Model Driven Engineering, VT2015

Matthias Tichy, matthias.tichy@cse.gu.se
Grischa Liebel, grischa@chalmers.se
Fredrik Einarsson, freein@student.chalmers.se

Additional remarks for all assignment deliverables

- *State the authors of the deliverable (the group members)*
- *Correct language use, no grammar or spelling errors*
- *Reference and describe all figures/tables in the text*
- *Figures and graphs should be readable from a quality perspective*
- *Reference literature in your text where appropriate*
- *Define non-obvious acronyms*
- *The deliverable should be easily readable, understandable and complete*
- *Give arguments for your decisions (also using references)*
- *Show critical thinking*
- *Be prepared to get frustrated if something does not work as you think it should. Rise to the challenge!*

Note

There are several different ways to specify OCL constraints in Ecore. Here, we will use both the (interactive) OCL console, where you can directly execute expressions on single model elements in a console-like fashion, and the OCLinEcore editor, which allows you to place constraints in the Meta Model in a Java-like fashion. In both cases, the context is implicitly given. For the OCL console, the context is dependent on which model element you currently have selected. For the OCLinEcore editor, the context depends on where you place the constraint in the code.
However, when listing your invariants in the report (part B), we expect you to provide the context as well!

Assignment 3 – Object Constraint Language

Hard deadline for Handin via PingPong: 11.2. 23:59 (CET)

The meta model of the manufacturing system specification language which you specified in assignment 2 (probably) allows models which would result in incorrect specifications of manufacturing systems.

For example, the solution meta model (you can download the whole project *assignment2.zip* from PingPong) allows two manufacturing steps to be connected, without an intermediate transport step.

In order to restrict the meta model to allow only valid models, you will extend the meta model using OCL.

A. Execute OCL constraints

Download the meta model solution for the previous assignment in PingPong. You will use this instead of your own meta model solution. Execute the following OCL expressions on the provided example model (example.xmi; an example illustration of the manufacturing system, with some omitted details, is depicted in Figure 1), first manually and then using the interactive OCL console:

1. **Expression:** `self.consistsOf.transition`
   Context: `ManufacturingSystem FurnitureSystem`
2. **Expression:** `self.consistsOf ->forAll(m:ManufacturingSystemElement| m.name <> null)`
   Context: `ManufacturingSystem FurnitureSystem`
3. **Expression:** `self.consistsOf ->select(oclIsKindOf(Step)).oclAsType(Step).speed->sum()`
   Context: `ManufacturingSystem FurnitureSystem`
4. **Expression:** `self.output.input->forAll(i:InputCondition|self.input ->includes(i))`
   Context: `TransportStep EndTransport`
5. **Expression:** `self.transition->exists(m:ManufacturingSystemElement| m.oclIsTypeOf(QualityAssuranceStep)) implies self.oclIsKindOf(ManufactoringStep)`
   Context: `CompositeStep CuttingComposite`
6. **Expression:** `ManufacturingSystem.allInstances() ->forAll(ms:ManufacturingSystem|ms.transforms.hasType ->includes(self) implies ms.uses->includes(self))`
   Context: `WorkPieceType CutMetal`

For each expression, explain in one paragraph what the expression does.

B. Extend the Meta Model

Use OCL to define additional constraints as static semantics for each of the following requirements:

1. Each Step shall have exactly one Responsible.
2. Each Responsible shall be responsible for between one and three (inclusive) Steps.
3. There shall not exist a single Step in a ManufacturingSystem which has the same Responsible as the ManufacturingSystem itself.
4. No start StoragePoint shall have incoming Transitions.
5. No end StoragePoint shall have outgoing Transitions.

6. Transitions shall only connect ManufacturingSystemElements within the same ManufacturingSystem. E.g. in the example in Figure 1, it would not be allowed to connect RawMaterialStorage directly to the Step Cutting, as Cutting is inside another ManufacturingSystem (CuttingComposite).
7. Outputs and inputs of connected Steps shall be compatible. That is, the input condition of a Step has to include at least one condition on a WorkPieceType that is produced by at least one Step connected to it via an incoming transition. E.g. in the example in Figure 1, the output of Step Cutting is compatible with Step Drilling, as at least one of the possible outputs of Cutting (CutWood and CutMetal) occur in the input of Drilling.
8. For composite steps, the input needs to be compatible (see 7.) with the types stored in one of the start points and the output needs to be compatible with the types stored in one of the end points.
9. For each Step, the previous ManufacturingSystemElement (if existent) shall either be a StoragePoint or the speed of that Step shall be lesser or equal than the speed of the current Step. That is, the speed of the manufacturing system can not suddenly become lower without a storage (bottleneck).

Integrate the validation of the OCL constraints into the meta model using the OCLinEcore editor. For each constraint, describe how the OCL constraint works and list the constraint in the report.

Validate the example model and state which of the constraints are violated for which model elements in the example model and explain why.

You are also allowed to make changes to the meta model if this simplifies the OCL constraints.

Deliverable

- Document (pdf) reporting about the results for the assignment 3 as described above.
- Maximum 4 pages for the content from this assignment, 12 font size
- A ZIP-archive of all plugins you developed/generated.

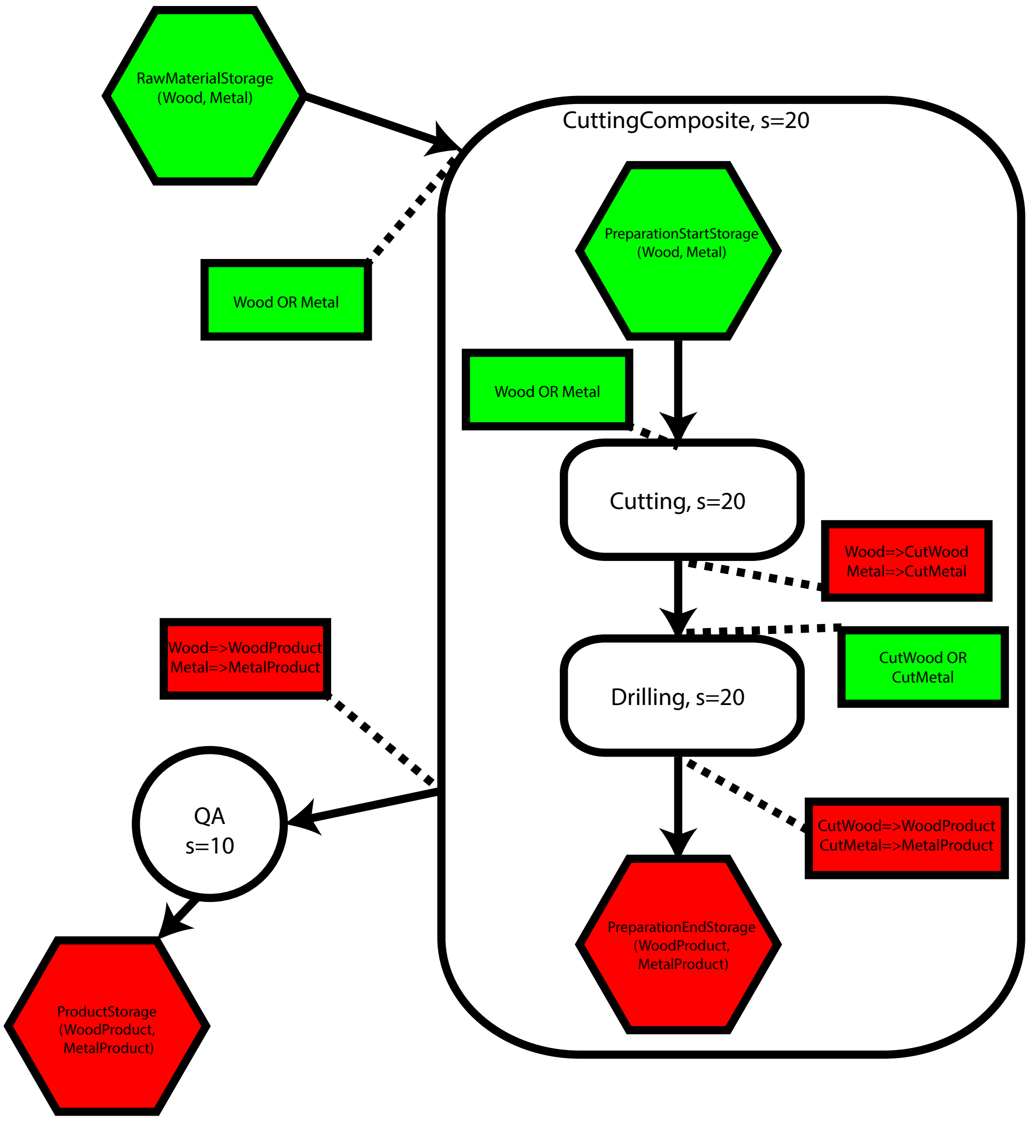
RawMaterialStorage (Wood, Metal)
Wood OR Metal
CuttingComposite, s=20
PreparationStartStorage (Wood, Metal)
Wood OR Metal
Cutting, s=20
Wood=>CutWood Metal=>CutMetal
CutWood OR CutMetal
Drilling, s=20
CutWood=>WoodProduct CutMetal=>MetalProduct
PreparationEndStorage (WoodProduct, MetalProduct)
Wood=>WoodProduct Metal=>MetalProduct
QA s=10
ProductStorage (WoodProduct, MetalProduct)

Figure 1: Example Illustration