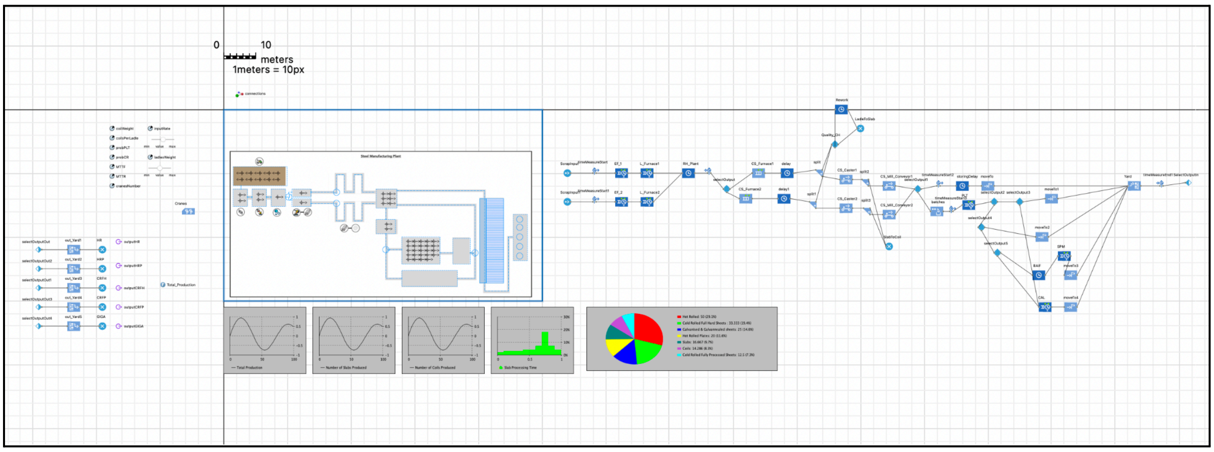
Steel Manufacturing - Assignment #5

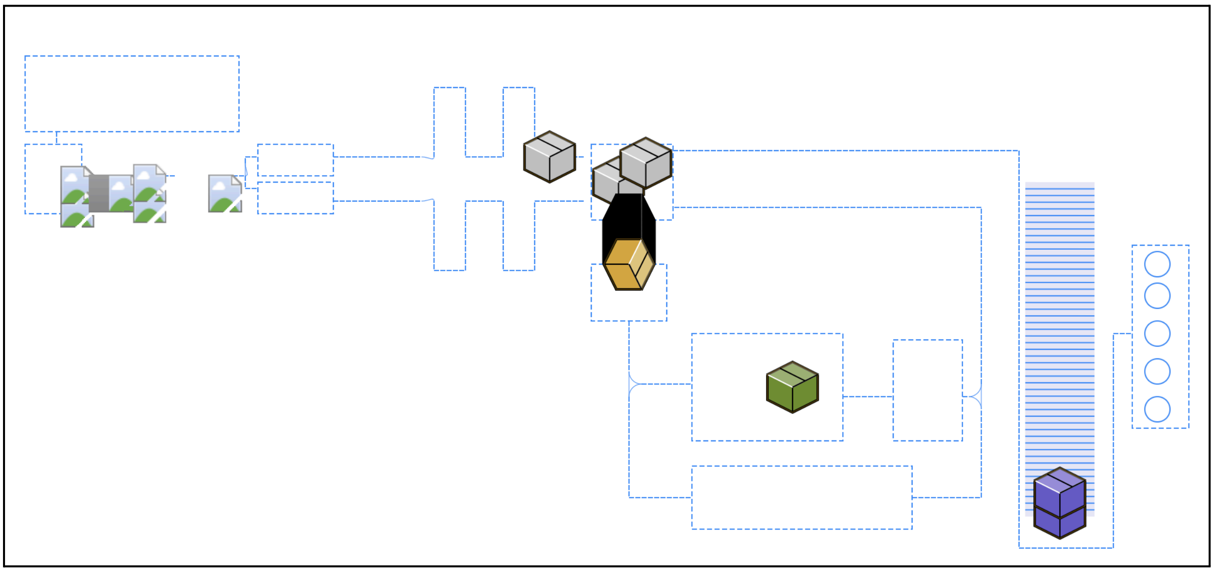
Group 10

Wednesday, 15 January 2024.
Prof. Yilmaz Uygun

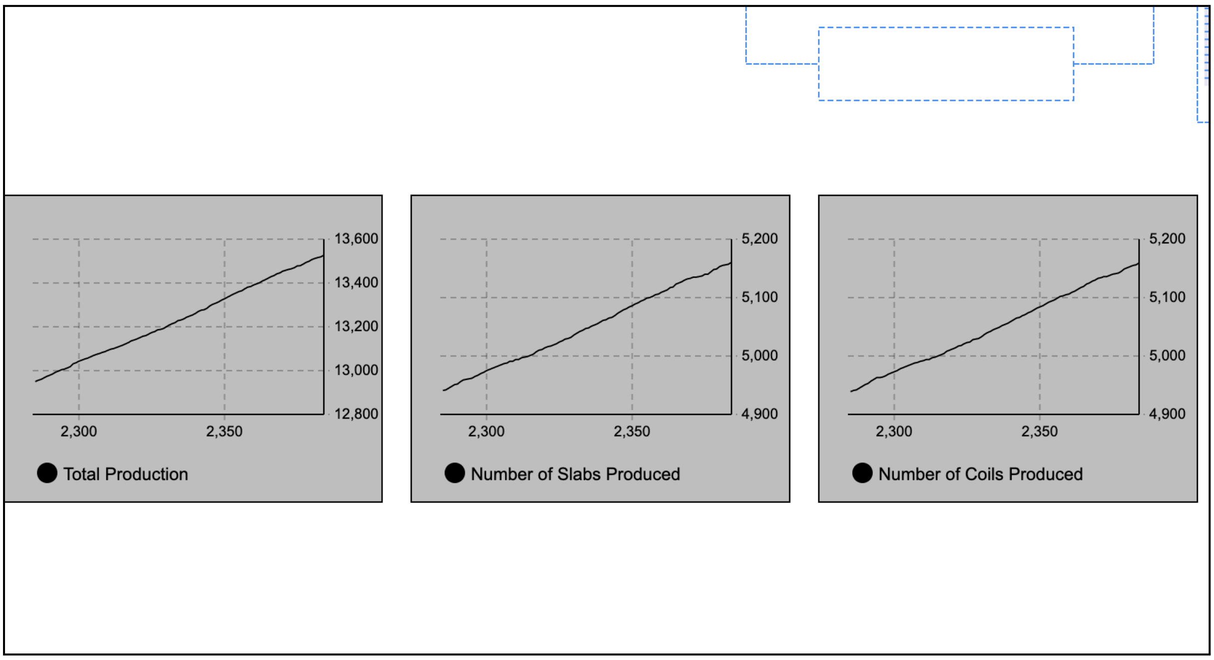
Lis Fazliu
Edin Berisha
Hannah Paulus
Camila Pinto

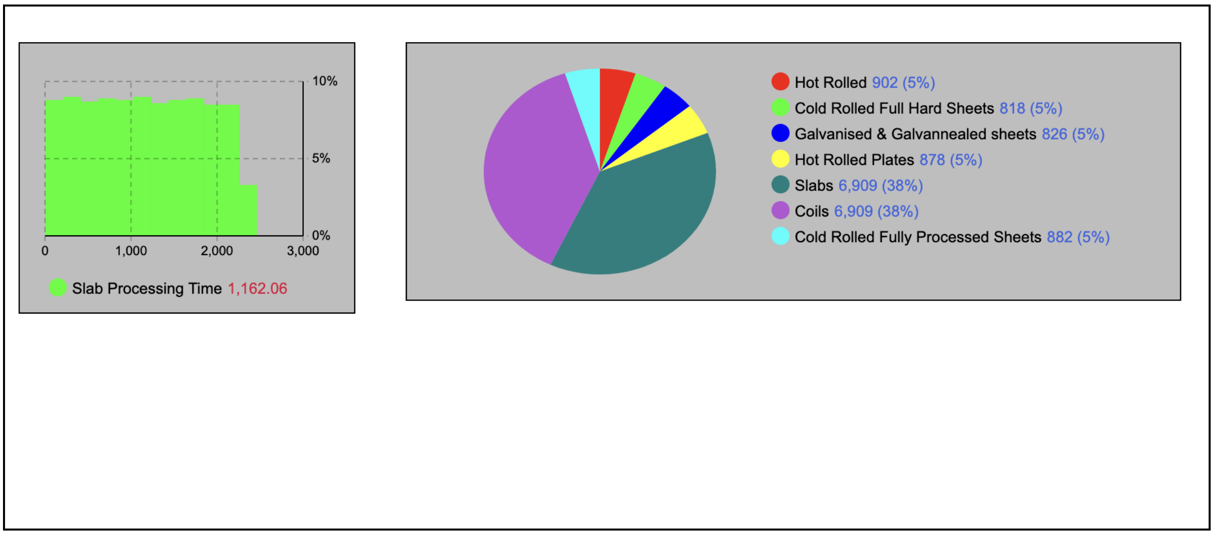
(Done by: Edin Berisha)



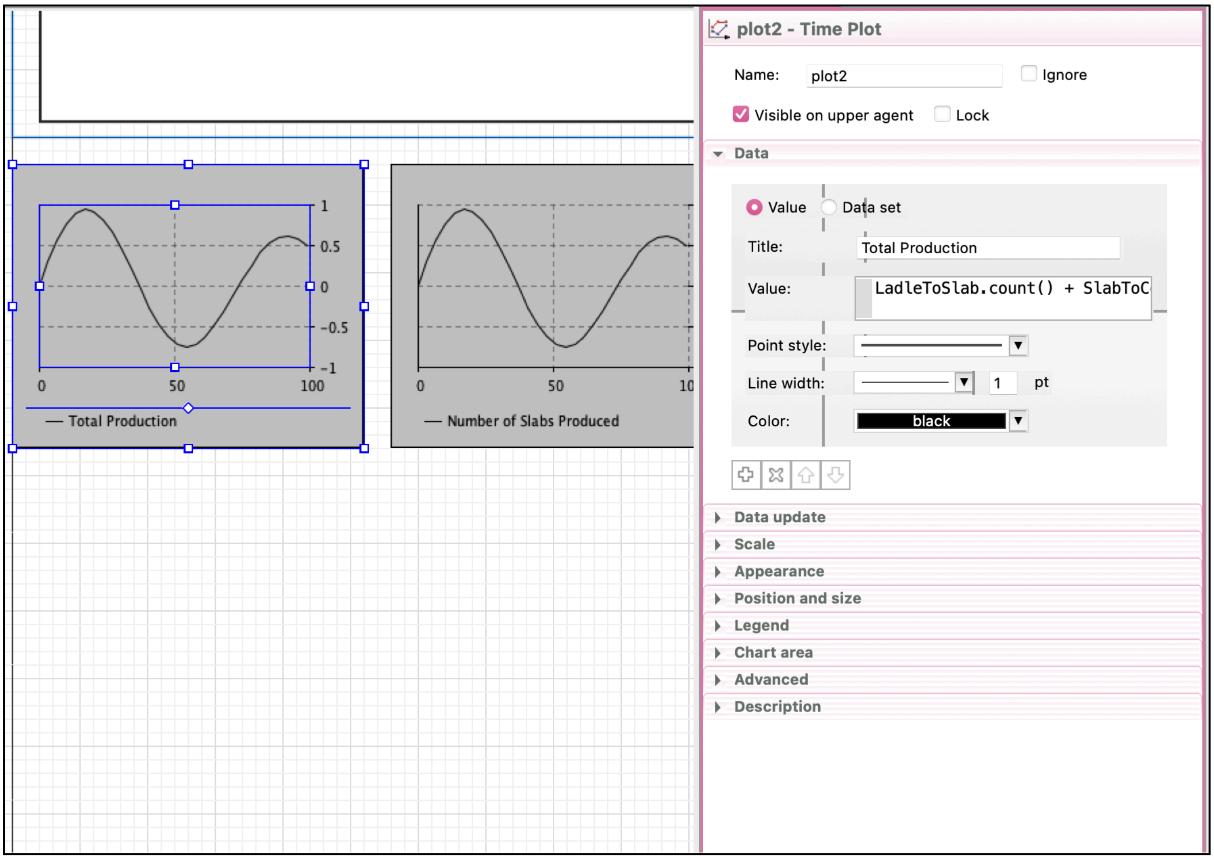


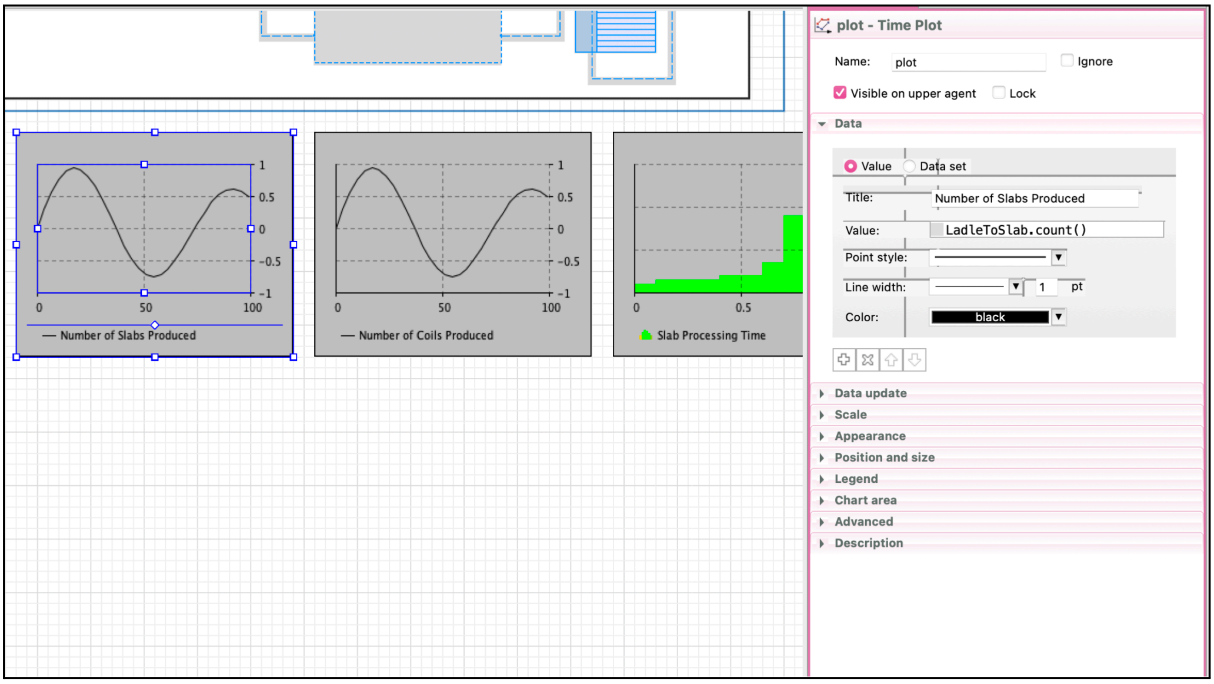
(Done by: Lis Fazliu)



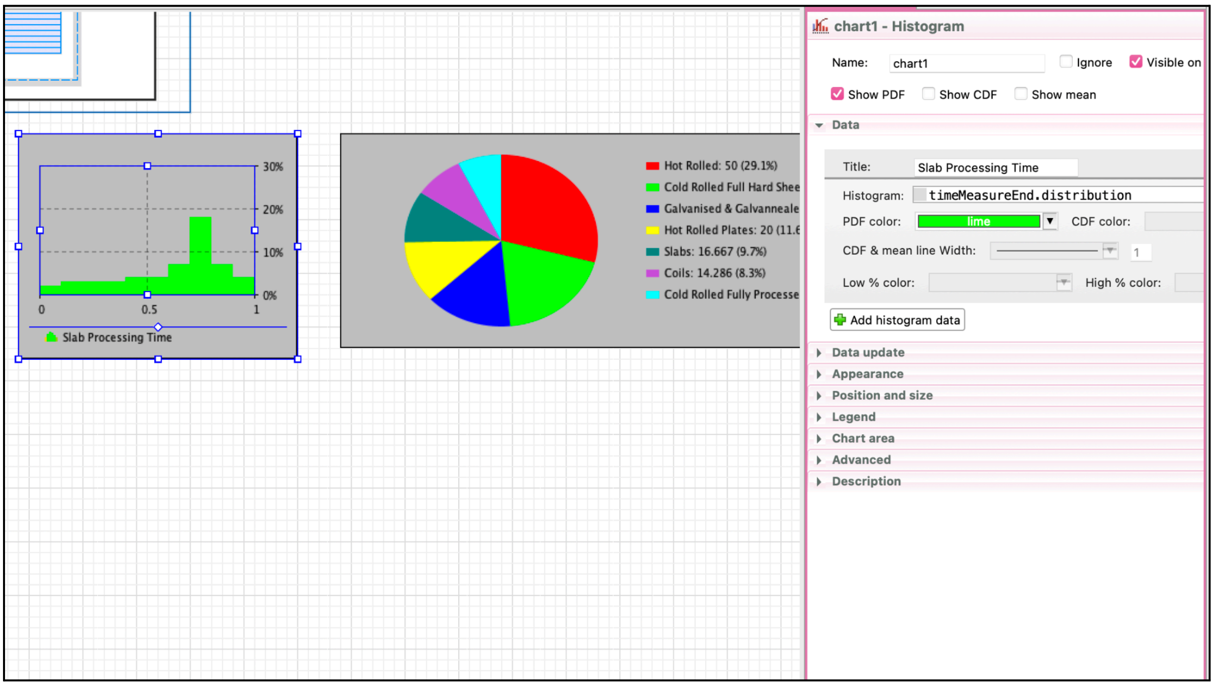


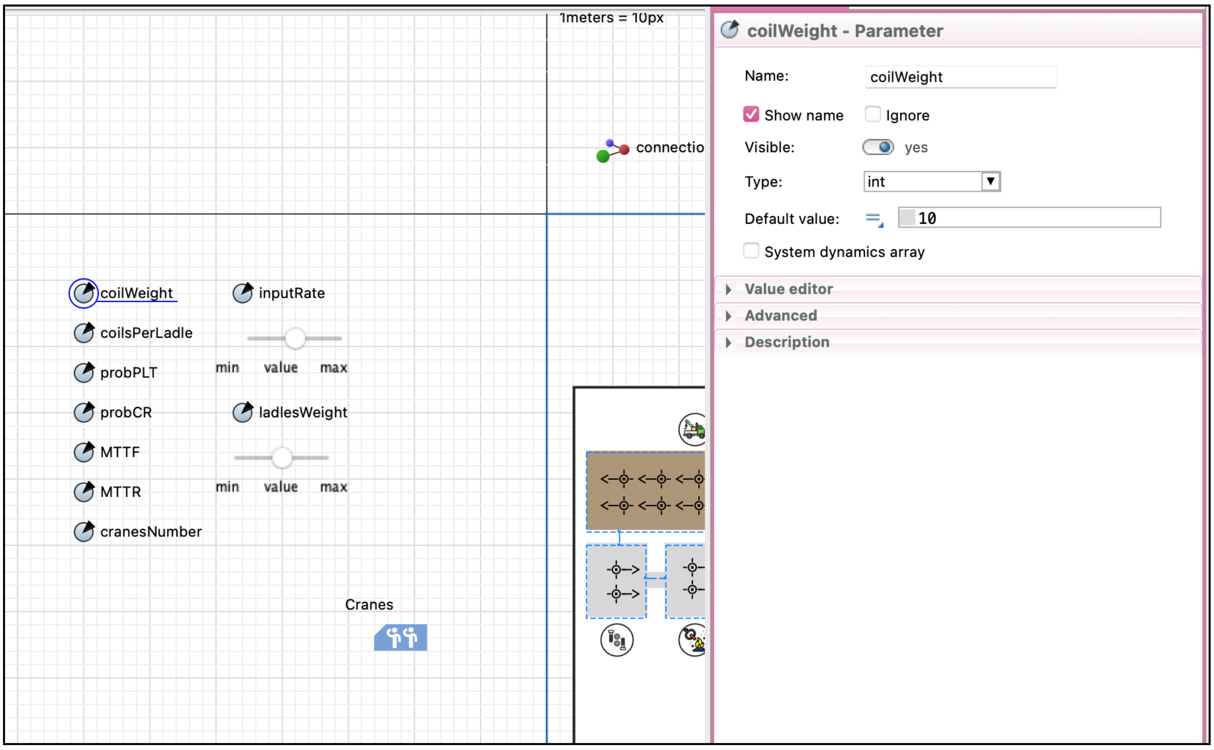
(Done by: Hannah Paulus)





(Done by: Camila Pinto)





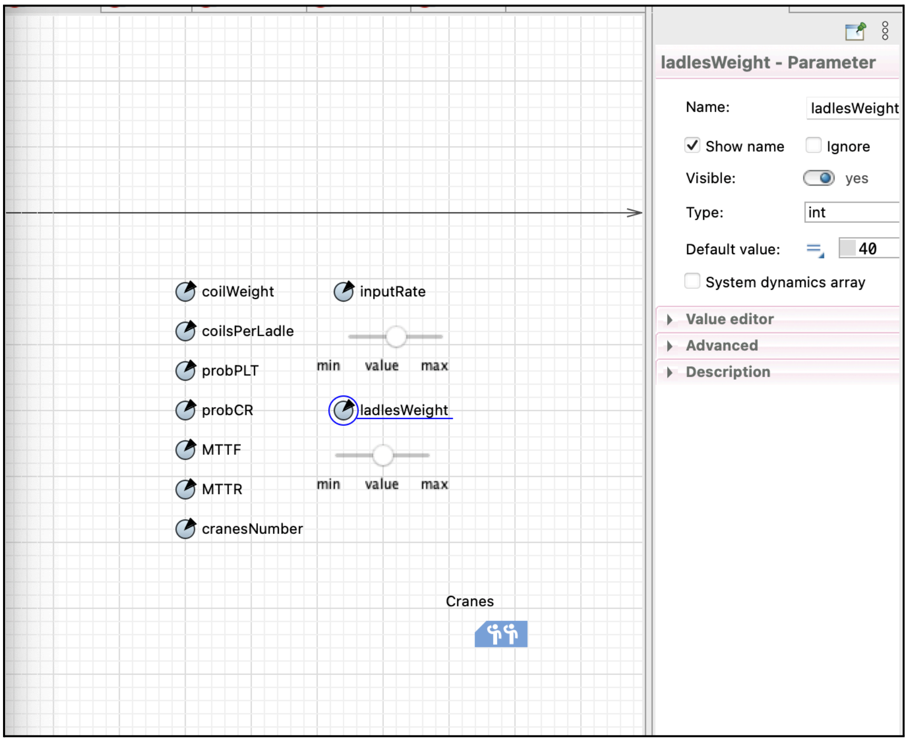
(Done by: Lis, Edin, Hannah, and Camila)

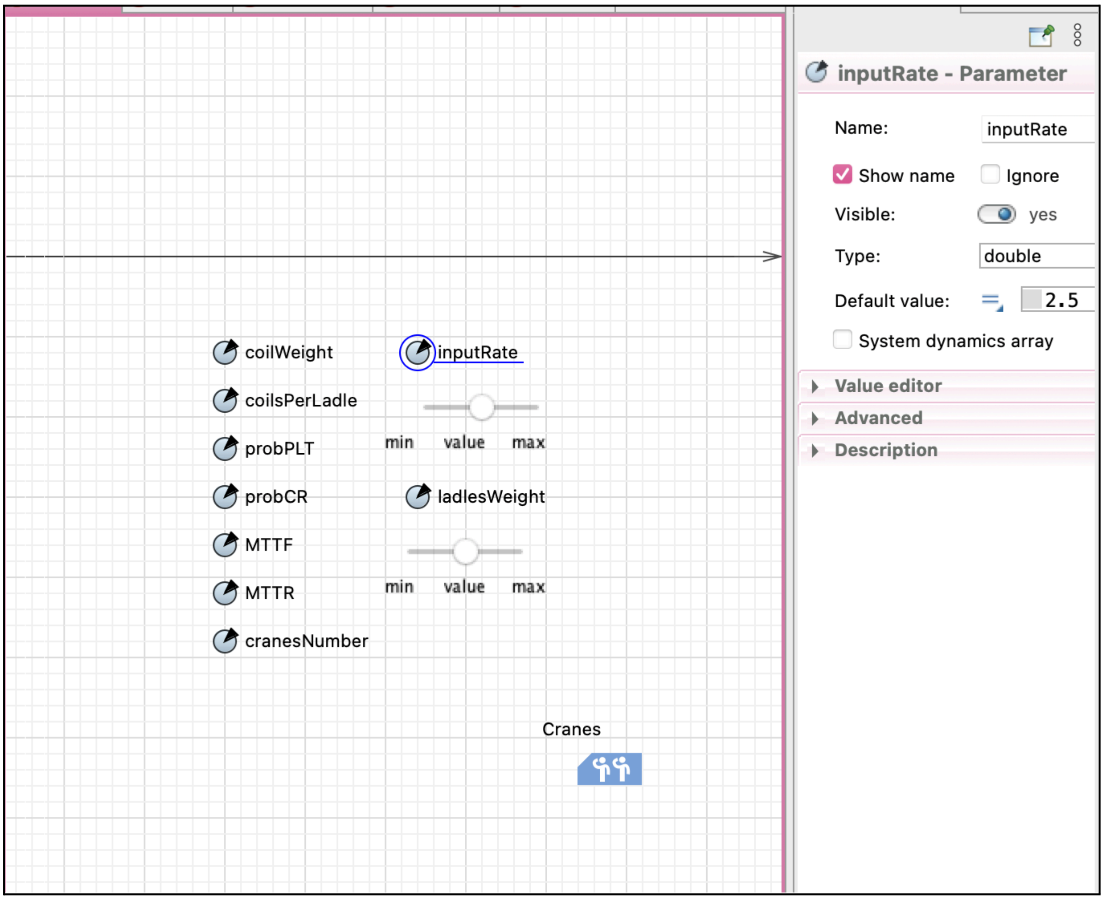
→ **Overview of the Model Visualization**: The images above provide an overall view of the model, highlighting the key components and elements that make up our program. The visuals incorporate a variety of charts to effectively communicate the data being processed. These include metrics such as total production, the number of slabs and coils produced, slab processing time, and a detailed pie chart that breaks down various data sets.

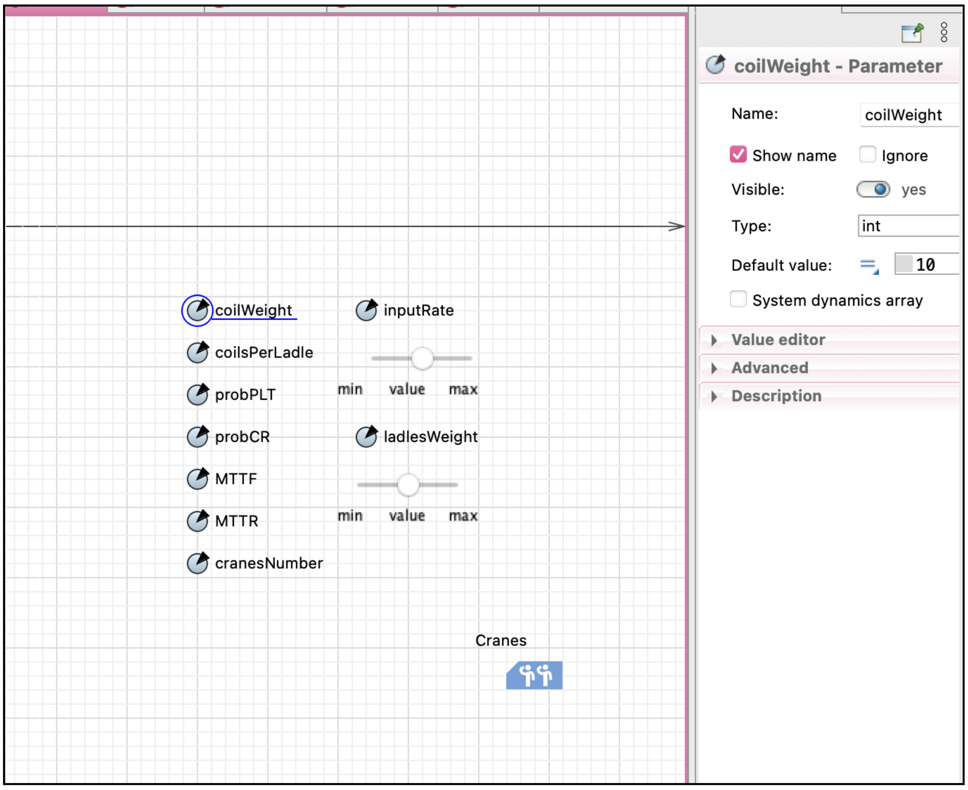
→ **Pie Chart Details and Graph Types**: The pie chart, one of the central visualizations, operates on a recurrence time interval of 60 minutes. Alongside it, other graphs such as histograms and time plot graphs were integrated to provide real-time data insights while the program was running. These visual aids ensure clarity in interpreting the production and operational performance metrics.

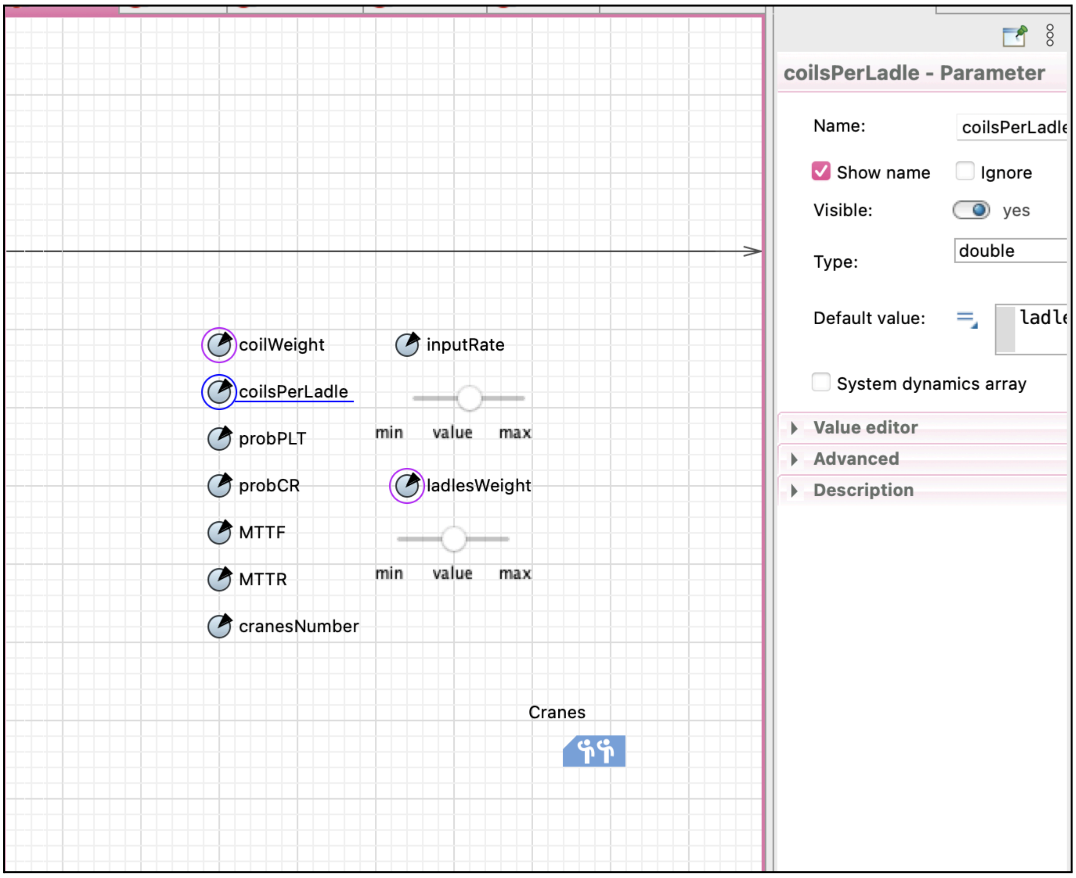
→ **Parameter Specifications**: Several parameters were carefully configured to enhance the model's functionality. The **coil weight parameter** was introduced with a default value of 10. The **coilsPerLadle parameter** was set as a double type to ensure precision, while the **probPLT** and **proCR parameters**, also of double type, were assigned default values of 0.5. For reliability and maintenance tracking, the **MTTF parameter** was set as a double type with a default value of 5, and the **MTTR parameter** had a default value of 1.

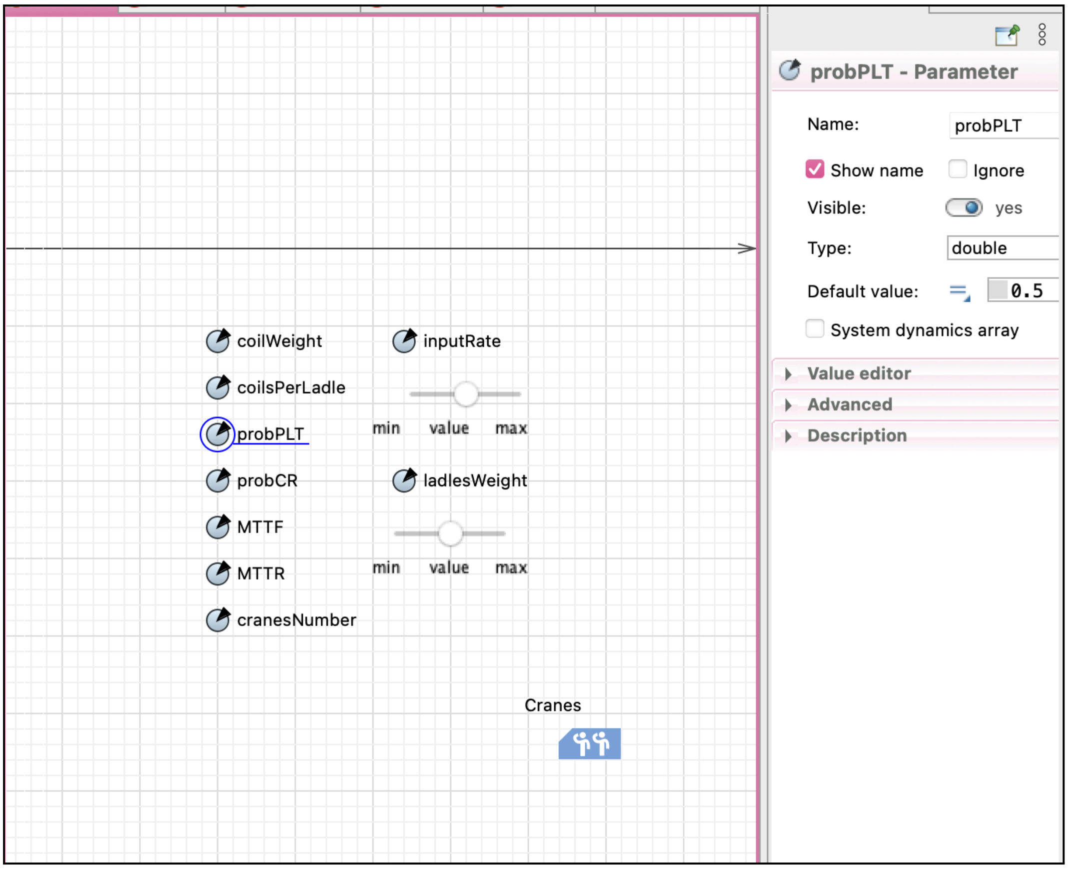
→ **Production and Ladle Parameters**: To control the input rate of materials into the system, the **input rate parameter** was set to 2.5. Additionally, the ladle weight was configured with a default value of 40, ensuring balanced material handling and realistic simulations of production workflows.

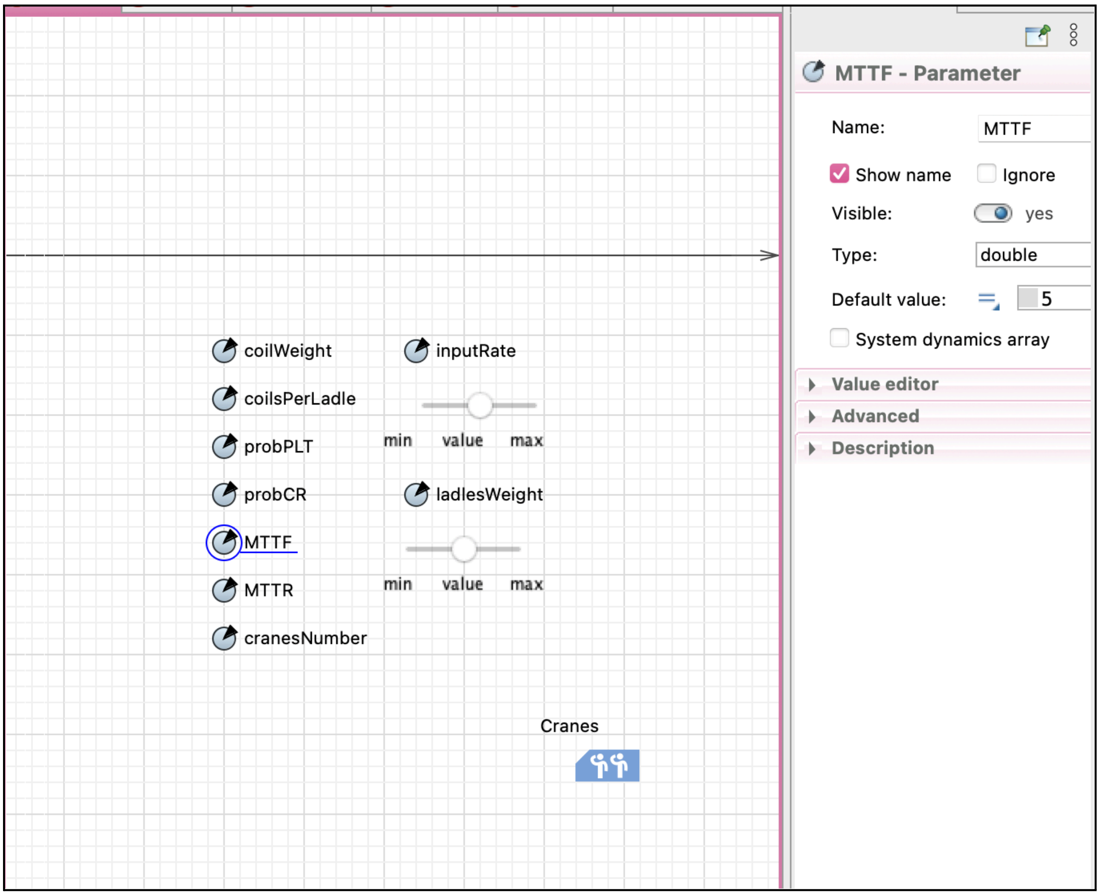


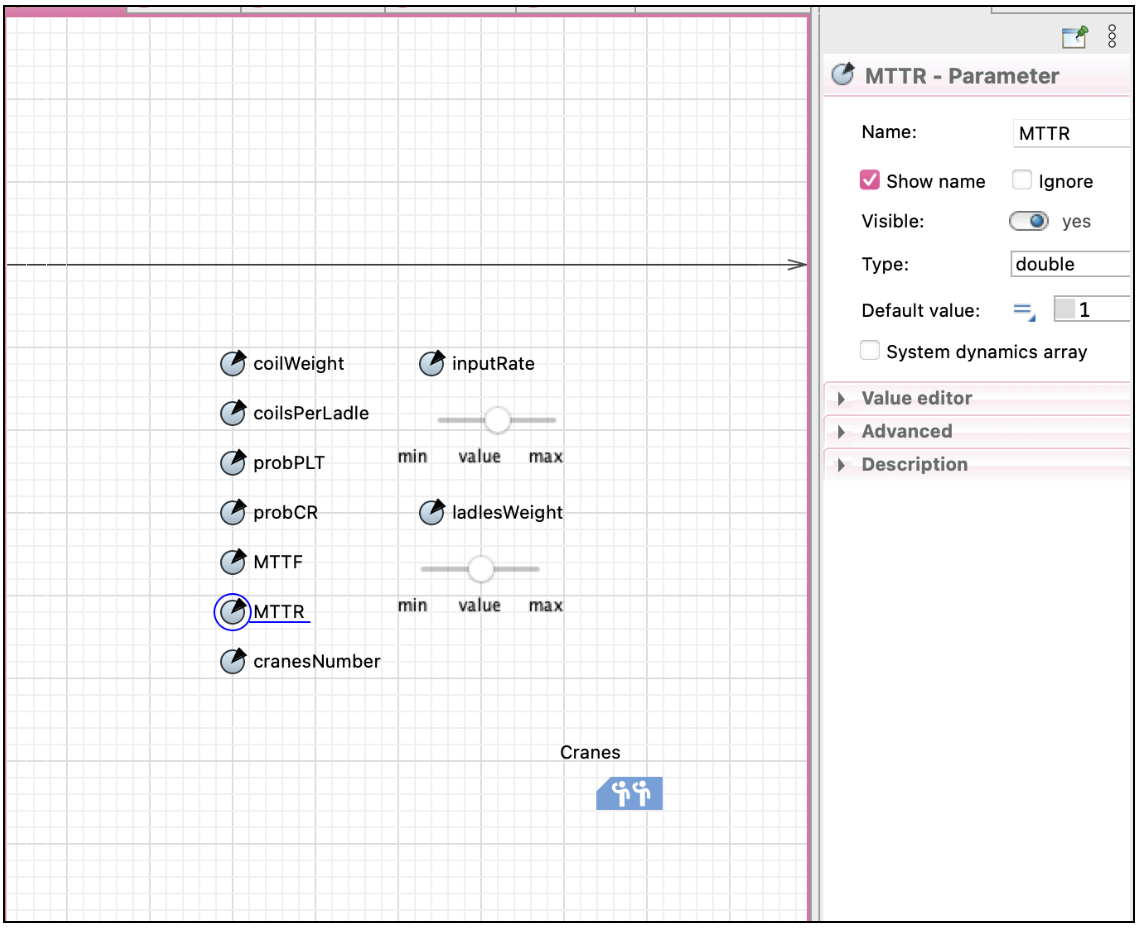



coilWeight - Parameter
Name: coilWeight
Show name
Ignore
Visible: yes
Type: int
Default value: 10
System dynamics array
Value editor
Advanced
Description
coilWeight
inputRate
coilsPerLadle
min value max
probPLT
probCR
ladlesWeight
MTTF
MTTR
min value max
cranesNumber
Cranes

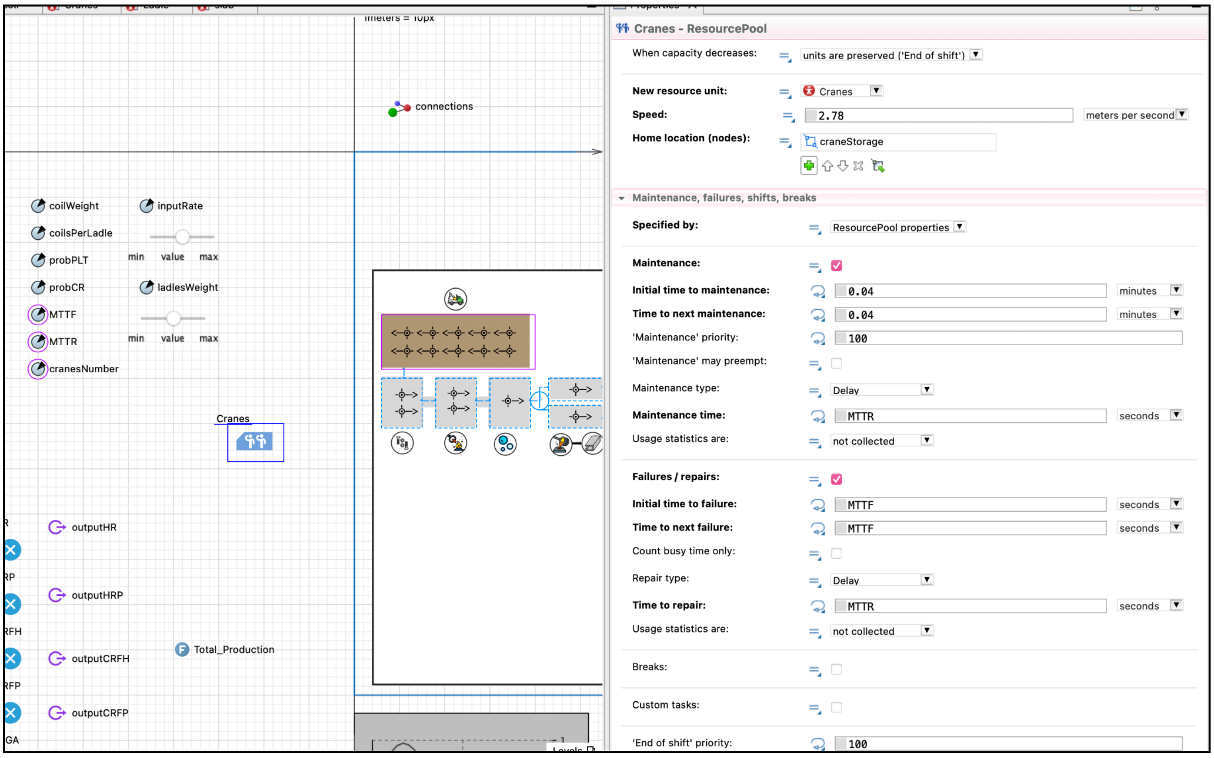

coilsPerLadle - Parameter
Name: coilsPerLadle
Show name
Ignore
Visible: yes
Type: double
Default value: ladle
System dynamics array
Value editor
Advanced
Description
coilWeight
inputRate
coilsPerLadle
min value max
probPLT
probCR
ladlesWeight
MTTF
MTTR
min value max
cranesNumber
Cranes


probPLT - Parameter
Name: probPLT
Show name
Ignore
Visible: yes
Type: double
Default value: 0.5
System dynamics array
Value editor
Advanced
Description
coilWeight
inputRate
coilsPerLadle
probPLT
min value max
probCR
ladlesWeight
MTTF
MTTR
min value max
cranesNumber
Cranes


MTTF - Parameter
Name: MTTF
Show name
Ignore
Visible: yes
Type: double
Default value: 5
System dynamics array
Value editor
Advanced
Description
coilWeight
inputRate
coilsPerLadle
probPLT
min value max
probCR
ladlesWeight
MTTF
MTTR
min value max
cranesNumber
Cranes

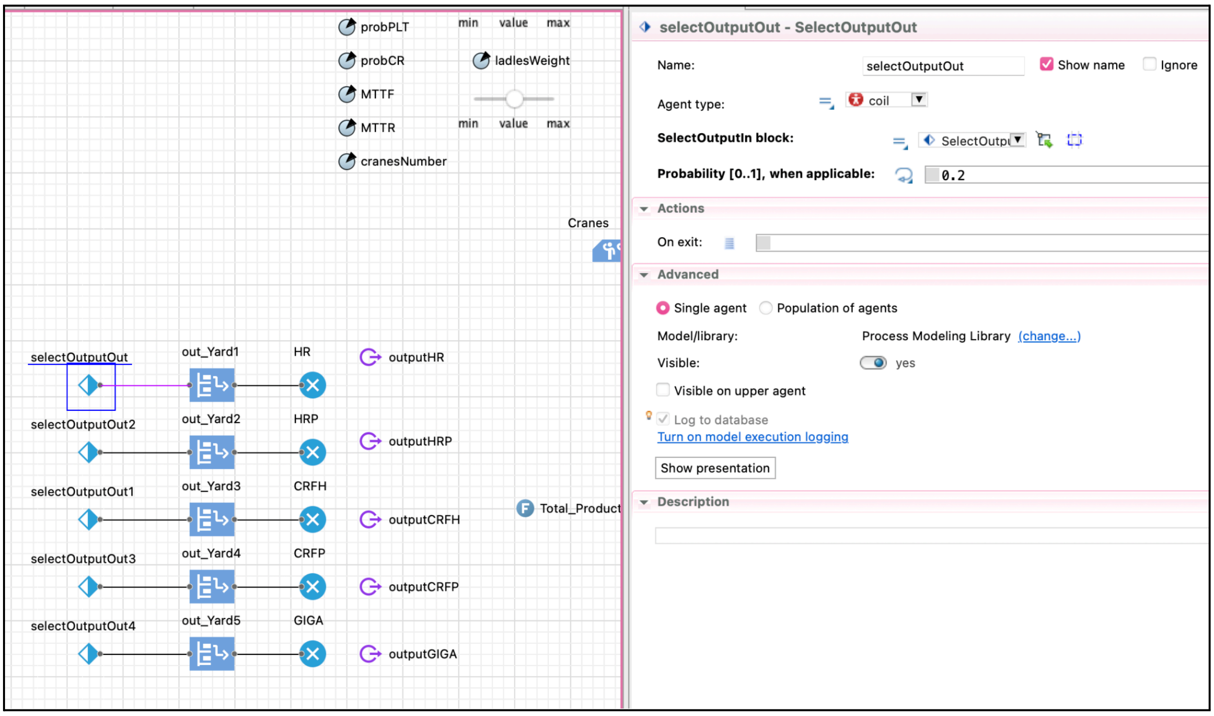


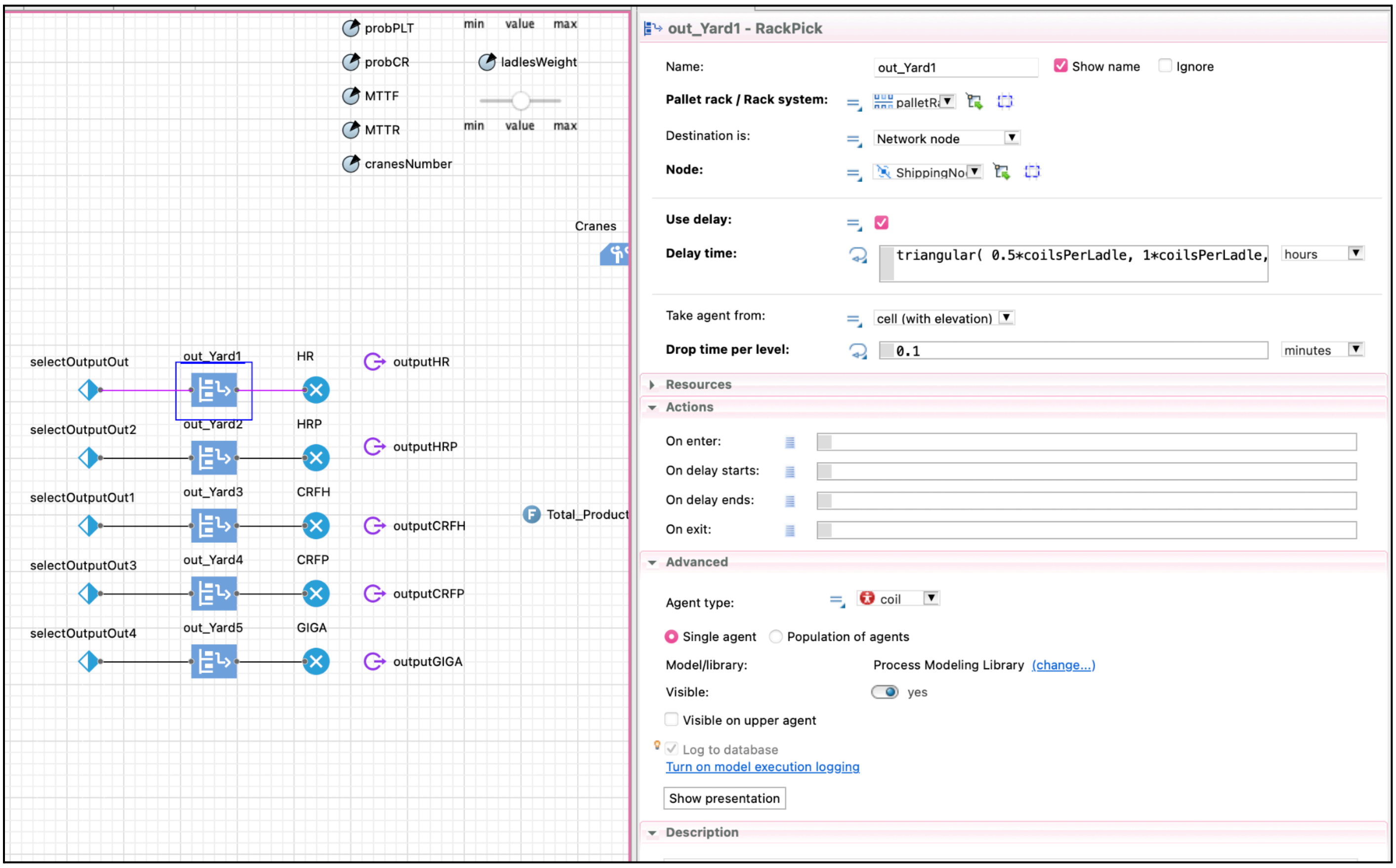
MTTR - Parameter
Name:
MTTR
Show name
Ignore
Visible:
yes
Type:
double
Default value:
1
System dynamics array
Value editor
Advanced
Description
coilWeight
inputRate
coilsPerLadle
min
value
max
probPLT
probCR
ladlesWeight
MTTF
MTTR
cranesNumber
Cranes

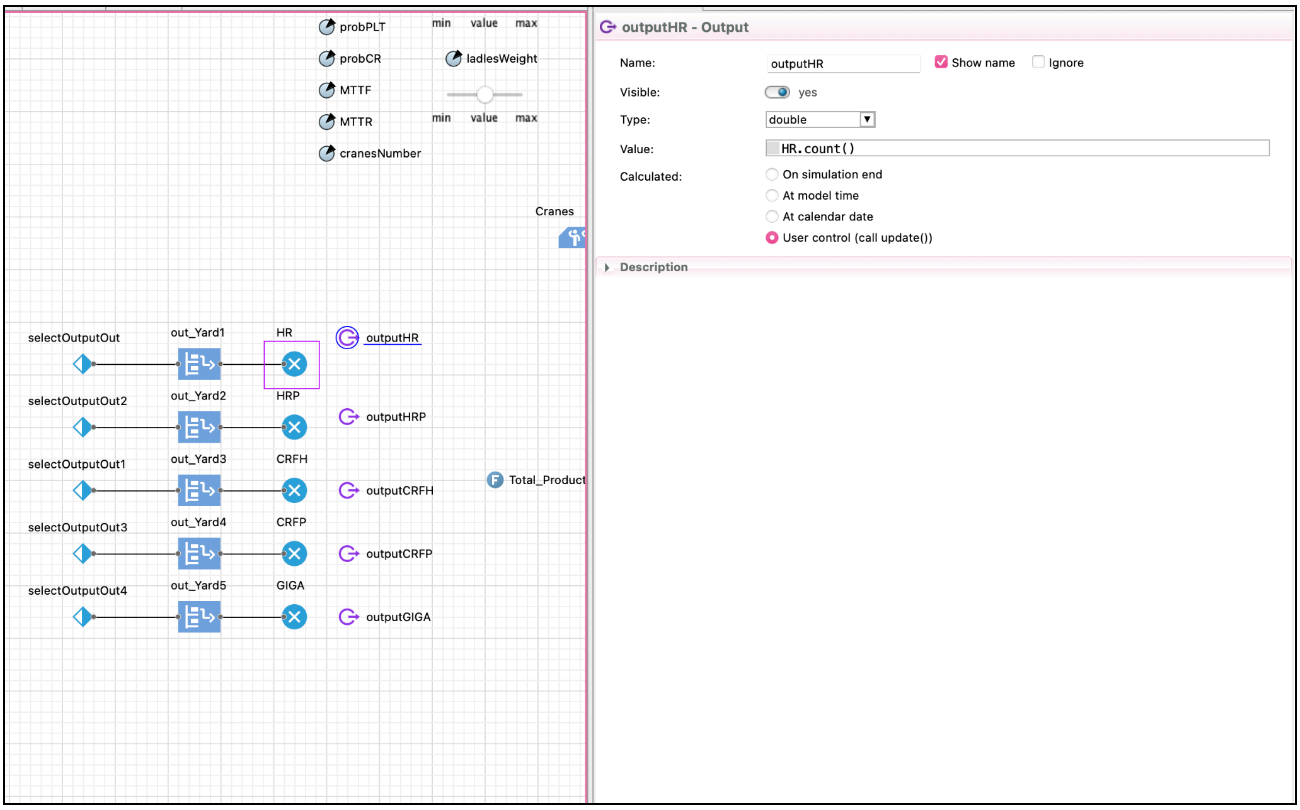


Cranes - ResourcePool
When capacity decreases:
units are preserved ('End of shift')
New resource unit:
Cranes
Speed:
2.78
meters per second
Home location (nodes):
craneStorage
Maintenance, failures, shifts, breaks
Specified by:
ResourcePool properties
Maintenance:
Initial time to maintenance:
0.04
minutes
Time to next maintenance:
0.04
minutes
'Maintenance' priority:
100
'Maintenance' may preempt:
Maintenance type:
Delay
Maintenance time:
MTTR
seconds
Usage statistics are:
not collected
Failures / repairs:
Initial time to failure:
MTTF
seconds
Time to next failure:
MTTF
seconds
Count busy time only:
Repair type:
Delay
Time to repair:
MTTR
seconds
Usage statistics are:
not collected
Breaks:
Custom tasks:
'End of shift' priority:
100
connections
coilWeight
inputRate
coilsPerLadle
probPLT
min
value
max
probCR
ladlesWeight
MTTF
MTTR
cranesNumber
Cranes
outputHR
outputHRP
outputCRFH
Total_Production
outputCRFP

(Done by Edin and Lis)



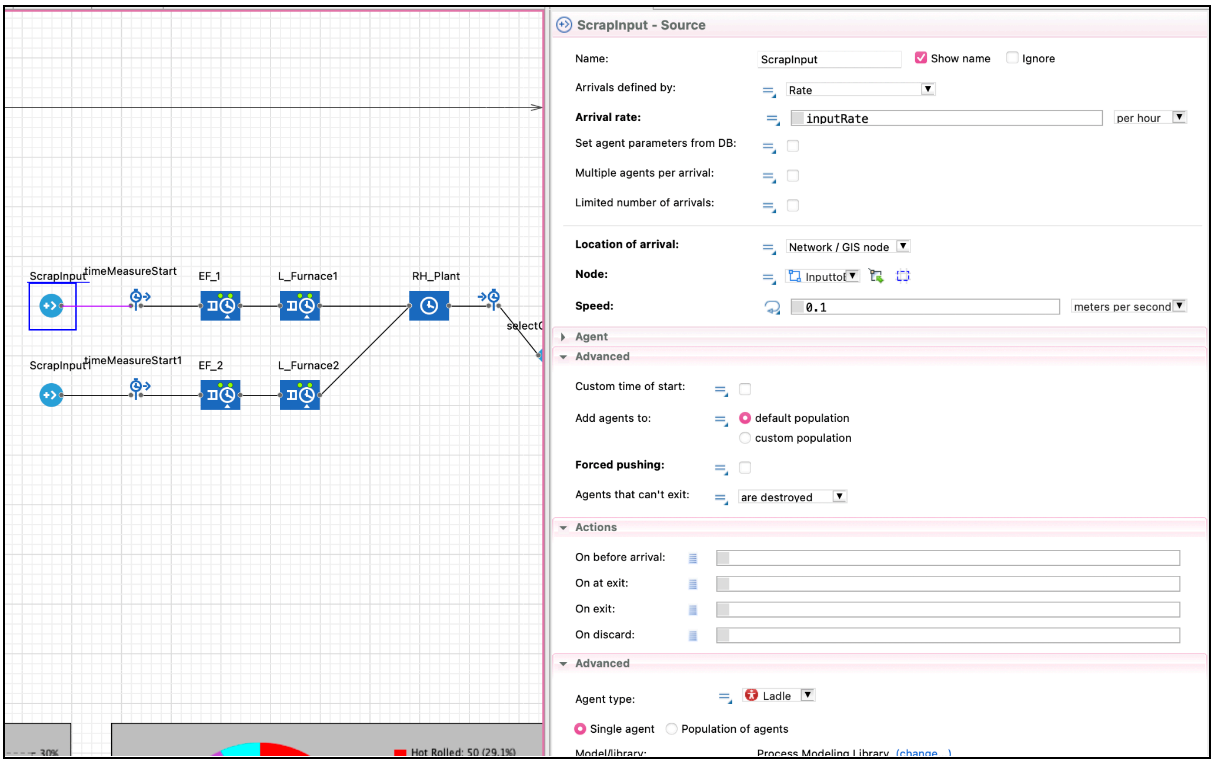


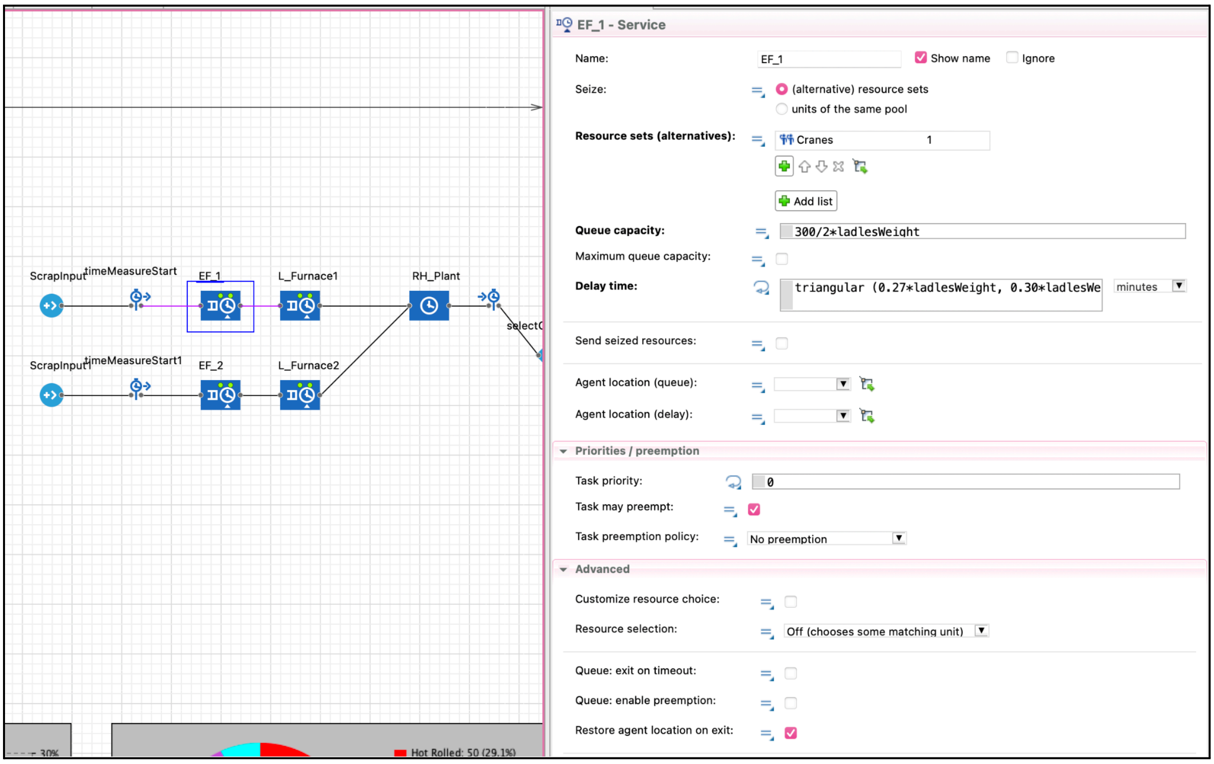


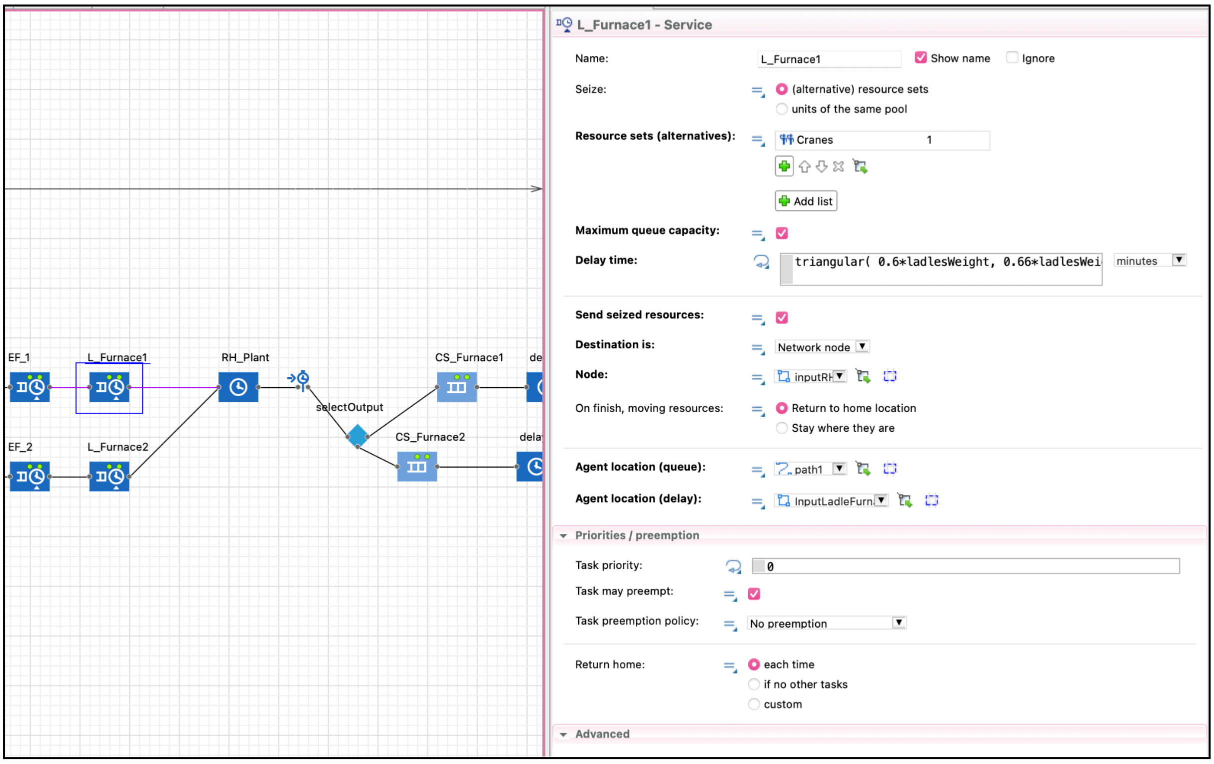
→ **Output Configuration:** For each of the outputs, we used selectOutputOut to control the flow of materials, with a RackPick extending from each output. The RackPick was set to a triangular distribution defined as **triangular(0.5*coilsPerLadle, 1*coilsPerLadle, 1.5*coilsPerLadle)**, allowing variability in the number of coils processed, ranging from half to 1.5 times the coilsPerLadle value. A drop time per level of 0.1 minutes was included to simulate the time required for materials to transfer between levels, ensuring realistic timing in the process.

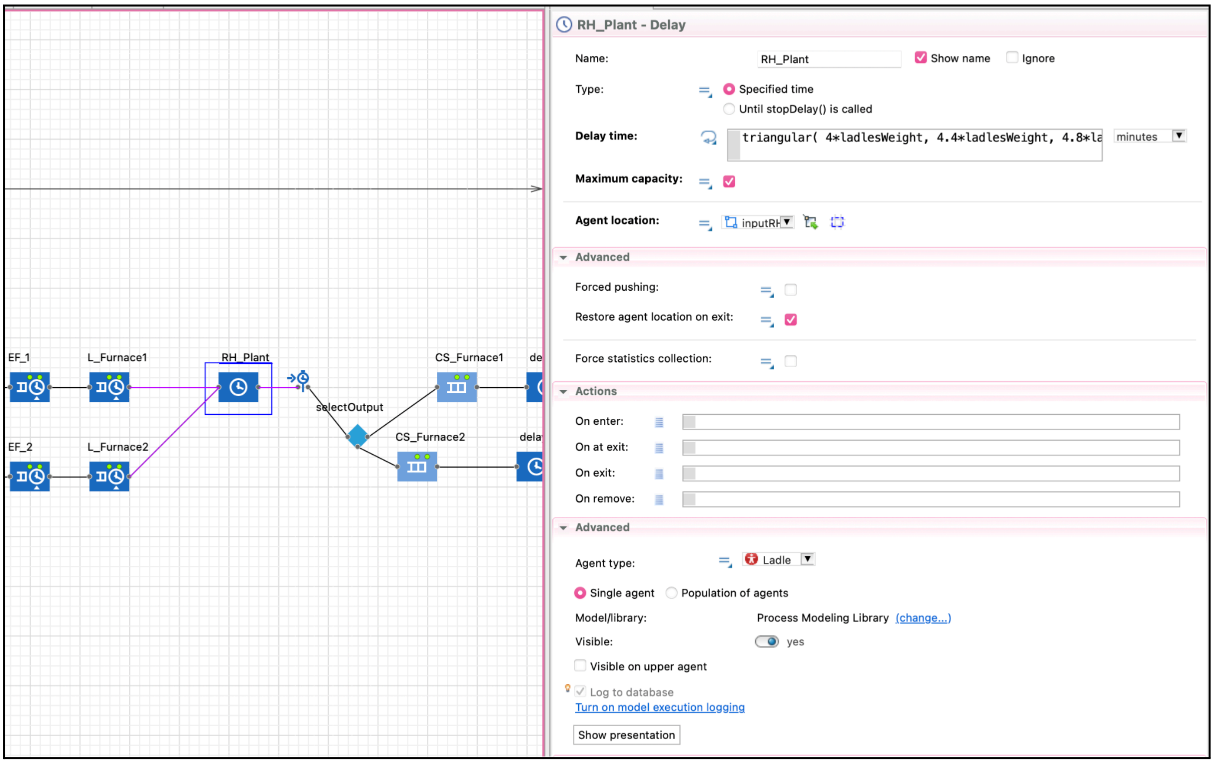
→ Each RackPick was connected to a designated sink—**HR, HRP, CRFH, CRFP, and GIGA**—which represented specific endpoints in the production process. These connections facilitated efficient routing of materials, and at the conclusion of the process, the output for each sink was displayed. This setup provided a comprehensive overview of material distribution, enabling detailed analysis of system performance and ensuring accurate tracking of outputs across all sinks.

(Done by Camila and Hannah)



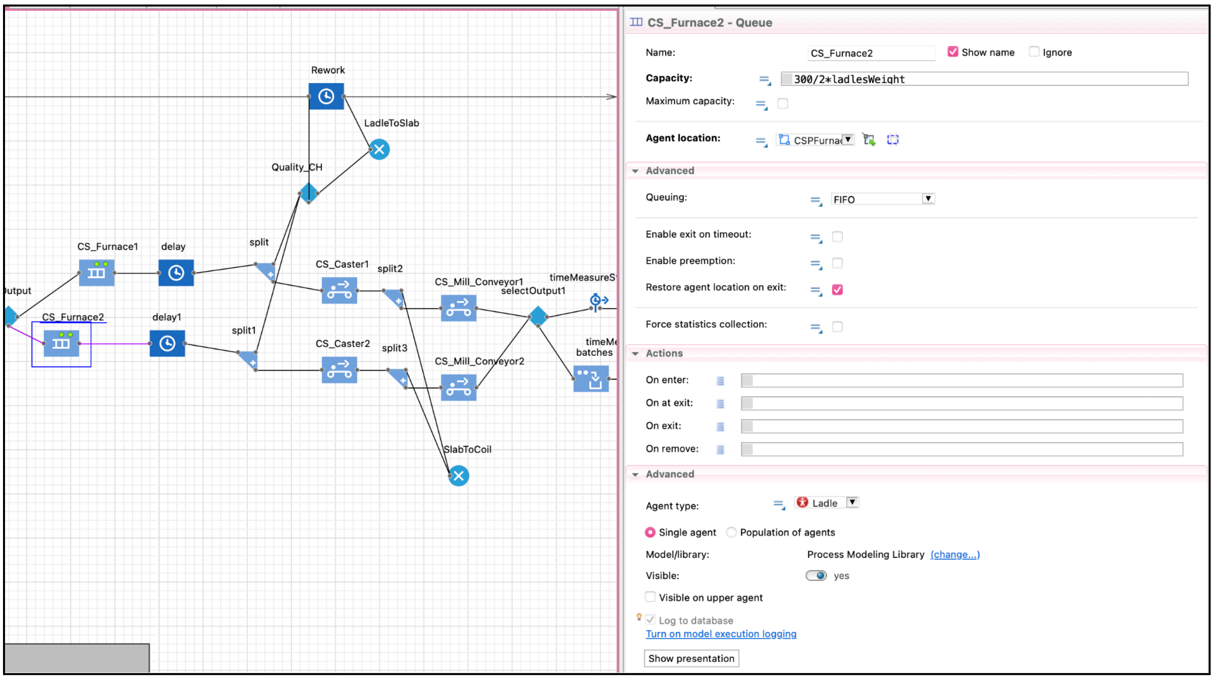


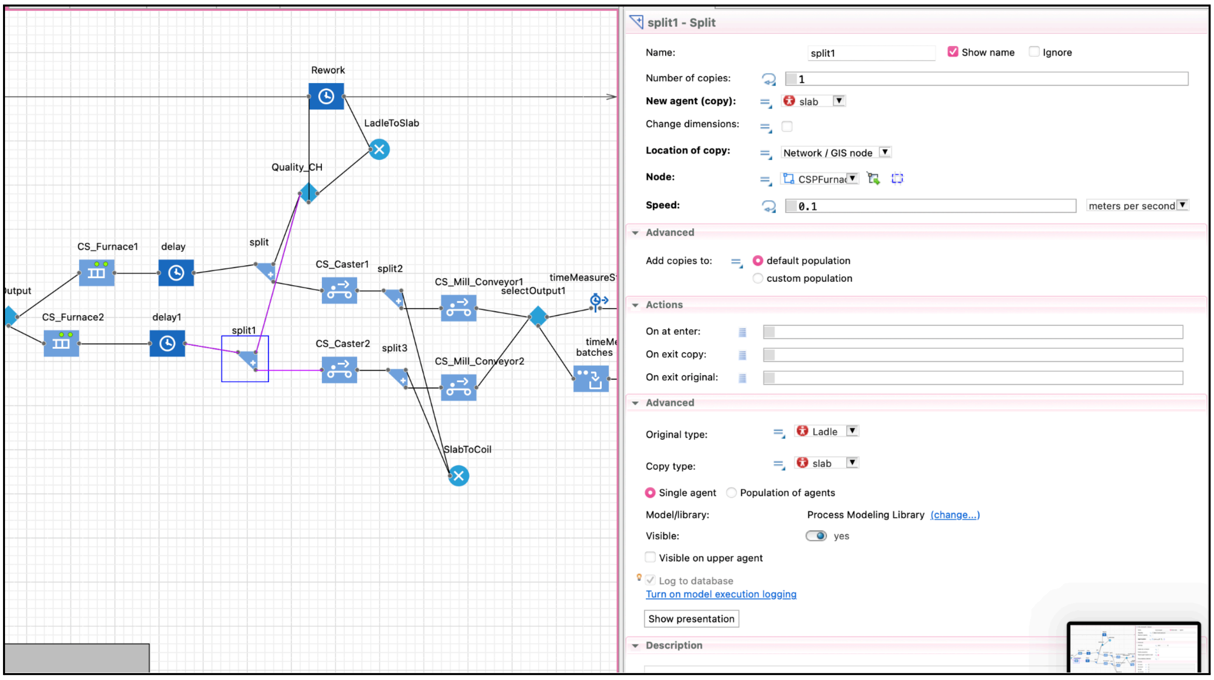




→ The images above depict the initial setup and starting areas of our steel manufacturing plant network. The process begins with a source that introduces materials into the system at a speed of 0.1 meters per second, establishing the foundational flow of operations. Following this, we added two services, EF_1 and EF_2, which are directly connected to cranes. These services were designed with a queue capacity of 300 / 2 × ladlesWeight 300/2×ladlesWeight, ensuring adequate space for handling materials. The delay time for these services was configured using a triangular distribution of $\text{triangular}(0.27 \times \text{ladlesWeight}, 0.30 \times \text{ladlesWeight}, 0.33 \times \text{ladlesWeight})$, providing a realistic range for processing times. Additionally, we incorporated two more services, L_Furnace 1 and L_Furnace 2, which are also connected to the cranes. These furnaces were set with delay times following a triangular distribution of $\text{triangular}(0.6 \times \text{ladlesWeight}, 0.66 \times \text{ladlesWeight}, 0.72 \times \text{ladlesWeight})$, simulating their operational variability. After passing through these initial stages, the materials converge at the RH_plant delay. This stage was configured with a delay time of $\text{triangular}(4 \times \text{ladlesWeight}, 4.4 \times \text{ladlesWeight}, 4.8 \times \text{ladlesWeight})$, representing the final processing phase in this section of the network before moving forward.

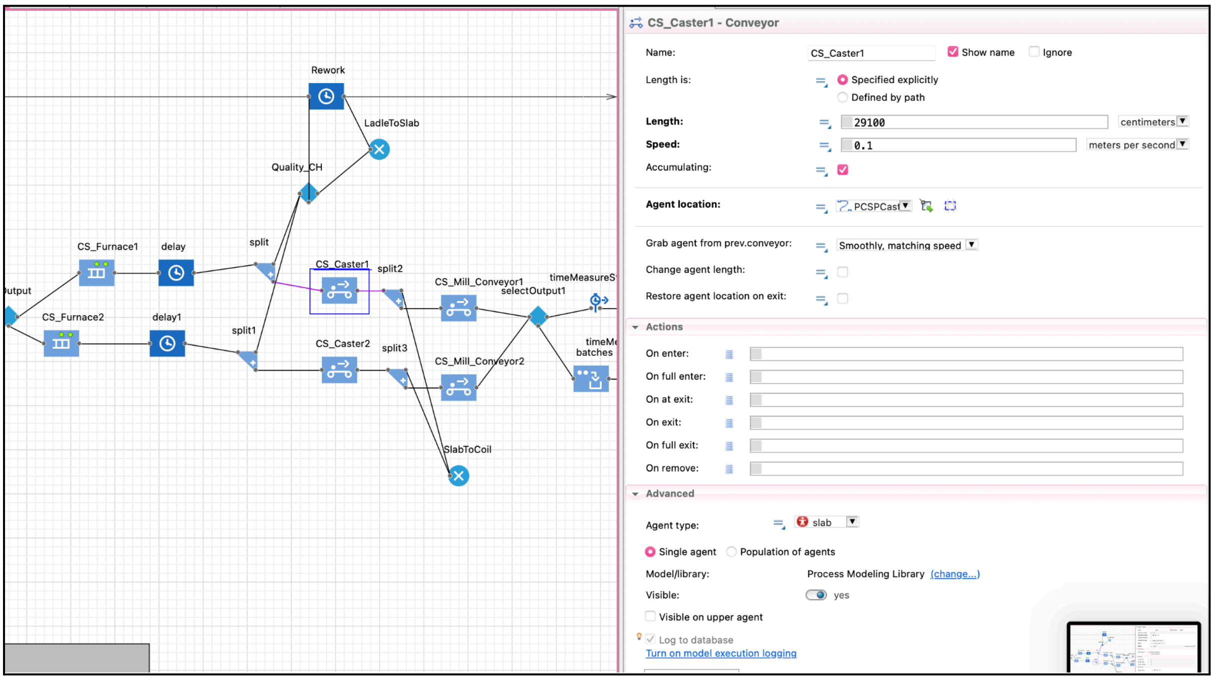
(Done by Lis)

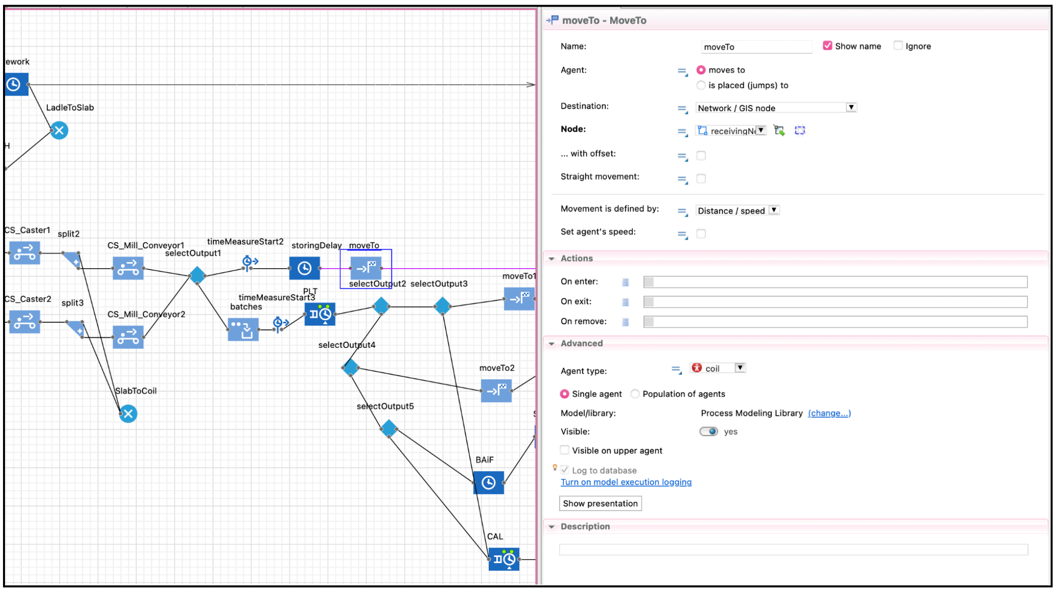




→ We proceeded by incorporating two Queues, named CS_Furnace1 and CS_Furnace2, each designed with a capacity of 300/2×ladlesWeight300/2×ladlesWeight. These queues were carefully configured to ensure adequate space for managing material flow and balancing the workload between the two furnaces. Additionally, we added two Splits into the network to further distribute materials efficiently. Each Split was set with a speed of 0.1 meters per second to maintain a consistent flow rate. The location for the copy operation of these Splits was set to network/gis node, ensuring that the positioning and replication of the splits aligned seamlessly with the network's structure. These additions enhanced the system's ability to handle materials dynamically and ensured an organized transition between processing stages, supporting the overall efficiency of the plant's workflow.

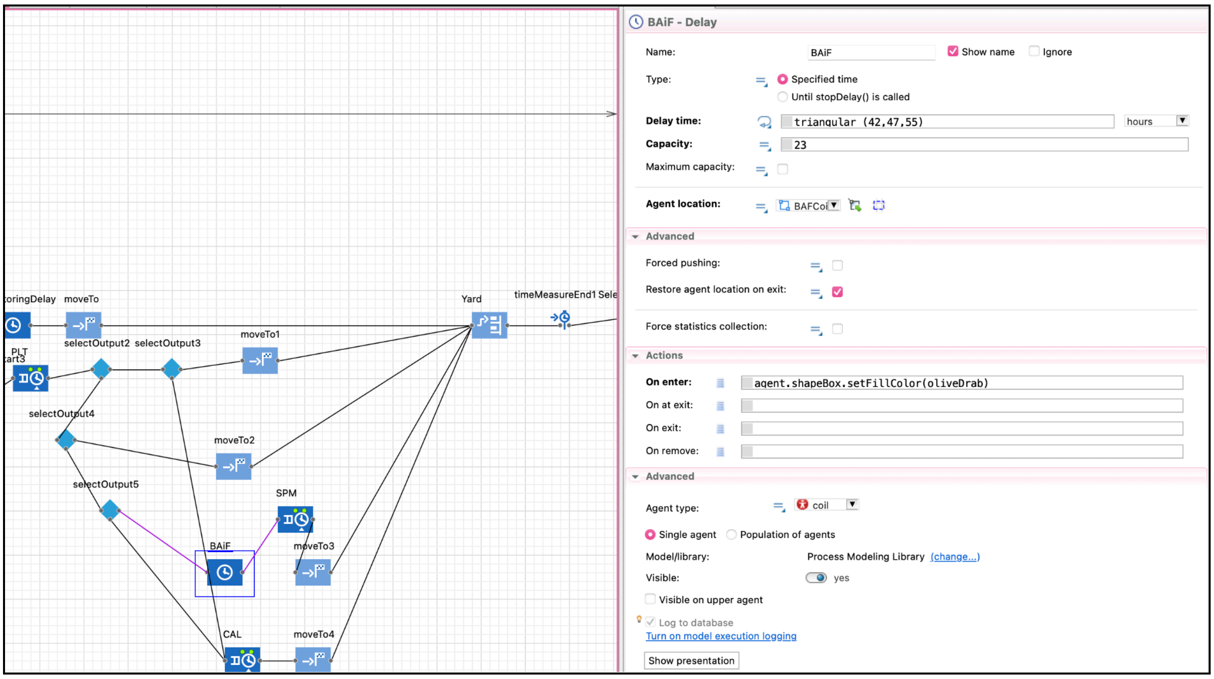
(Done by Edin)

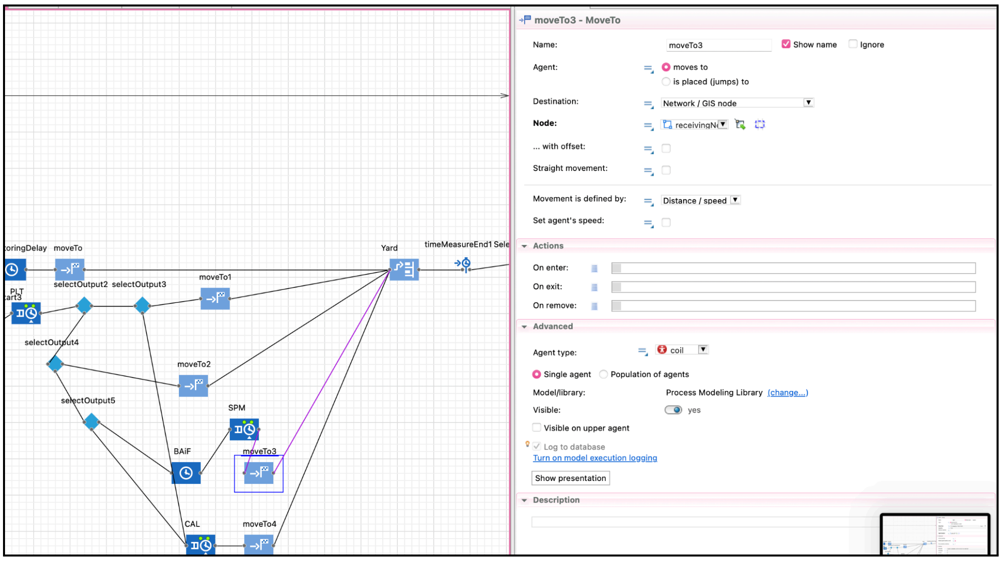




→ In this section, we added conveyors with lengths of 29,100 units, each configured to operate at a speed of 0.1 meters per second. These conveyors were designed to facilitate the smooth and consistent movement of materials across the system, ensuring an efficient transfer between stages of the production process. To enhance the functionality of this section, we included a "move to" feature, which was strategically connected to the receiving node as its starting point. The destination for this feature was set to the network/gis node, enabling precise control over the material flow and ensuring alignment with the system's spatial layout. These additions were crucial for maintaining the connectivity and operational efficiency of the manufacturing network, optimizing the transition of materials through this stage.

(Done by Hannah)

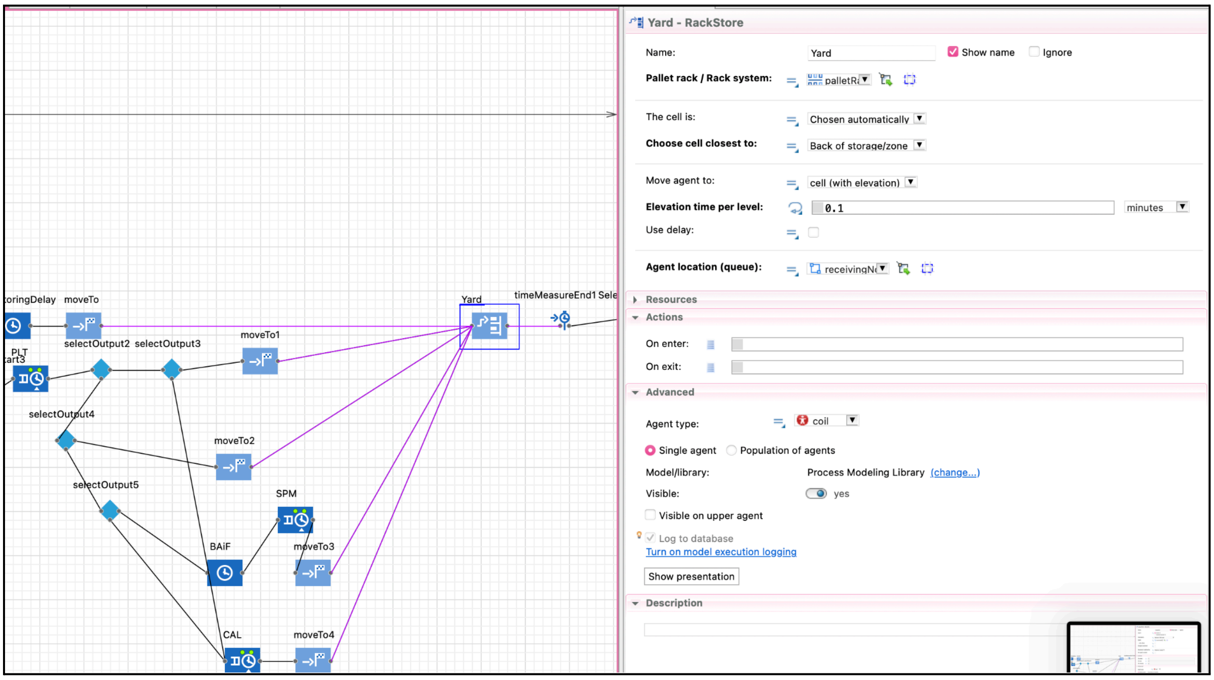


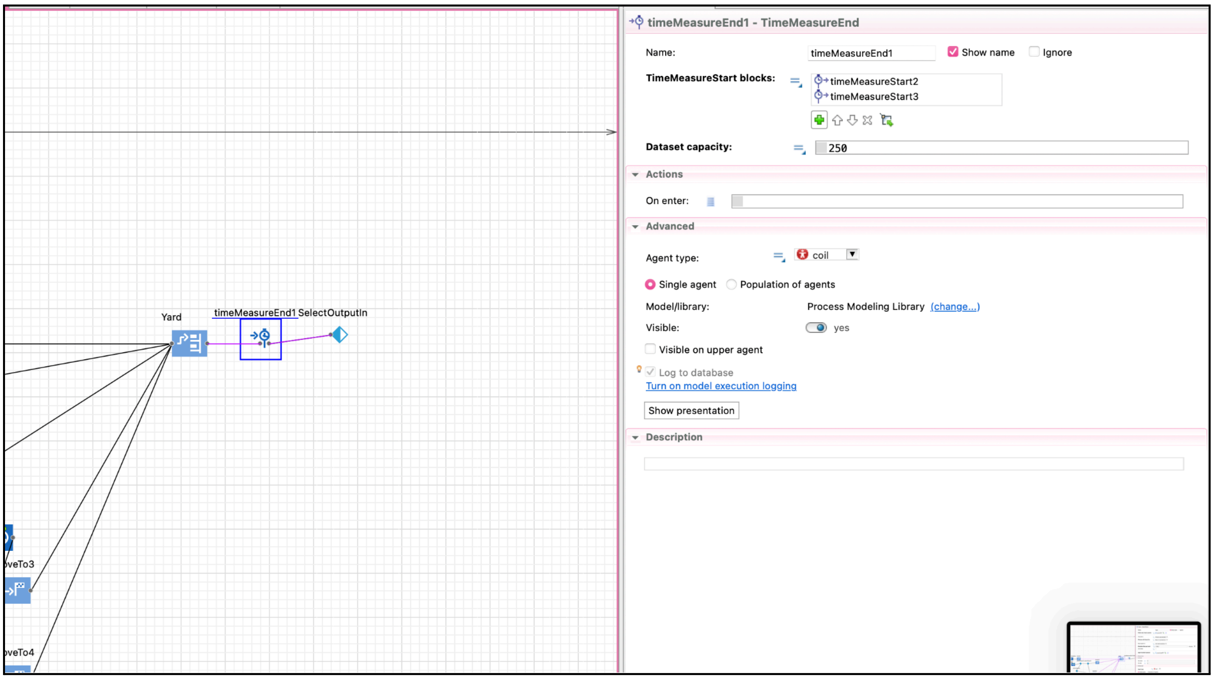


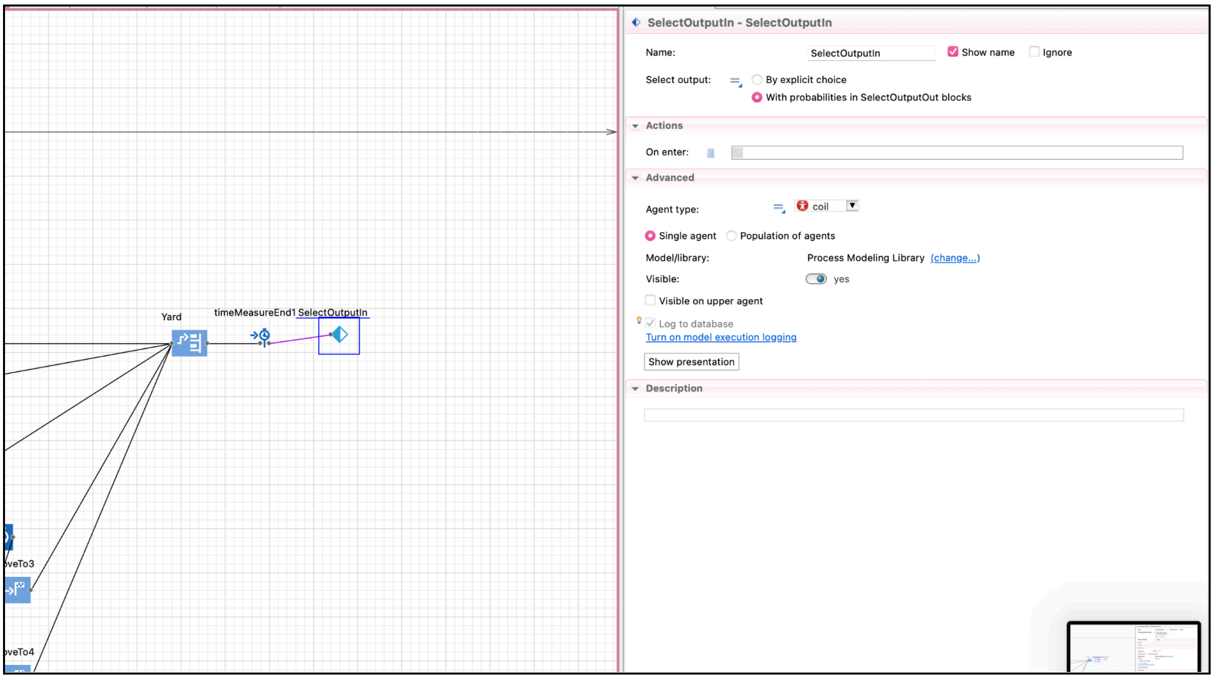
→ For this part of the setup, we introduced a BAiF delay, configured with a delay time using the triangular distribution triangular(42,47,55)triangular(42,47,55) hours. This delay represented a key processing phase, allowing for variability in the operation duration while ensuring realistic timing within the system. The capacity for the BAiF delay was set at 23, accommodating a controlled number of agents at a time to maintain an organized flow through this stage.

→ Additionally, we included specific actions for this component. Upon entering, the onEnter action was implemented, where the agent's shapeBox was set to change its fill color to oliveDrab. This visual indicator provided clarity in tracking the status and movement of agents as they progressed through the delay. To ensure seamless material flow, we used moveTo elements once again, connecting them to the receiving node as the designated endpoint. These configurations collectively enhanced the system's precision and visualization, while maintaining smooth transitions between stages of the manufacturing process.

(Done by Camila)







→ For the finishing steps, we implemented a rackstore and named it "yard" to manage the storage of materials at the end of the process. The rackstore was configured to automatically select the cell closest to the storage area, optimizing space utilization and ensuring efficient storage placement. The elevation time per level was set to 0.1 minutes, representing the time required to move materials vertically between levels within the storage area. The location of the rackstore was placed at the receiving node, ensuring materials were directed appropriately within the network for final processing.

→ To track and measure the processing time, we incorporated the TimeMeasureEnd element. This element was configured with a dataset capacity of 250, which allowed for accurate monitoring and recording of the processing times associated with materials as they passed through the finishing steps. Finally, we concluded the network flow with the selectoutputin, marking the endpoint for material processing and ensuring all materials were appropriately handled, tracked, and directed within the system.