Eclipse OCL for Juno

Edward Willink

Eclipse OCL Project Lead
OMG OCL RTF chair

EclipseCon 2012 Modeling Symposium
Reston, VA
26th March 2012
Overview

- Background
  - why OCL

- Embedded OCL
  - OCLinEcore editor
  - OCL 2 Java code generation

- Complementary OCL
  - Complete OCL editor
  - Load Complete OCL Resource

- Summary
Basic Modeling

Class Diagrams / Entity Relationship Diagrams

- structural modeling
  - packages, classes, features/properties, operation names

Easy with EMF / Ecore Diagram / ....
Enriched Modeling in EMF

Richer modeling

- derived features
  - constants only in EMF
- operation bodies
  - signature only in EMF
- class constraints / invariants
  - name only in EMF annotations

manual '@generated not' Java

/**
 * <!-- begin-user-doc -->
 * <!-- end-user-doc -->
 * @generated
 */

@Override
public <R, C> R accept(Visitor<R, C> visitor) {
    return visitor.visitClass(this);
}
Enriched Modeling beyond EMF

- Manual Java code
- OCL in Ecore XML CDATA
- OCL in Ecore EAnnotations
- OCL in Ecore as EAnnotation delegates [Helios]
  - compiled and interpreted at run-time
- OCL in Ecore as EAnnotation delegates [Juno]
  - compiled to Java by genmodel
  - executed by OCL VM at run-time
- Java/Xbase in Xcore [Juno]
Xcore and friends / OCL

- If all you want is Java
  - use Xcore/Xbase not OCL

- OCL is a Specification language -
  - platform neutral

- Sharper syntax
  - much of it adopted by Xbase

- Declarative
  - side effect free, no assignment

- Foundation for MOFM2T(Acceleo), QVT, ...
OCL in Ecore

- Available since Helios
- Very like Xcore
- Tutorial.ecore Outline/Hovertext/loans->select
  - Member.loans
- Tutorial.xmi Validate/Property m3
- Eval m3 loans, loans.book
- Tutorial.genmodel, Global OCL preference
Complete OCL

- Available since Helios
- Loadable since Juno M6
  - Semanticed UML.ocl @ non_final_parents
    - syntax complete on is|Final ... coments
  - xtext.ocl, Arithmetics.xtext Validate
    - add xtext.ocl, Validate, INTa, AbstractDefinition9, no ecore
  - Tutorial.ecore as Ecore, select package,
    - load Tutorial.ocl, Validate
Summary

Eclipse OCL
- originally targetted at Java API

OCL in Ecore editor
- now compilable to Java

Complete OCL editor
- now loadable into Xtext or Ecore or ... editors/...
Less is more

- C compared to assembler
  - no stack, no condition codes => no stack/flow bugs
- C++ compared to C
  - no malloc... => no memory anomalies
- Java compared to C++
  - no delete, Object => 'no' memory leaks, polymorphic
- OCL compared to Java
  - no assignment => no side effects
  - => OCL is analyzable, fast incremental evaluation
Pre-Demo

- Close OCL Xtext Console
- Delete Tutorial src, MANIFEST, plugin.xml
- 1 Open Tutorial.ecore with OCLinEcore
- 2 Open Tutorial.xmi
- 3 Open Tutorial.genmodel
- Open OCL Xtext Console
- 4 Open Semanticed UML.ocl
- 5 Open xtext.ocl
- 6 Open Arithmetics.xtext