Assignment 1: Meta-Modeling

Software-Languages Group 2

Task 1:

Anbei hinzugefügte Datei

Task 2:

A **ManufacturingSystem** stands for a full production line or nested sub-system:
- It has a name string used to identify the production system.
- It has two **StorageFacilites** as a start and endpoint
- It holds zero or more **Steps** for this system.
- It has any amount of **Persons** as workers

A **Person** works inside a **ManufacturingSystem** and can be responsible for a **Step**:
- It has a name for identification

A **StorageFacility** stores **Workpieces** and declares **WorkpieceTypes**:
- It has an ID and a name for identification
- It has a storage of any amount of **Workpieces**
- It declares any amount of **WorkpieceTypes**

A **WorkPiece** is a Material the System uses to work with:
- It has and ID for identification
- It has a **WorkpieceType** to say its type

A **WorkpieceType** is used to define Types of Workpieces a StorageFacility holds:
- It has a name for its type

A **Step** is an abstract class that has three sub-classes (**ProcessStep**, **TransportStep**, **QualityAssuranceStep**):
- It has a duration
- It can have up to one responsible **Person**
- It has at least one **Workpiece** as an input
- It has at least one **Workpiece** as an output

A **ProcessStep** is an extension of **Step** which is used to Process **Workpieces**:
- It has all attributes of its super class
- It has a **Condition** for its Input
- It can have a whole **ManufacturingSystem** as a Sub-System

TransportStep and **QualityAssuranceStep** have no difference to their super class in this model

A **Condition** is an abstract class that has three sub-classes (**BinaryCondition**, **UnaryCondition**, **WorkpieceTypeCondition**), it creates Boolean conditions to filter **WorkPieceTypes** for **ProcessSteps**:
- It has no attributes of its own

A **BinaryCondition** is a **Condition** that affects two Types:
- It has a **Condition** on its left side
- It has a **Condition** on its right side
- It has a **BinaryOperator** to logically link these Condition

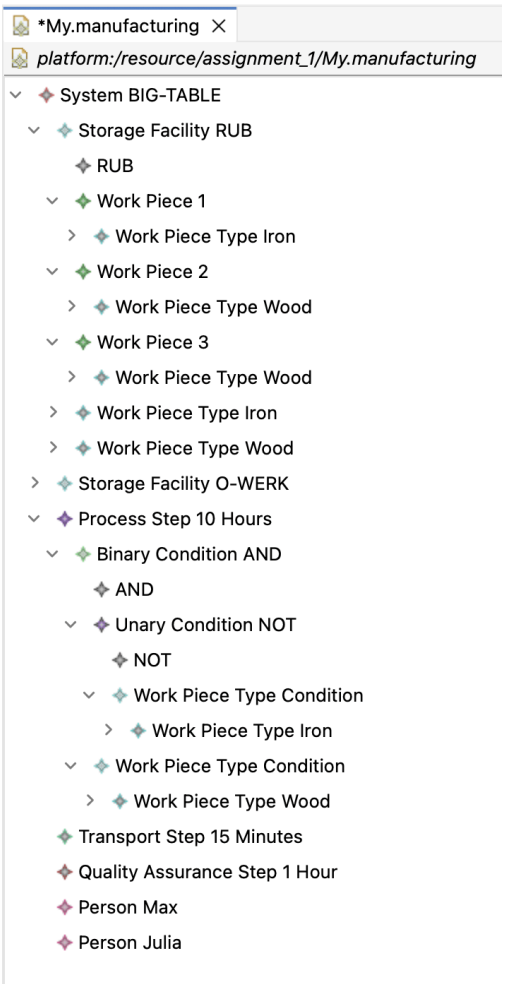
A **UnaryCondition** is a **Condition** that affects one Type:
- It has a **Condition**
- It has a **UnaryOperator** to affect the Condition

A **WorkPieceTypeCondition** is a **Condition** that represents a **WorkPieceType**:
- It has a **WorkPieceType**

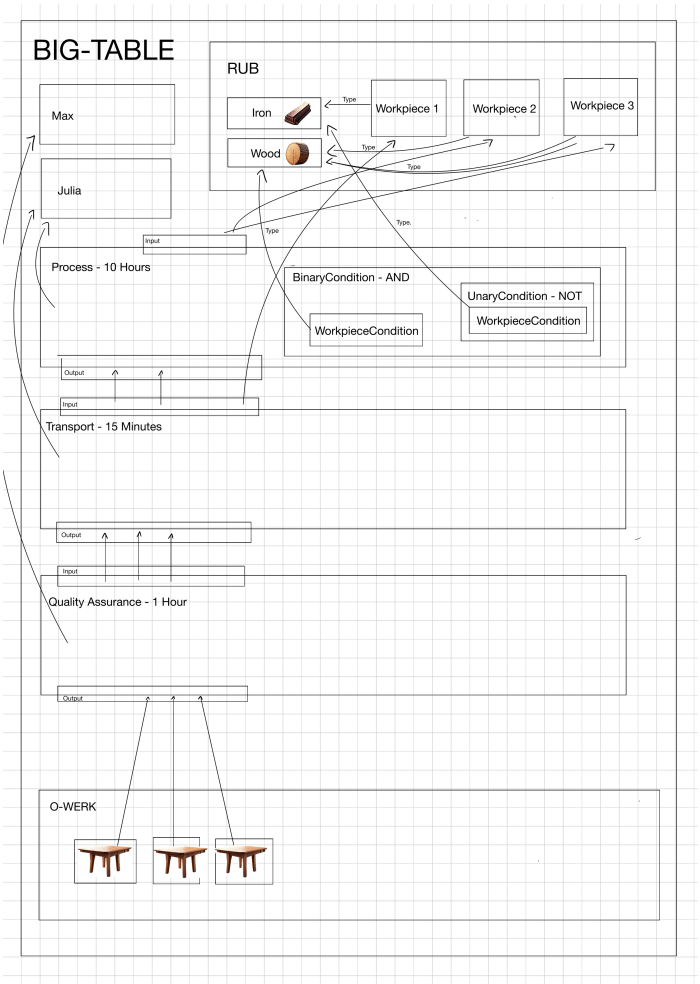
A **BinaryOperator** is an Enum with the „AND“ or „OR“ boolean operator

A **UnaryOperator** is an Enum with the „NOT“ boolean operator

Task 3:



Screenshot of the Tree view



Graphical Illustration of the Example

Task 4:

Our BIG-TABLE system has two storage locations: “Facility RUB” (start) and “O-WERK” (end). A wooden table is to be manufactured for a customer. Our employees Max and Julia are responsible for order processing. Starting in the RUB facility, two wooden workpieces and one iron workpiece are removed.

The first step takes 10 hours. The binary condition „AND“ only accepts materials of the type „Wood“ and another unary condition „NOT“ with „Iron“. (Wood AND NOT Iron) In the second step, the result (output) of the first step and the remaining iron materials are transported. These steps are monitored by Julia.

The last step is quality control (1 hour). Here, Max checks the materials delivered by the transport. After quality control, the manufactured wooden tables are stored in the O-Werk.