



# Modeling/MDT/OCL Scope Restructuring Proposal

Edward Willink

# History / Old Scope



## EMFT/OCL moved to MDT/UML2/OCL

(under <http://www.eclipse.org/proposals/mdt>)

- "It will be covered by the UML2 newsgroup (eclipse.modeling.mdt.uml2) and it will be listed as a component (e.g. UML2 OCL) under a Modeling/MDT product in Bugzilla."
- "OCL is an implementation of the OCL OMG standard for EMF-based models."

## EMFT/OCL actually moved to MDT/OCL

- As a top level component MDT/OCL has its own newsgroup and Bugzilla
- Clearly in-scope
  - MDT/OCL provides OCL OMG standard parsing to Ecore/UML2.
  - MDT/OCL provides OCL OMG standard evaluation to Ecore/UML2.
- Perhaps beyond scope
  - example interactive evaluation environment (for many years)
  - example editors (since Helios)

# Rationale



## OCLE users need more

- editors (with full model-driven checking)
  - enriched Ecore/UML meta-models
  - complete OCL documents
  - re-use within extended tools
- efficient code generation (for OCL embedded in Ecore/UML models)

## Intervening events

- MDT/OCL Tools proposal with similar goals failed to be created
- Ecore enables use of embedded OCL in generated and reflective models
- QVTo, Acceleo make efficient code for OCL validation feasible
- Xtext merits revisiting the core parsing
- OCL-based languages require library extensibility

# New Scope



## Existing 'Core' components

- OCL OMG standard parsing for Ecore/UML2
- OCL OMG standard evaluation for Ecore/UML2

## New 'Core' components

- Extensible OCL standard Library

## New 'Tools' Components

- Extensible Interactive Evaluation for OCL expressions on Ecore/UML models
- Extensible Xtext-based editors for
  - OCL Standard Library
  - OCL embedded in Ecore or UML
  - OCL Documents
  - OCL Expressions
- Extensible Java code generation for OCL expressions

# One or Two Projects



Separate Core and Tools projects logical

But

- no new committers (more welcome)
- cyclic development dependencies
  - Standard Library redefined by Standard Library Editor
  - Validation defined by Embedded Editor
  - Validation uses Code generation

Therefore one project

- reduces administrative distractions
- can be reviewed/restructured later

# Impact



## The MDT/OCL component is unchanged

- Core features will continue to deliver core functionality (at Platform +1)
  - programmers and reusers will see just normal evolution
    - e.g. Acceleo, QVTo, Modisco
- Tool features will provide an enhanced environment (at Platform +3)
  - users and interactive consumers will see a much richer tool kit
    - e.g. Papyrus