Assingment 2

1.

In the given metamodel, the ManufacturingSystem EClass contains one or more ManufacturingSystemElement instances, with this containment relationship named consistsOf. Additionally, ManufacturingSystemElement instances can have transitions to other ManufacturingSystemElement instances. As a result, invoking self.consistsOf.transition returns all transitions among the ManufacturingSystemElements contained within a FurnitureSystem. Upon manual inspection of the model, it is observed that the ManufacturingSystemElements include StoragePoint (such as ProductStorage and RawMaterialStorage) and various types of Step elements (ManufacturingStep, TransportStep, QualityAssuranceStep, and CompositeManufacturingStep). When self.consistsOf.transition() is evaluated, it should return all transitions between these elements. The model reveals the following transitions: from RawMaterialStorage to both CuttingComposite and StartStorageTransport, from StartStorageTransport to CuttingComposite and Cutting, from CuttingComposite to ProductQA, from ProductQA to CuttingCompositeTransport, from CuttingCompositeTransport to ProductStorage, and from ProductStorage back to RawMaterialStorage. The OCLConsole returns all these transitions (see Figure 1).

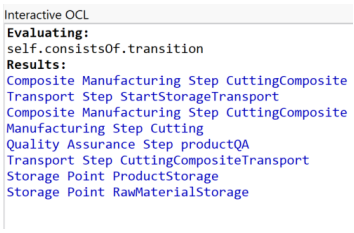

Figure 1: OCLConsole

2.

The second expression is an expression used to check that all ManufacturingSystemElement instances within the FurnitureSystem have a non-null name. In other words, this expression returns true if every ManufacturingSystemElement inside the FurnitureSystem has a name. If even one of them does not have a name, it returns false. During manual inspection, it was observed that each ManufacturingSystemElement had a name, so the expression correctly returned true (see Figure 2).

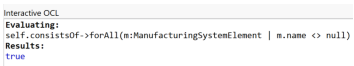

Figure 2: OCLConsole 2.Expression

3.
All classes of type Step have a speed attribute. The third expression is expected to sum the speed values of all instances of type Step within the FurnitureSystem and return the result. When calculated manually, this expression should yield (20 + 20 + 10 + 20) = 70. Executing this expression in the OCL Console also returns 70 as the output (see Figure 3).

Figure 3: OCLConsole 3.Expression

4.

The Step class has references to the InputCondition and OutputDecision EClasses, representing its inputs and outputs, respectively. The fourth expression operates in the context of CuttingCompositeTransport and checks whether all the required inputs for the next step are available. Within CuttingCompositeTransport, there are two types of OutputDecision, each expecting final_wood and final_metal as inputs, respectively. Since both of these inputs are present within the CuttingCompositeTransport step, the fourth expression should evaluate to true (see Figure 4).

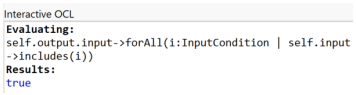

Figure 4: OCLConsole 4.Expression

5. MISSING
6. MISSING