OCL Tooling for OMG specifications

Edward Willink

Thales representative for OMG OCL RTF
Eclipse OCL Project Lead

Thales representative for OMG QVT RTF
Eclipse QVT Declarative Project Lead

OMG Quarterly meeting
Santa Clara
13th December 2011

Overview

- Background
  - OMG OCL, Eclipse OCL, E.D.Willink
- OMG/Eclipse OCL Status/Progress
  - Old and New Eclipse OCL
- Eclipse OCL Tools
- UML.xmi syntactic/semantic errors
  - smells
- Summary

OCL Specification Background

- OCL 2.0 split off from UML 2.0
  - UML 2.0 not available for OCL alignment
  - UML FTF lacked resources to finalize
- OCL 2.0 'finalized' by QVT 1.0 team
  - no significant resolution of alignment
  - many incomplete areas
- OCL 2.2 minor changes
- OCL 2.3 minor improvements
- auto-generation for major improvements

Eclipse OCL Background

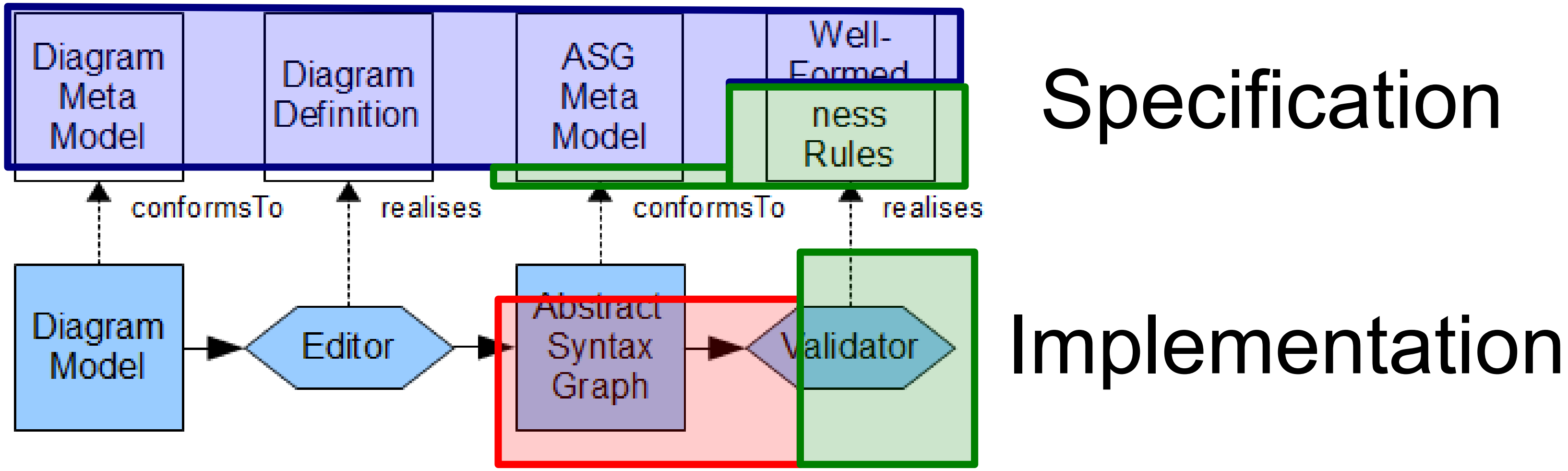
- OMG OCL 1.x within IBM
- OMG OCL 2.x within Eclipse
- Orginally Java API
  - OCL AST MM adjusted to extend Ecore MM
    - Ed Merks: "Never extend Ecore"
  - OCL MM made generic (Christian Damus)
    - OCL<.... Ecore .... >, OCL<.... UML ....>
    - long parameter lists - 10 types
    - wrong (base class) getXxx returns
  - Extensibility for QVTd (EDW) and QVTo (ASBH)

My Background

Embedded/DSP software synthesis

- UMLX - graphical model transformation
  - extension of QVT relations
    - based on OCL
    - make Eclipse OCL extensible
      - became Eclipse OCL (and QVTd) committer then leader
      - became Thales OCL (and QVT) RTF representative
    - modelize OCL
  - modelize QVT
- modelize UMLX

UML Specification/Implementation



[Diagrams: not this presentation, good new work]

Abstract Syntax Graph comprehensive

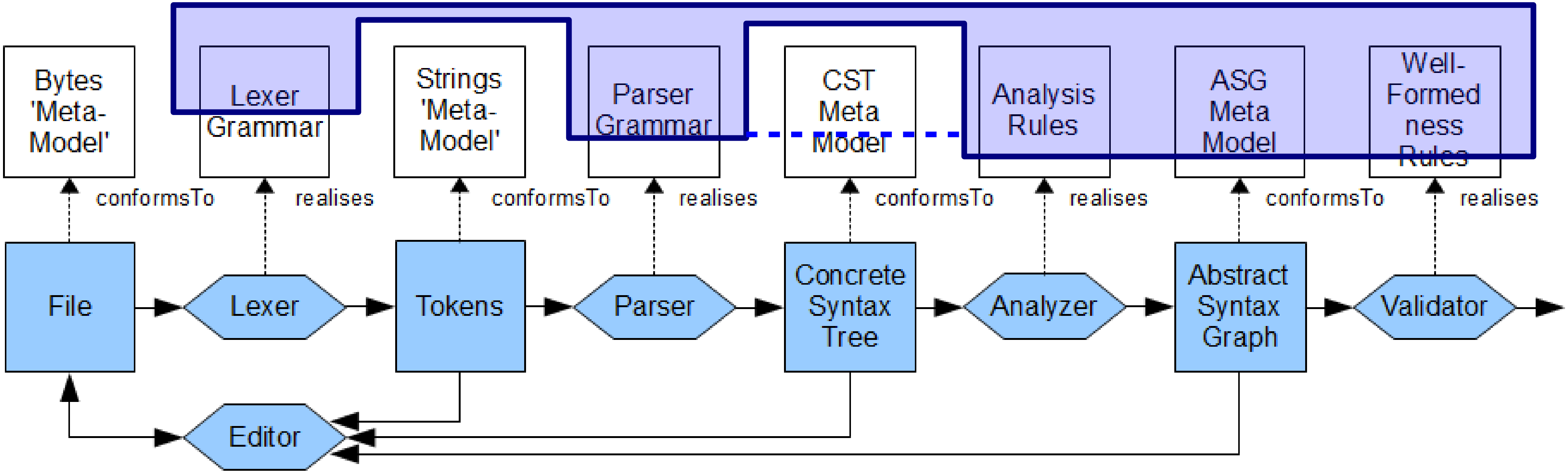
WFR 50% of 50% of 50% present correct accurate

Eclipse for UML 2.2 has significant manual input

- Eclipse for UML 2.4 is a bit more automated

Direct OCL to Java for fuller automation

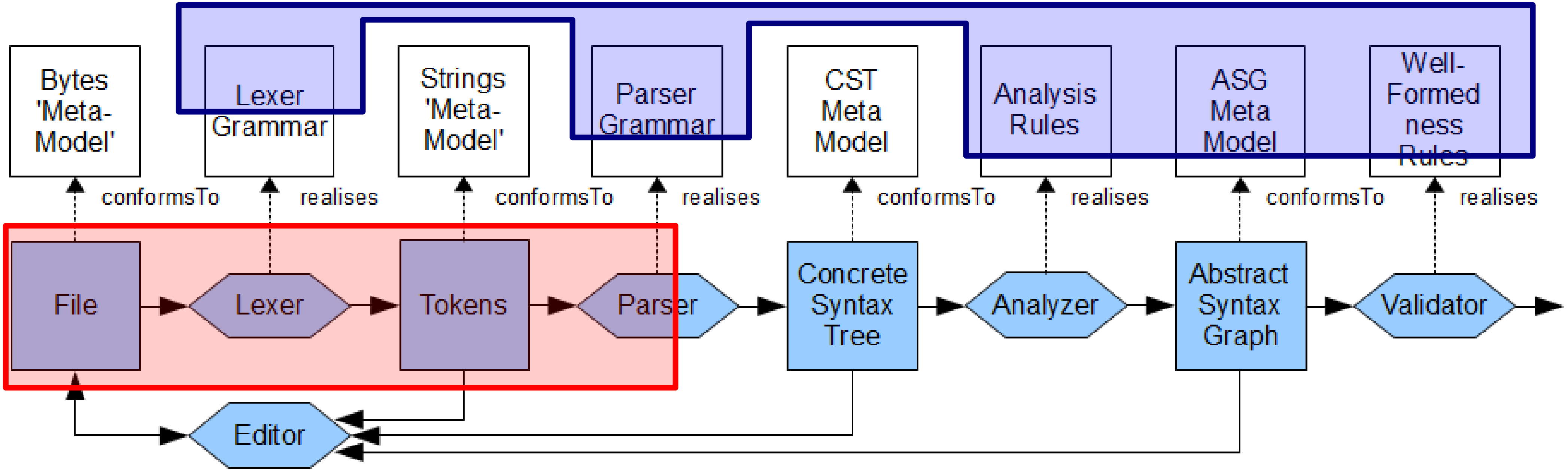
OCL Specification/Implementation



OCL specification is incomplete

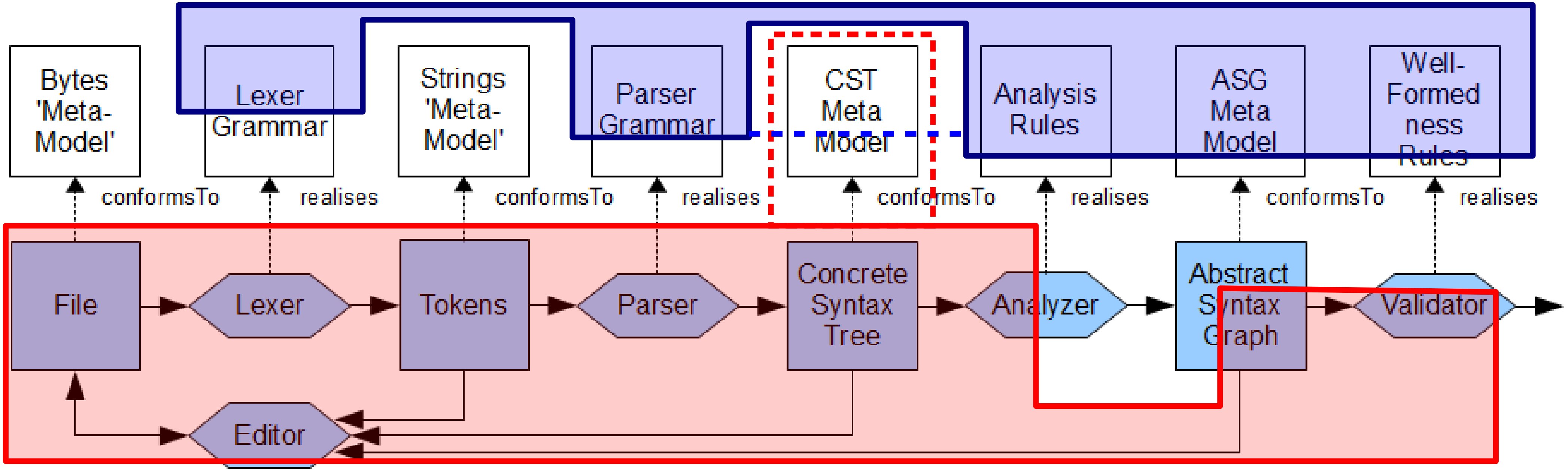
- EBNF grammars
- UML models
- OCL rules
- semi-OCL mappings - perhaps QVTr

The Old Eclipse OCL (RSA)



- Most of tool is manually coded
  - distinct Ecore, UML code, ~OCL 2.3
- Significant difficulties enhancing code
  - many bugs categorized as too difficult
- Significant difficulties exploiting OMG models

The New Eclipse OCL using Xtext

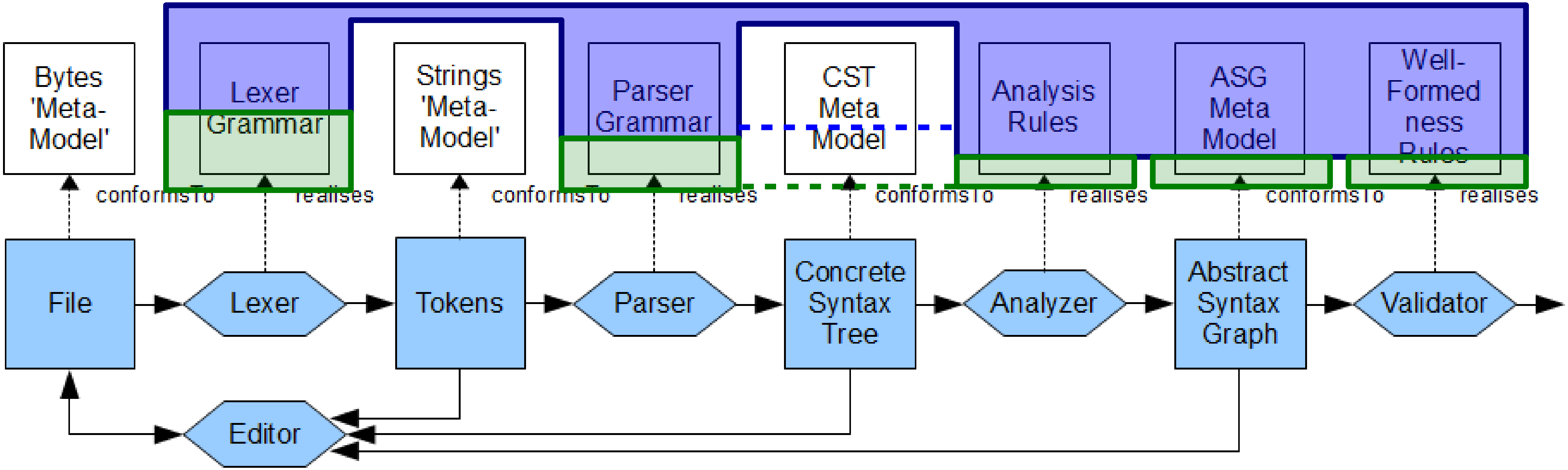


- Xtext covers large parts of an implementation
  - Xtext uses an EBNF with meta-model annotations

```
IfExpCS returns IfExpCS:
'if' condition=ExpCS
'then' thenExpression=ExpCS
'else' elseExpression=ExpCS
'endif';
```

- Modest gaps between specification and tooling

OCL Specification work



Define consumable EBNF grammars

Define consumable CST/ASG mapping

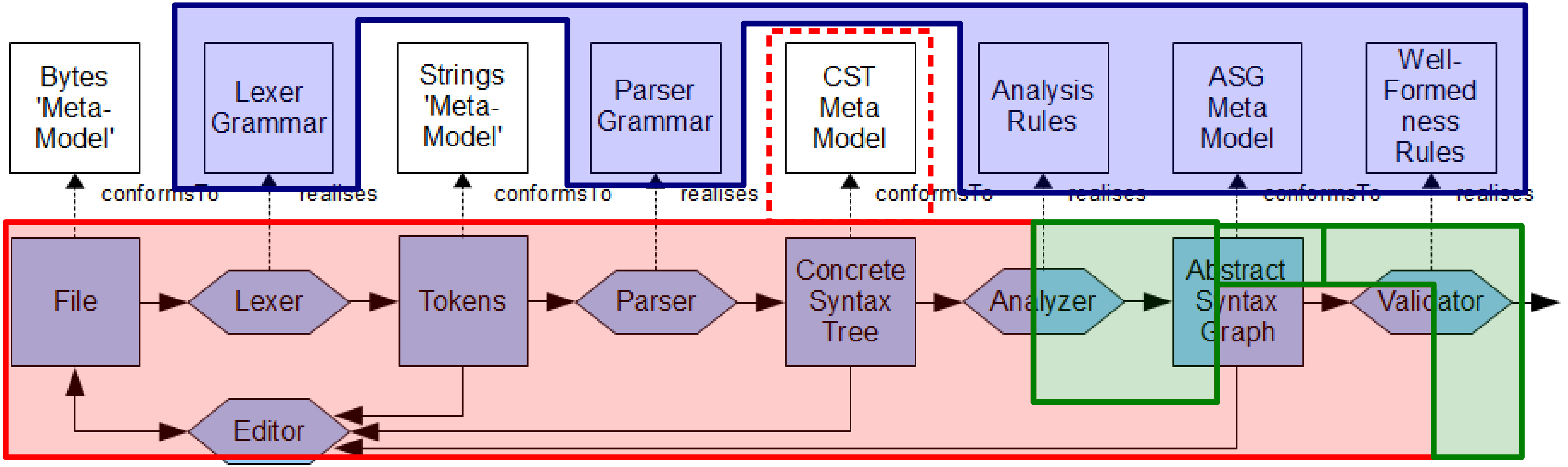
Define complete (UML+OCL) Pivot ASG model

Define accurate (UML+OCL) WFRs

Define extensible OCL Standard Library

Autogenerate specification from models

Eclipse OCL Tooling Work



OCL to Java automation of WFRs

- OCL-friendly Transformation Virtual Machine

Automation of Pivot ASG creation/mapping

Automation of CST to Pivot ASG mappings

Re-useable for QVT

The New Eclipse OCL

- OCL 2.4 prototype/candidate
- Shared specification/implementation models
  - UML element/type model
  - OCL expression/value model
  - OCL library model
- Auto-generated specification
- Re-useable model driven tooling
  - OCL, QVT
  - any EBNF/QVT/OCL defined language
- OCL Transformation Virtual Machine

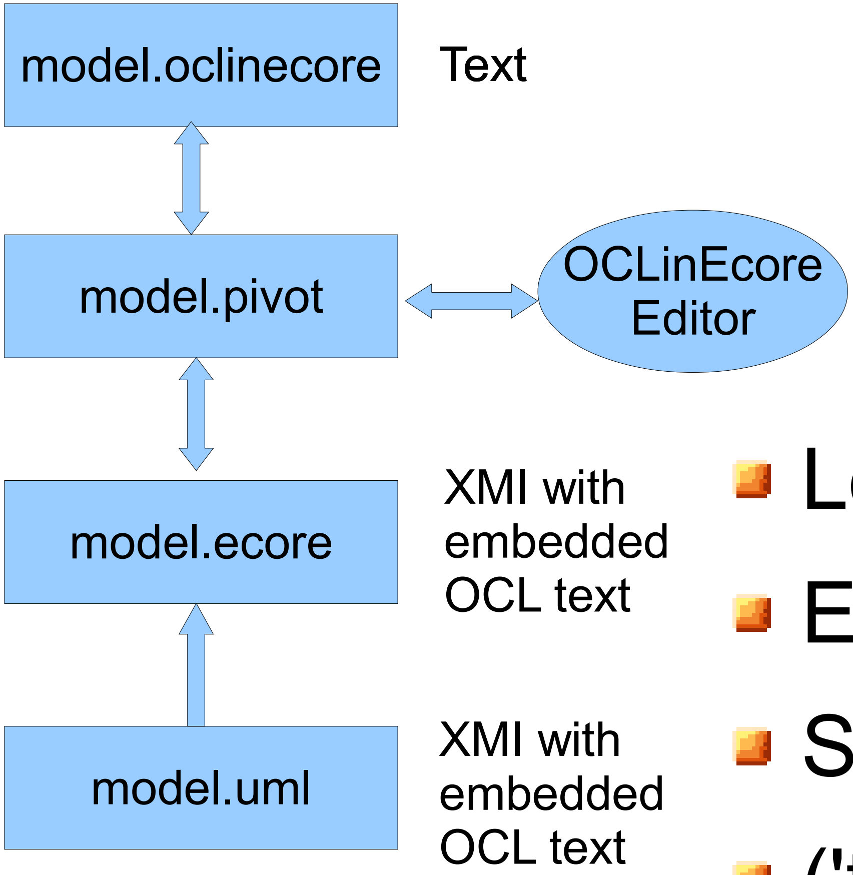
Old Eclipse OCL Tools: OCL ~2.3

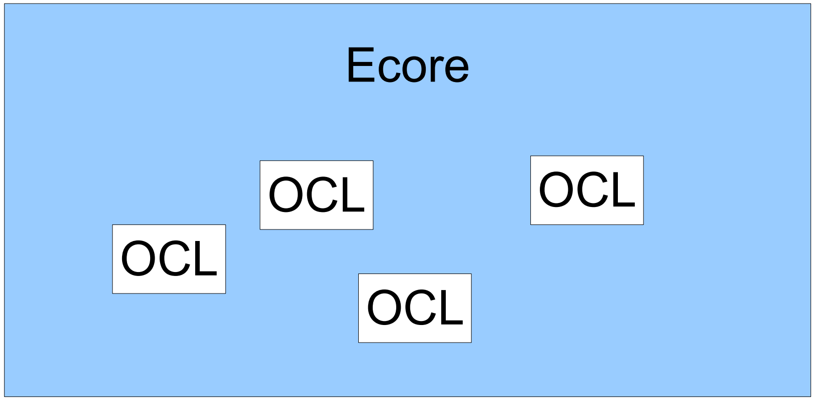
- Java API: parser, validator
- Java API: evaluator
- Model API: interpreted OCL execution
- UI: heuristic Essential OCL expression editor
- UI: interactive OCL console

New Eclipse OCL Tools : OCL 2.4...

  - Java API: parser, validator
  - Java API: evaluator
- Java API: library model and feature dispatch
- Model API: OCL 2 Java code generator
- Model API: Compiled OCL execution
- UI: Xtext Essential OCL expression editor
- UI: Xtext Complete OCL document editor
- UI: Xtext Composite OCL in Ecore model editor
- UI: Xtext OCL Standard Library editor
  - UI: interactive OCL console

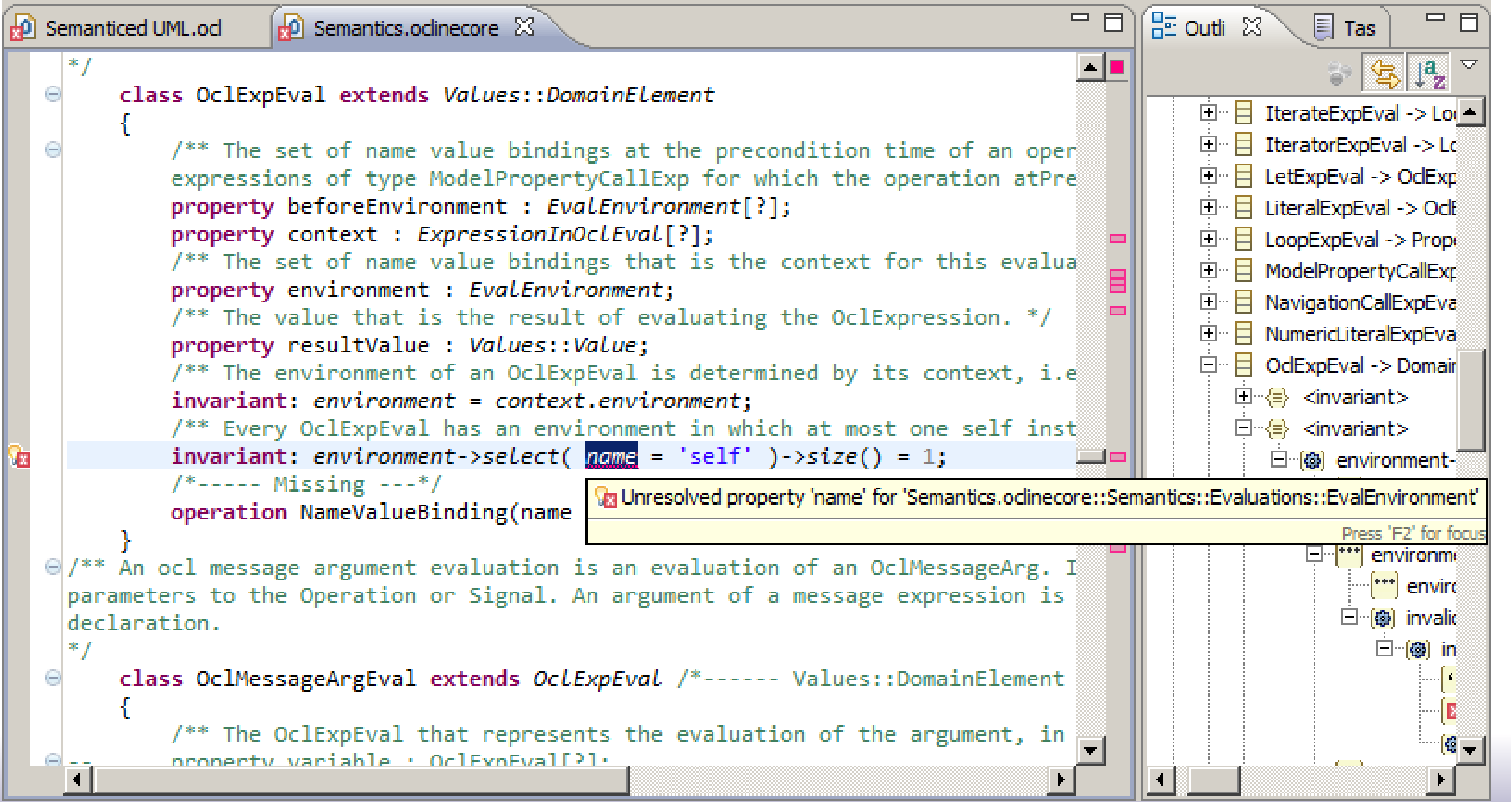
OCLinEcore Editor today





- Load UML, Ecore, OCLinEcore
- Edit Pivot as text
- Save as Ecore, OCLinEcore
- ('tomorrow': Save as UML)
- ('tomorrow': direct Pivot<>UML

OCLinEcore Example

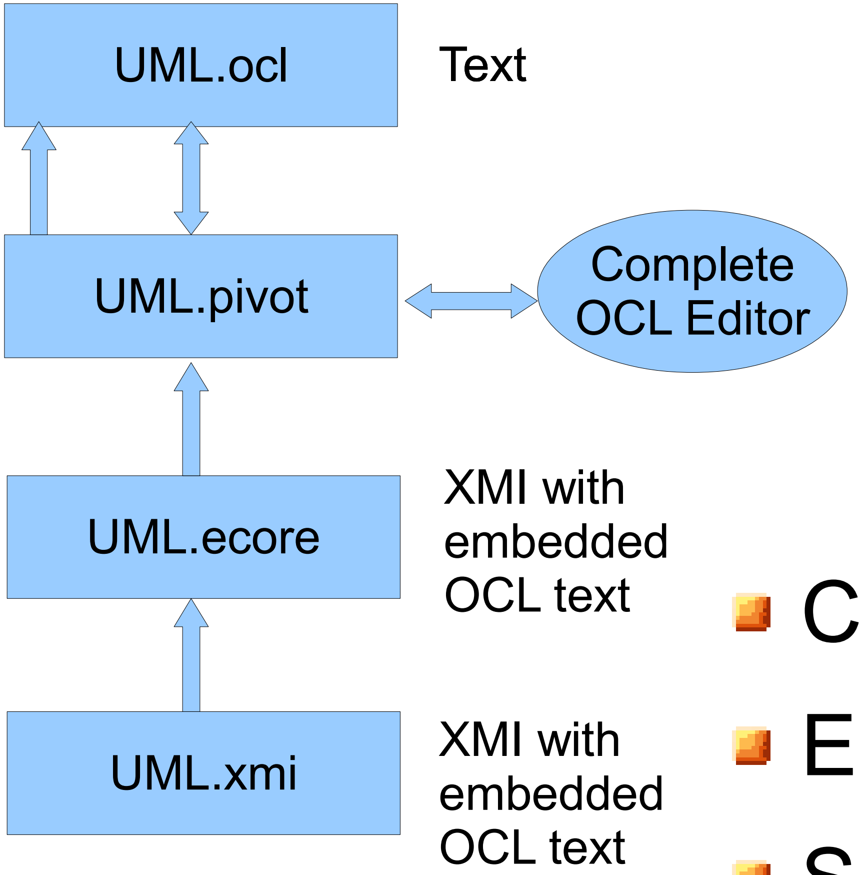


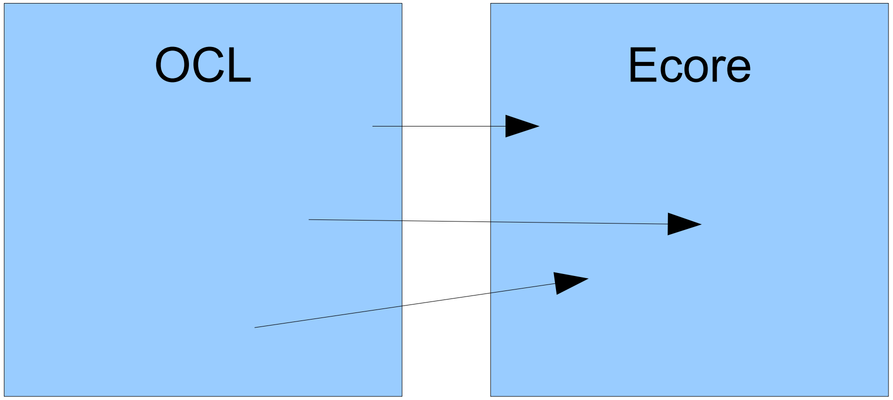
```
*/
    class OclExpEval extends Values::DomainElement
    {
        /** The set of name value bindings at the precondition time of an oper
        expressions of type ModelPropertyCallExp for which the operation atPre
        property beforeEnvironment : EvalEnvironment[?];
        property context : ExpressionInOclEval[?];
        /** The set of name value bindings that is the context for this evalua
        property environment : EvalEnvironment;
        /** The value that is the result of evaluating the OclExpression. */
        property resultValue : Values::Value;
        /** The environment of an OclExpEval is determined by its context, i.e
        invariant: environment = context.environment;
        /** Every OclExpEval has an environment in which at most one self inst
        invariant: environment->select( name = 'self' )->size() = 1;
        /*----- Missing ---*/
        operation NameValueBinding(name
    }
/** An ocl message argument evaluation is an evaluation of an OclMessageArg. I
parameters to the Operation or Signal. An argument of a message expression is
declaration.
*/
    class OclMessageArgEval extends OclExpEval /*------ Values::DomainElement
    {
        /** The OclExpEval that represents the evaluation of the argument, in
--      property variable : OclExpEval[?];
```

Unresolved property 'name' for 'Semantics.oclinecore::Semantics::Evaluations::EvalEnvironment'

Press 'F2' for focus

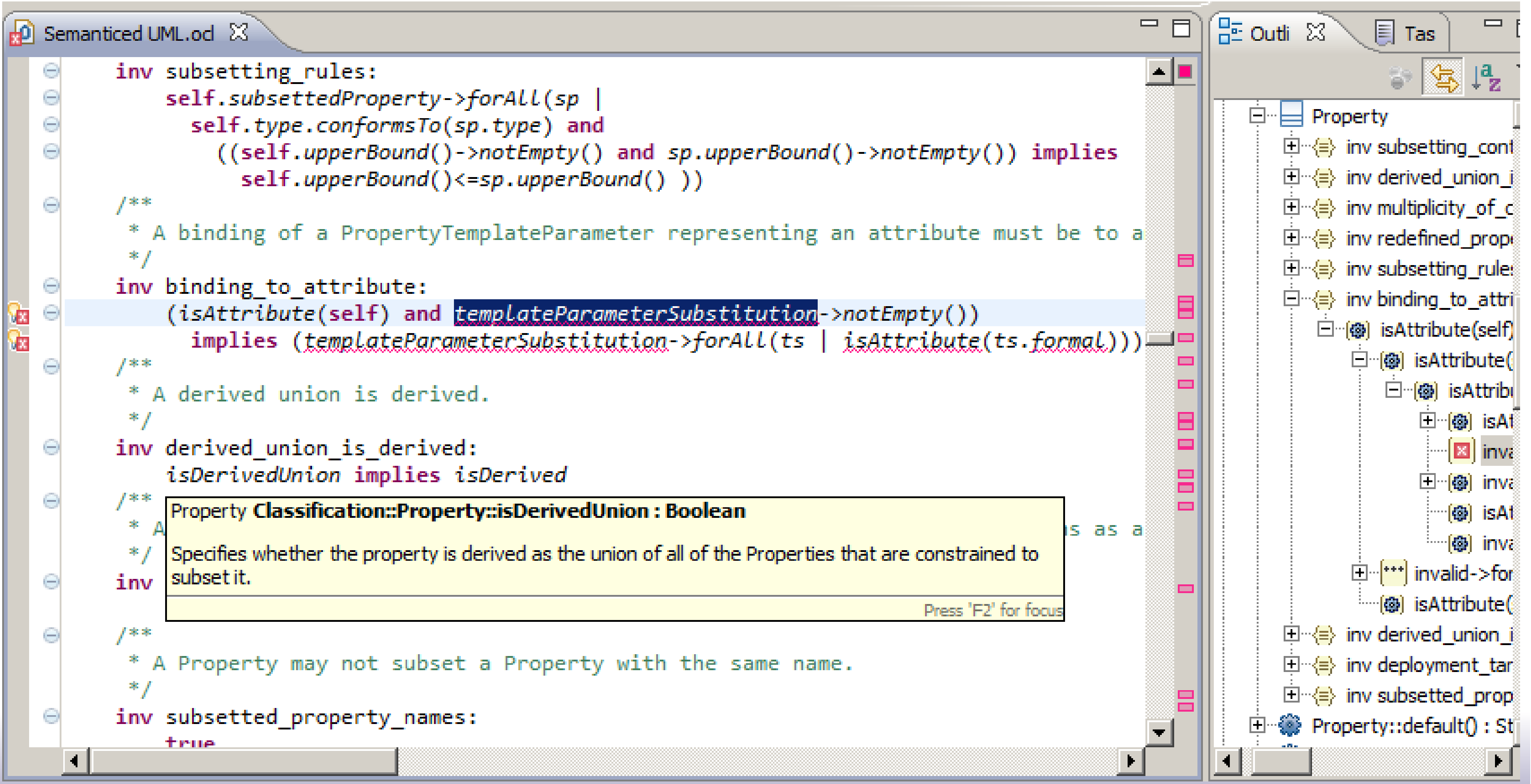
Complete OCL today





- Custom Complete OCL splitter
- Edit Pivot as text
- Save as Complete OCL text

Complete OCL Example



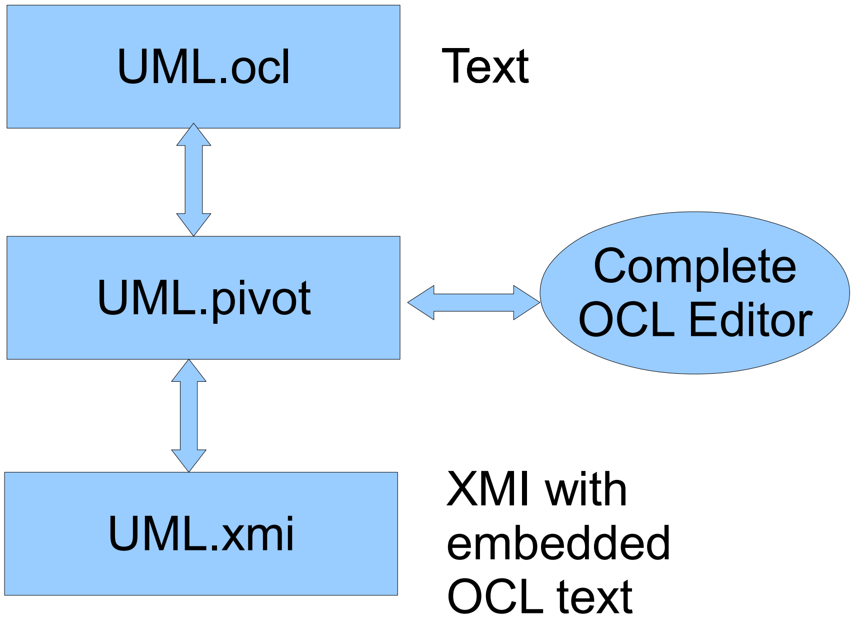
```
    inv subsetting_rules:
        self.subsettedProperty->forAll(sp |
          self.type.conformsTo(sp.type) and
            ((self.upperBound()->notEmpty() and sp.upperBound()->notEmpty()) implies
              self.upperBound()<=sp.upperBound() ))
    /**
     * A binding of a PropertyTemplateParameter representing an attribute must be to a
     */
    inv binding_to_attribute:
        (isAttribute(self) and templateParameterSubstitution->notEmpty())
          implies (templateParameterSubstitution->forAll(ts | isAttribute(ts.formal)))
    /**
     * A derived union is derived.
     */
    inv derived_union_is_derived:
        isDerivedUnion implies isDerived
    /**
     * A                                                                      s as a
     */
    inv

    /**
     * A Property may not subset a Property with the same name.
     */
    inv subsetted_property_names:
        true
```

Property **Classification::Property::isDerivedUnion : Boolean**

Specifies whether the property is derived as the union of all of the Properties that are constrained to subset it.

Press 'F2' for focus

Complete OCL Editor 'tomorrow'



UML.ocl

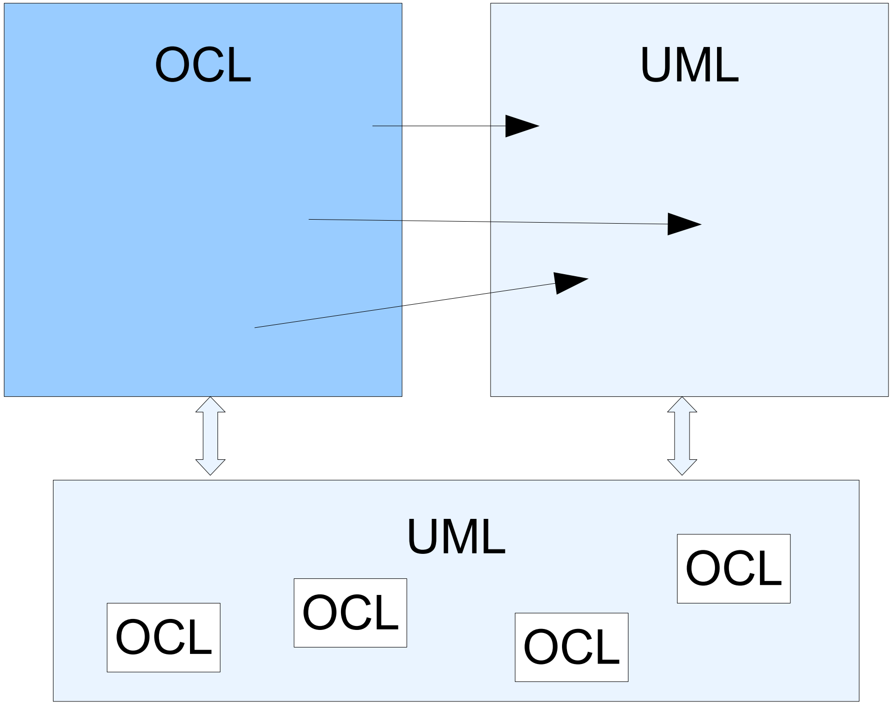
Text

UML.pivot

Complete OCL Editor

UML.xmi

XMI with embedded OCL text



OCL

UML

UML

OCL

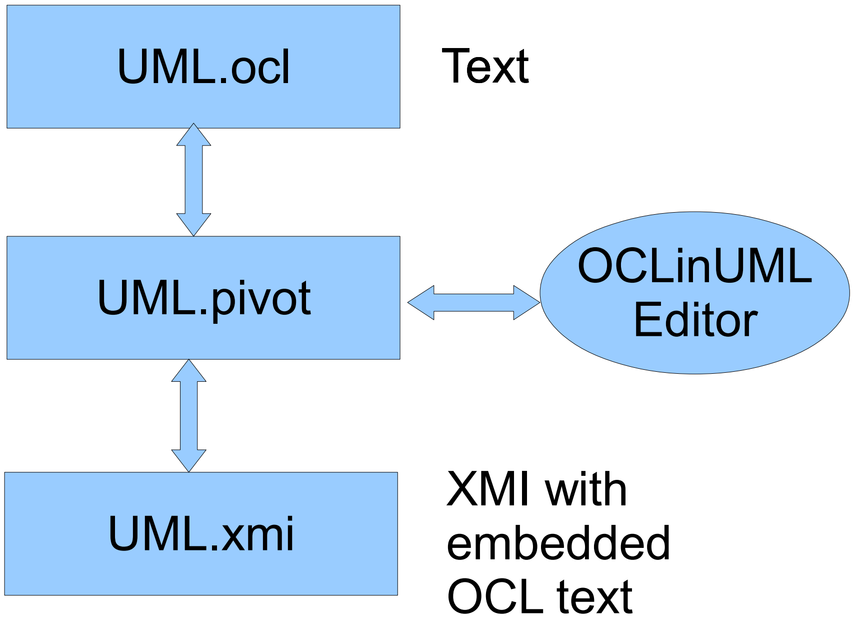
OCL

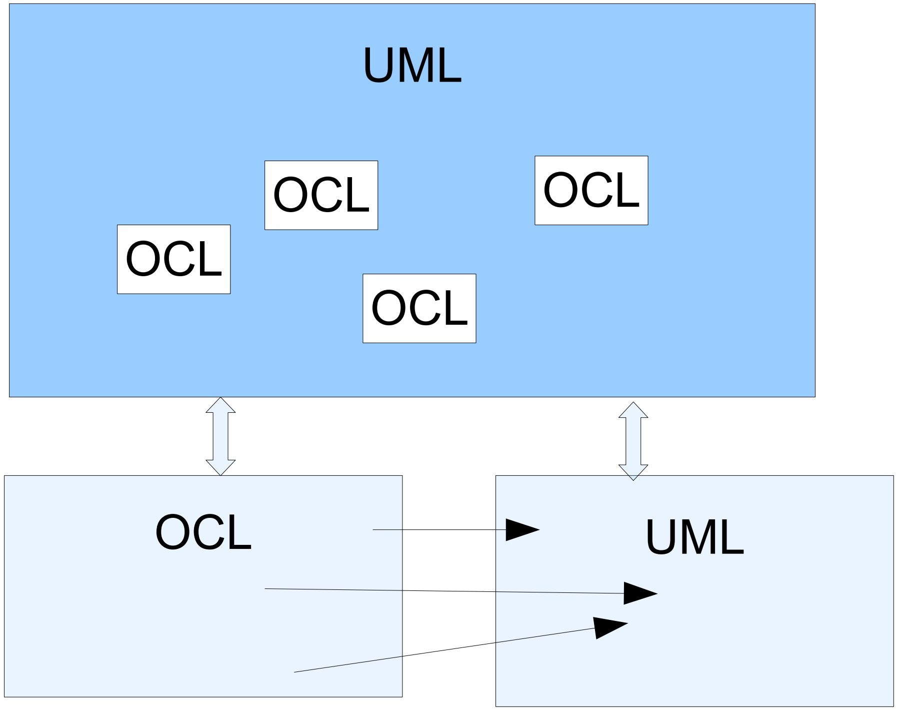
OCL

OCL

- Load separate/merged UML/OCL
- Edit as OCL text
- Save as separate/merged UML/OCL

OCL in UML editor 'tomorrow'





- Load separate/merged UML/OCL
- Edit as UML and OCL text
- Save as separate/merged UML/OCL

UML <2.5 Syntax Errors

Problem	Wrong Text	Correct Text	Diagnosis
Bad built-in literal	#true	true	Error: [Never valid]
OCL 1.6 enum literal	#out	ParameterDirectionKind::out	Error: [Very obsolete]

Local def	inv xxx:
	  def: defBody
	  invBody
		def: defBody
		inv xxx:
		   invBody
			Error:

Missing parenthesis			Error: 'endpackage' expected
Extra parenthesis			Error: red squiggle under extra

Reserved names	forAll(in | in)	forAll(_in | _in)
		forAll(_'in' | _'in')
			OCL 2.0, 2.2
			OCL >=2.3

Hyphenation	a- >b	a->b	
Missing endif	if a then b else c	if a then b else c endif	
Missing else endif	if a then b	a implies b	
Complex endif			

UML <2.5 Syntax Smells

Problem	Wrong Text	Correct Text	Diagnosis
Missing body	inv x:	inv x: TBD	
Multi-invariant	inv multi_invariant: complex1 and complex2	inv multi_invariant1: complex1 inv multi_invariant2: complex2	Ok: confuses errors in complex

- 74 easy to fix syntax errors

UML <2.5 Semantic Errors

Problem	Wrong Text	Correct Text	Diagnosis
~~Name Qualification~~	~~Class~~	~~structuredClassifiers::Class~~	~~Error: unresolved type~~
Typos	isOclKindOf isComoposite associations	oclIsKindOf isComposite association	Error: unresolved operation
Result	body: result = yyy	body: yyy	Error:
Bad any	a->any()	a->any(true)	
Bad - on non-Set	a - b	a->asSet() - b->asSet()	
Bad String Concat	'pfx'->union(separator())	'pfx' + separator() 'pfx'.concat(separator())	OCL >= 2.2 OCL 2.0
Bad Types	Aggregation NavigableEnd	AggregationKind ????	
Bad Code			

UML <2.5 OCL 2.3 Semantic Errors

Problem	Wrong Text	Correct Text	Diagnosis
~~Collections~~	~~Set(Class)~~	~~Set(structuredClassifiers::Class)~~	~~Error: unresolved type~~
Inferred Opposites	p.informationFlow	p.InformationFlow	Make explicit
Result	body: result = yyy	body: yyy	Error:
Missing parentheses	oclType	oclType()	
Extra parentheses	specification()	specification	
Navigation operator	a.select(...) b->c	a->select(...) b.c	
Complex parentheses			
Bad navigation	self.argument.multiplicity.is(1,1)	self.input.is(1,1)	

OCL Navigation Operators

- Collections reify UML Multiplicities
- Collections are not UML class instances

	Object	Collection
.	Navigation	?
->	?	Navigation

`anObject. ...`	object navigation
`aCollection-> ...`	collection navigation

Shorthands

`aCollection....`	`aCollection->collect(...)`
`anObject->...`	`anObject.oclAsSet()-> ...`

UML <2.5 Complete OCL

Problem	Wrong Text	Correct Text	Diagnosis
~~Imports~~		~~structuredClassifiers::Class~~	~~Error: unresolved type~~
Primitive Types	types::Integer	Integer	
Underspecified types	Set{}->includes(a)	Set{a}	OCL 2.3 defines an upper/lower bound
Closure	... union ... recursive call	aClass->closure(superClass)	OCL 2.3 in RAS 8

UML <2.5 Semantic Smells

Problem	Suspect Text	Correct Text	Diagnosis
Redundant iterator	a->forAll(b | b.c)	a->forAll(c)	
Double implication	(a implies b) and ((not a) implies (not b))	a = b	If Boolean
Redundant self	self.input	input	Useful in iteration bodies

- Over 250 semantic error regions

UML <2.5 Well Formedness Errors

Problem	Wrong Text	Correct Text	Diagnosis
Set{} as null	Set{}	null	OCL 1.x practice

- Expect many hundreds
- Expect minor easy fixes
  - e.g. give each invariant a distinguishable name

Smells

- Upper case navigation names
  - low confidence in Association tooling
- oclIsTypeOf (exact type comparison)
  - oclIsKindOf (equivalent type comparison)
- oclAsType (type cast)
  - may be redundant
  - may need an oclIsKindOf predicate
  - perhaps the meta-model is at fault

Summary

- OCL and UML specifications inadequate
  - most problems soluble with tooled models
  - tooling becoming useful
- Code generation from OCL for WFR tooling
  - today's activity for Eclipse OCL
- UML OCL can be tooled
  - RSA (old Eclipse OCL) viable
  - new tooling should be complementary