- InstanceValue shall be used for other pre-defined types and user-defined types
- InstanceValue is also used for representing a SDL variable access






Fraunhofer ESK

# 3. Value Specification Model
# Limitations of the actual model

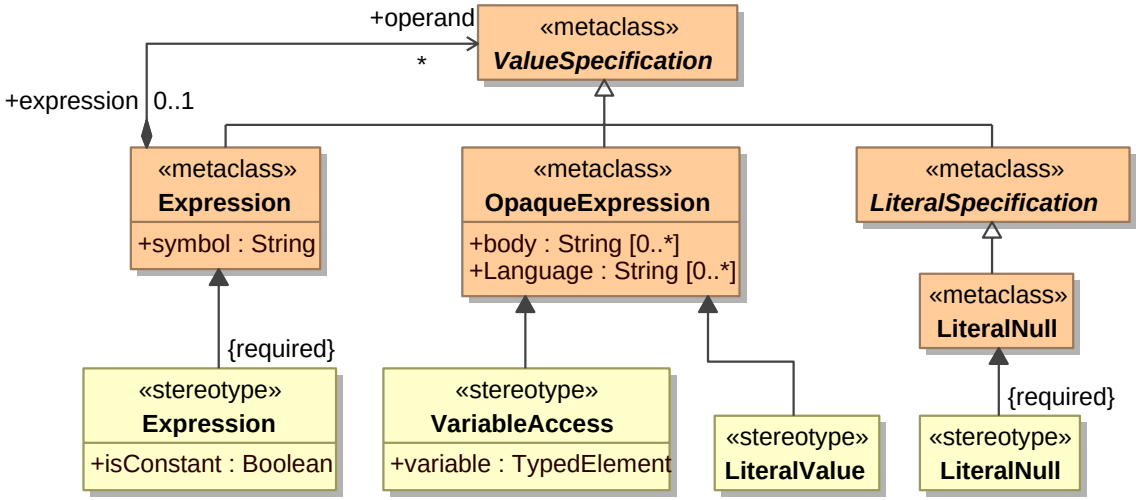
- The qualifiedName of <<InstanceValue>> InstanceValue can not be used for the mapping towards an SDL *Variable-access* or a *Literal*.

- All predefined data types of SDL-UML are specified in terms of a <<Primitive-Type>> PrimitiveType. But as instance property of an <<InstanceValue>> InstanceValue only an EnumerationLiteral or an InstanceSpecification can used.

- No constraints or rules are specified in which manner the mandatory instance property of an <<InstanceValue>> InstanceValue has to be handled.

# 3. Value Specification Model
## Proposed model

- <<LiteralValue>> OpaqueExpression shall be used for all value specifications.
- A SDL variable access is represented by <<VariableAccess>> OpaqueExpression.
- For <<Expression>> Expression and <<LiteralNull>> LiteralNull the semantics and constraints as specified by the Z.109 rec. shall apply.






Fraunhofer ESK

# 4. Conclusion & Future Work

- New models for data type as well as value specification for the upcoming version of the SDL-UML profile were presented.
- If it is considered to adopt both models, additional effort will be required in order to specify precise mapping rules and constraints.
- Constraints and mapping rules should not be specified only in a textual manner.
- For the specification of constraints, the Object Constraint Language (OCL) could be used additionally.