



AutomationML OCL Integration

Josef Prinz
INPRO



Short introduction to OCL
AutomationML OCL integration concept
AutomationML OCL Tools

- OCL (Object Constraint Language) is a formal language for the description of object constraints
- OCL is part of the UML (Unified Modeling Language)
- OCL is OMG standard and ISO standard ISO/IEC 19507
- OCL is “a formal language that remains easy to read and write”
- OCL is linked to an object model through UML embedding

■ Some terms and definitions

- Constraint: A constraint is a restriction on one or more values of an object-oriented model or system.
- Context: Object model (class), for which the constraint is defined.
- Instance: The object for which the constraint is calculated.
- Invariant: An invariant is a constraint that should be true for an object throughout its lifetime:

context <class name>

inv[<constraint name>]:<OCL expression>

Invariant example:

```
context Process inv: self.end > self.start
```

self always refers to the object for which the constraint is calculated.

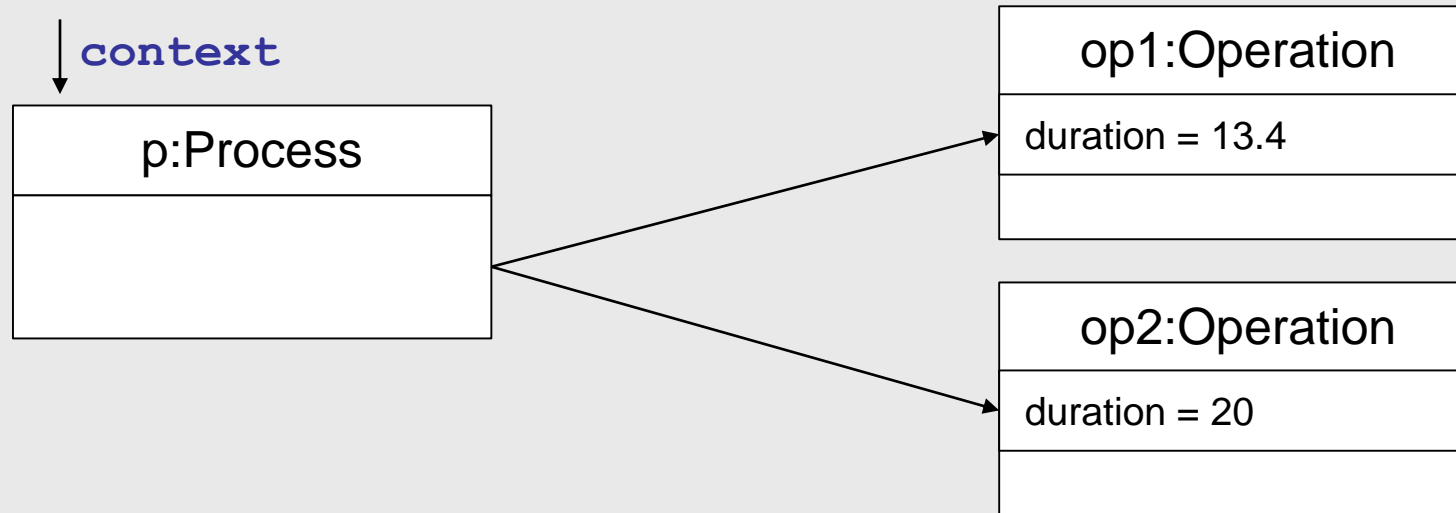
```
context Process inv: end > start
```

The keyword 'self' can be omitted.

```
context Process inv startEndConstraint: end > start
```

Constraints can be named.

OCL Navigation



`self.operation` -> {Operation op1, Operation op2}
`self.Operation.duration` -> {13.4, 20}

Some examples

- `context Process`
`inv: self.operation->forall (op|op.duration < self.duration)`
- `context Process`
`inv: if self.temperature > 20`
`then self.operation.pressure > 100`
`else self.operation.pressure <= 100`
`endif`

Iterator based Operations

- Ocl defines operations for collections using Iterators

Predefined Operations	Explanation
<i>select (expression)</i>	results in a collection, that contains all the elements from <i>collection</i> for which the expression evaluates to true.
<i>reject (expression)</i>	results in a collection, that does not contain the elements from <i>collection</i> for which the expression evaluates to true.
<i>collect (expression)</i>	results in a collection of elements that results from applying the expression to every member of the source collection
<i>forAll (expression)</i>	expression, which must hold for all objects in a collection
<i>exists(expression)</i>	checks whether a collection contains an element specified by expr
<i>isUnique(expression)</i>	returns true if the given expr evaluated on the body returns only different values.
<i>closure(expression)</i>	iterating over a transitive relationship (i. e. a tree structure)
<i>iterate(expression)</i>	iterating over all elements of a collection

OCL further reading

- OMG OCL language specification:
<https://www.omg.org/spec/OCL>
- OCL tutorial (by Jordi Cabot)
<https://www.slideshare.net/jcabot/ocl-tutorial>
<https://modeling-languages.com/wp-content/uploads/2012/03/OCLChapter.pdf>
- OCL tools
 - OCL Eclipse plug-in (OCLInEcore)
 - USE, a system for the specification and validation of information systems based on a subset of the Unified Modeling Language (UML) and the Object Constraint Language (OCL)
 - SimpleOCL is a proof-of-concept implementation of the OCL standard, built on top of the Eclipse Modeling Framework (EMF) and EMFText.



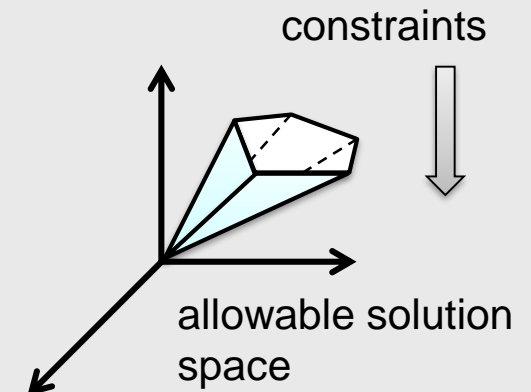
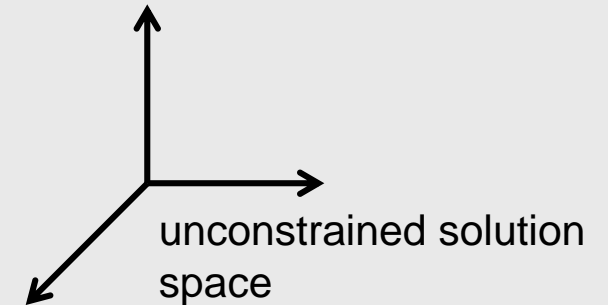
AutomationML OCL Integration

■ Motivation

- Feasible Designs (a feasible design is a design, where all constraints are satisfied)

■ Types of constraints

- Safety Constraints
- Quality Constraints
- Manufacturing Constraints
- Timing Constraints
- Component / Interface Compatibility
- Design Guides

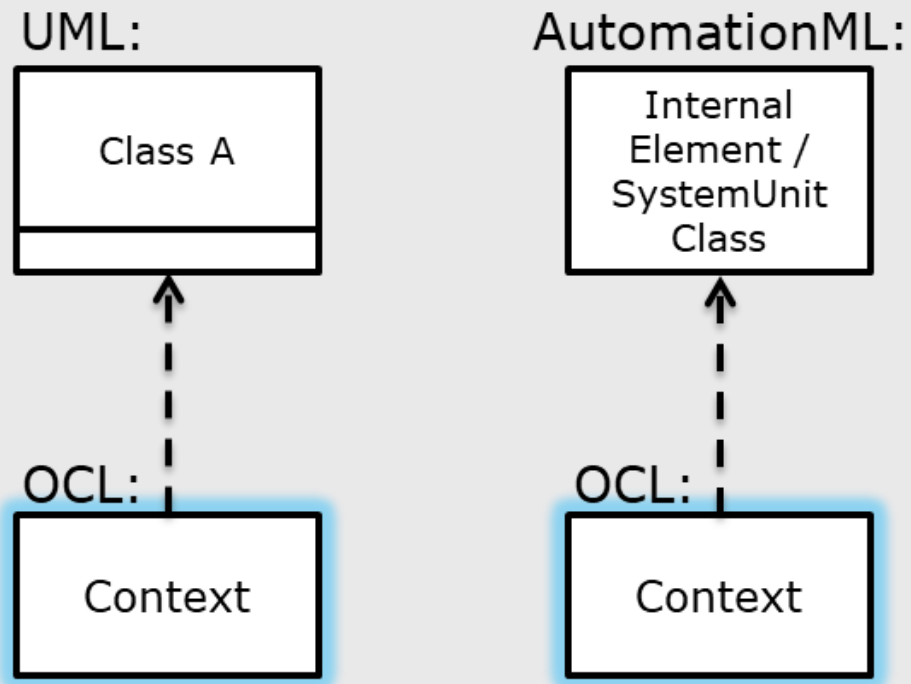


- **OCLE AutomationML object model embedding**
 - classes
 - SystemUnitClass, RoleClass, InterfaceClass
 - instances
 - InternalElement, ExternalInterface
 - relations
 - inheritance relations
 - class instance relations
 - instance instance relations (InternalLink)
 - parent child relations
 - attributes

■ OCL AutomationML object model embedding

Contextual Type	Contextual Instance	Properties	Associations	
			Related Class	Relationship
SystemUnitClass	InternalElement	Attribute InternalElement ExternalInterface <i>RoleRequirement</i>	SystemUnitClass RoleClass	Parent Child
			InterfaceClass	Instance to Instance
RoleClass	InternalElement	Attribute ExternalInterface	RoleClass	Parent Child
			InterfaceClass	Instance to Instance
InterfaceClass	ExternalInterface	Attribute ExternalInterface	InterfaceClass	Instance to Instance

■ OCL AutomationML navigation to associated objects

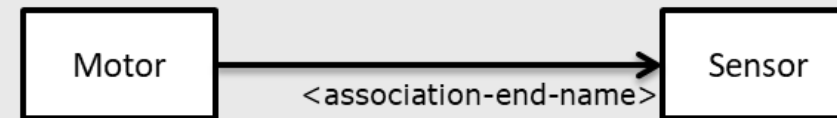


An OCL-context can be defined for:

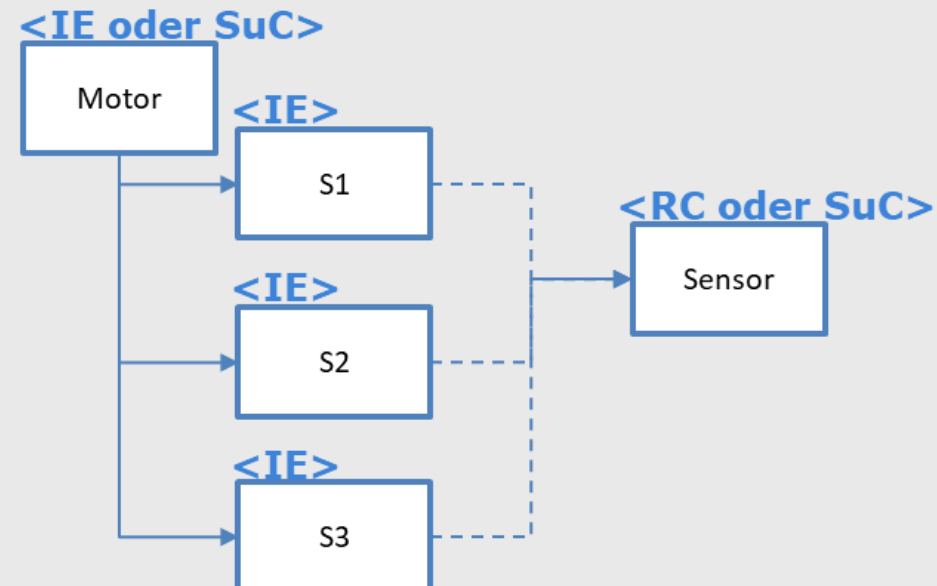
- SystemUnitClass
 - RoleClass
 - InterfaceClass
-
- InternalElement
 - ExternalInterface

- Associations in an AutomationML model are defined by the role classes and SystemUnit classes that are referenced by the child elements of an AutomationML object
- Example: An object motor (IE or SuC) that has IE children (parts) that reference either the RC "Sensor" or a SuC "Sensor" has an association: **Motor->Sensor**

UML:



AutomationML:



- If there is at least one robot there must also be at least one controller

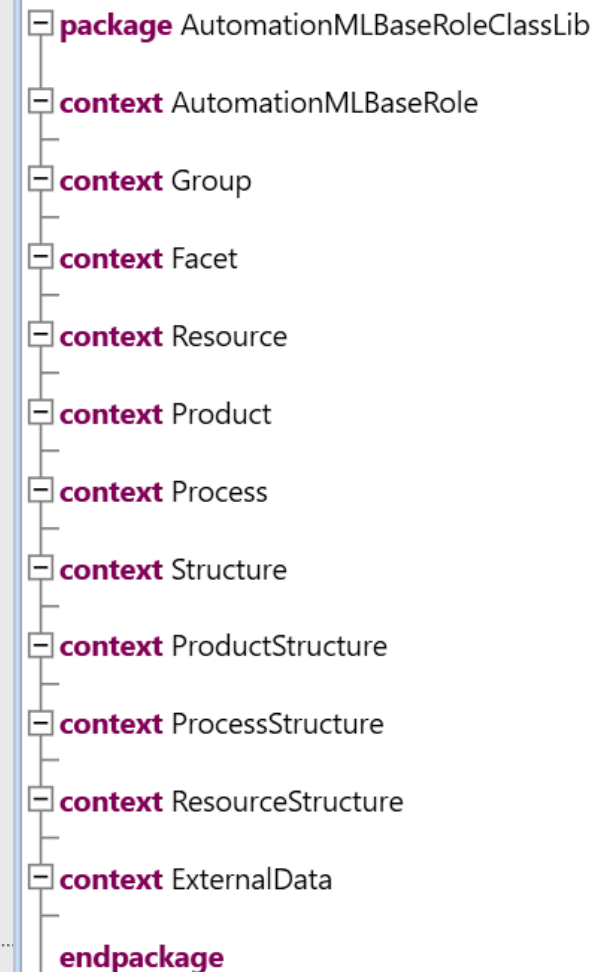
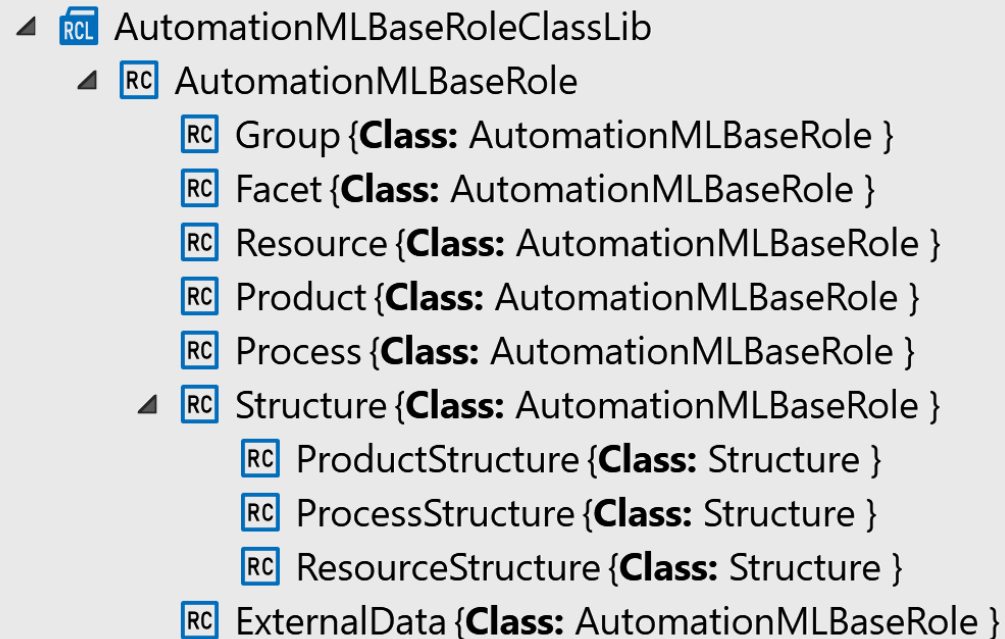
*„Robot-> notEmpty() **implies** Controler-> notEmpty()“*

- „->“ is a quantity operator
implies is a link operator for constraints
notEmpty() analog **size()>0**
- If there is at least one IE that references a SuC "Robot" or a RC "Robot", then there must also be at least one IE that references either a SuC or a RC "Controller". These can also be more specific classes.

- **OCL context and AutomationML class relationship**
 - Usage of class names as association end names
 - Class names are not unique across AutomationML libraries and need not be unique in single libraries
 - OCL provides a way of explicitly referring to types in other packages by using a package-pathname prefix.
 - Names of libraries have to be unique in AutomationML. Library names can be used as package-pathname.
 - Ambiguity of class names in a single library can be solved by using the CAEX Pathname as a package-pathname








■ OCL context and AutomationML class relationship

A package, created from the
AutomationMLBaseRoleClassLib



■ OCL context and AutomationML class relationship

A package, created from a user defined
SystemUnitClassLib

- ▲  Equipment
 - ▲  ABB
 -  RCS_Module
 - ▲  FANUC
 -  RCS_Module
 - ▲  KUKA
 -  RCS_Module

```

- package Equipment
- context ABB
- context FANUC
- context KUKA
- endpackage
  
```

```

- package Equipment::ABB
- context RCS_Module
- endpackage
  
```

```

- package Equipment::FANUC
- context RCS_Module
- endpackage
  
```

```

- package Equipment::KUKA
- context RCS_Module
- endpackage
  
```



AutomationML OCL Tools

- **Available Tools (for AutomationML members)**
 - OCL Editor and Interpreter
 - available as an AutomationML Editor PlugIn
 - OCL Engine
 - available as a .NET DLL



Thanks for your attention!

Josef Prinz : josef.prinz@inpro.de

inpro

Innovationsgesellschaft für fortgeschrittene
Produktionssysteme in der Fahrzeugindustrie mbH

Steinplatz 2

D-10623 Berlin

www.inpro.de