STEEL MANUFACTURING MODEL

Process Modeling and Simulation

Steel Manufacturing
Professor. Dr. Yilmaz Uygun
Group 2: Hala Abuhassan, Alexandra Gkragkopoulou, Joelle Karadsheh, Nada Martinovic
December 3, 2023

Introduction (Alexandra)

The goal of this assignment is to simulate the manufacturing of steel. Steel is mainly used while constructing buildings, infrastructure, tools, ships, automobiles, machines, appliances and weapons. The main steps that are involved in the manufacturing of steel are: melting of scrap, degassing, compact strip production, cold strip milling. Our report consists of a 2D animation, graphs and diagrams for total production, slab processing time, number of slabs and coils produced, and a graph showing different products. We also discuss important parameters of the system, we run an optimization experiment where we change the input rate, a maintenance model and develop a system dynamics model.

Question 1 (Joelle and Nada)

The model below shows the steel manufacturing plant. Steel is manufactured through different processes. The steel plant produces galvanized and galvannealed sheets, cold rolled fully processed sheets, hot rolled plates, and cold rolled full hard sheets).

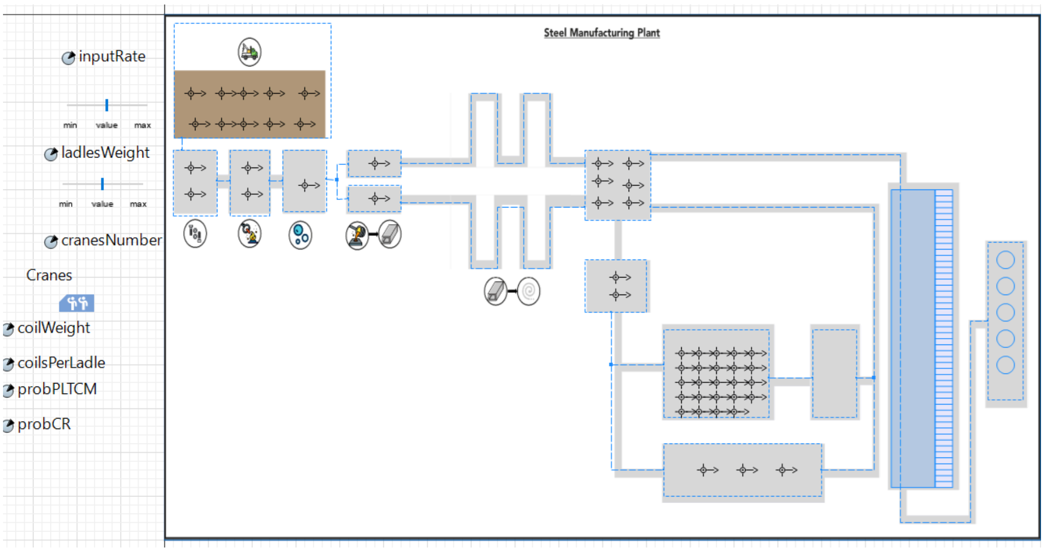


Fig.1 Steel Plant

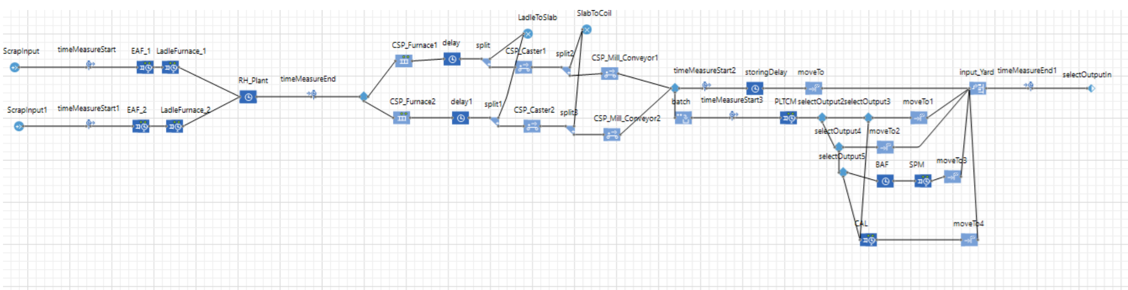


Fig.2 Production flow

Question 2 (Hala)

First graph: Total production

In order to visualize the total production at the steel plant we added a time plot graph and named it"Total Production". We used a time plot specifically to show us the total production over time. It is very important to note that the total production also considers the amount of slab produced in earlier stages as well as coil production, hence our total production consists of (steel slabs, Coil, hot rolled HR, cold rolled hard sheet, GIGA, CRFP, HRP, CRFH).

Total production = amount produced * LadlesWeight

The amount of each product produced is taken from the output. Hence, in the time plot and under "value" we add up all the outputs for all the products and then we multiply them with the ladles weight which is in tons. The graph on the right shows an increasing graph because as more time passes we are producing more products.

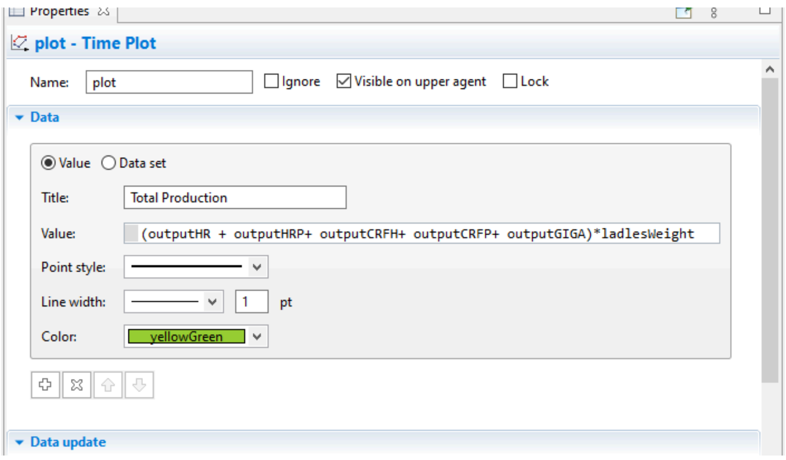


Fig.3 Time plot properties

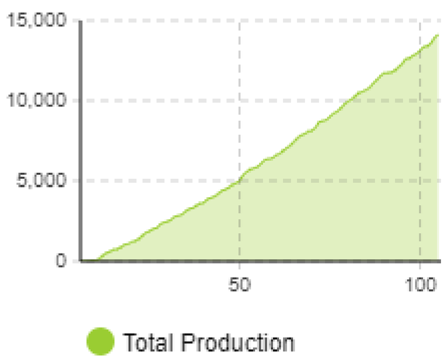
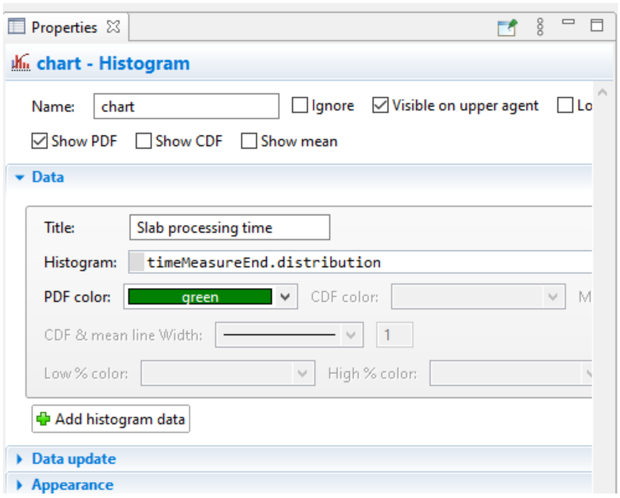


Fig.4 Amount produced in tons over time

Second graph: It is for the slab processing time, for this we need to consider the timeMeasureEnd as it measures the time. We use the histogram graph and change the properties in it. We put "timeMeasureEnd.distribution" and we name the graph "Slab processing time". Finally, we notice that the mean slab processing time is 3.6.



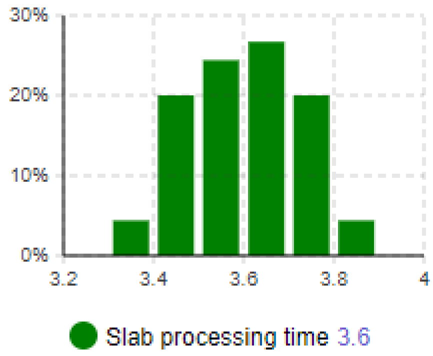
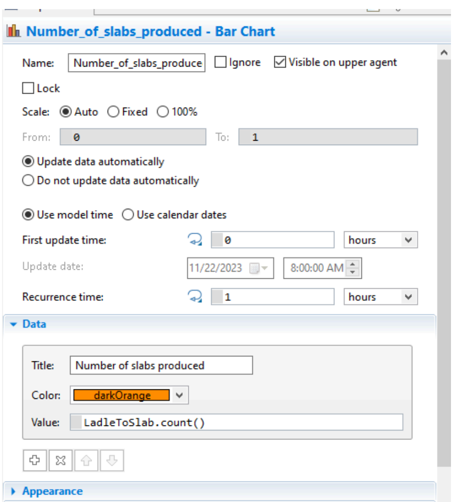


Fig.5 Histogram Properties

Fig.6 Distribution slab processing time

Third graph: Number of slabs produced

First, we create a bar chart and assign it to the name "Number of slabs produced" Then we set the value in the data to "LadleToSlab,count()" This will count the slabs at the sink because that is the total amount of steel slabs produced. The number of steel slabs produced is 15,373 and that is the amount reached when the max number of agents is reached.



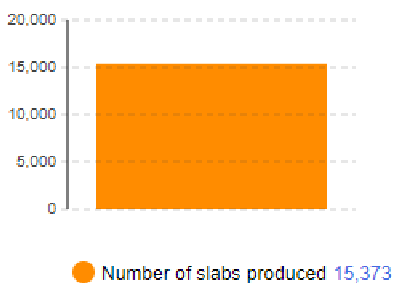


Fig .7 Bar chart properties

Fig. 8 Total amount of slabs produced

Fourth graph: Number of coils produced

we add another bar chart. This time we want to count the number of coils produced. Hence, we consider the sink "SlabToCoil" where the total amount of coil produced is collected. We use a count function to measure the total number of coils produced in this steel plant. We notice that the total amount is 15,364.

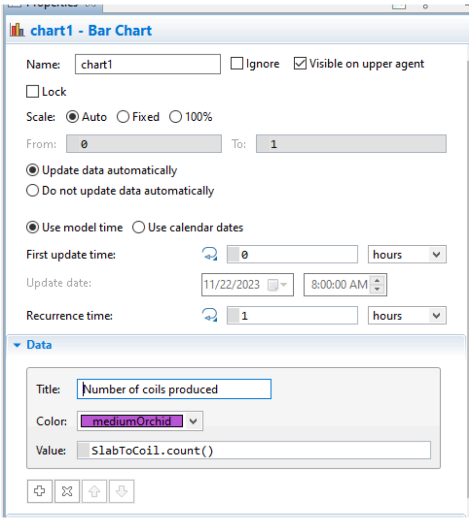


Fig.9 Bar chart properties

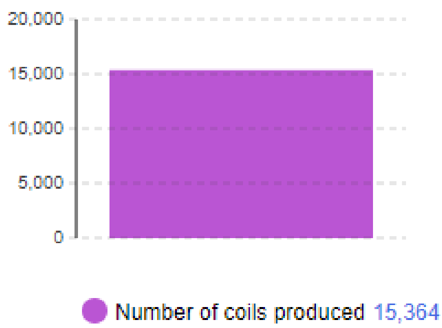
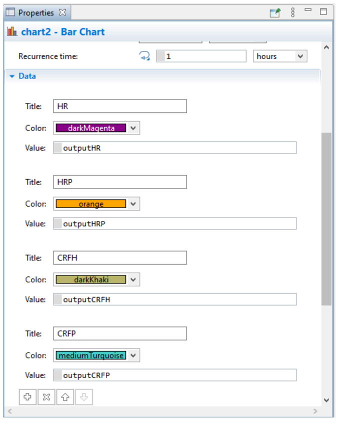


Fig.10 Bar Chart for total amount of coils produced

Fifth graph: Variations of products at the plant

The given flow chart in the assignment shows the total amount of products produced at the steel plant. However, it is still better to visualize them. Therefore, we added a bar chart and we added 5 products (HR,HRP,CRFH,CRFP,GIGA). Each has a different color to help distinguish between them, the values are different according to the product produced. Notice how at the beginning of the simulation, the amount of products produced is 0 since there is a process that needs to happen before it starts producing the products, but as they move from the conveyor, the products will start to show in the graph and the graph displays the following numbers.



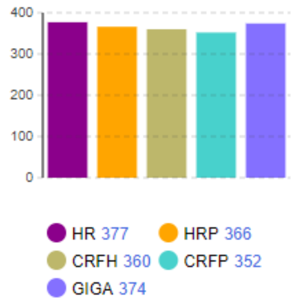


Fig.11 Bar Chart Properties **Fig.12 Different products at the plant**

Question 3 (Joelle)

To show the changes in different stages and highlight the different results from various processes, we included two statecharts in this model: one for the ladle and another for the coil.

1. **Ladle state chart**

To enhance the ladle diagram in our model, we introduced an oval shape at the top. This oval alternates between the colors blue, cyan, dodger blue and yellow Green, representing different phases the ladle undergoes. Our statechart outlines four main states: Delivery, Scrap, Molten Steel, and Post-Degassing, with transitions between these states initiated by specific messages.

The statechart starts in the Delivery state, moving to Scrap upon receiving the "delivering" message. This transition is programmed in the EAF_1 and EAF_2 blocks, where the function `send("delivering", agent)` is activated in the On Enter section. Similarly, the Molten Steel state is initiated by the "melting" message from the LadleFurnace_1 and LadleFurnace_2 blocks.

The final phase, Post-Degassing, is reached from the Molten Steel state through a transition triggered by the "degassing" message. This message is sent by the RH_Plant block using the function `send("degassing", agent)` in its On Enter section.

To visually track these transitions, we implemented the `oval.setFillColor(Color)` function in the Entry and Exit actions of each state. This ensures the oval changes color corresponding to each state transition, vividly illustrating the progress through each step of the process.

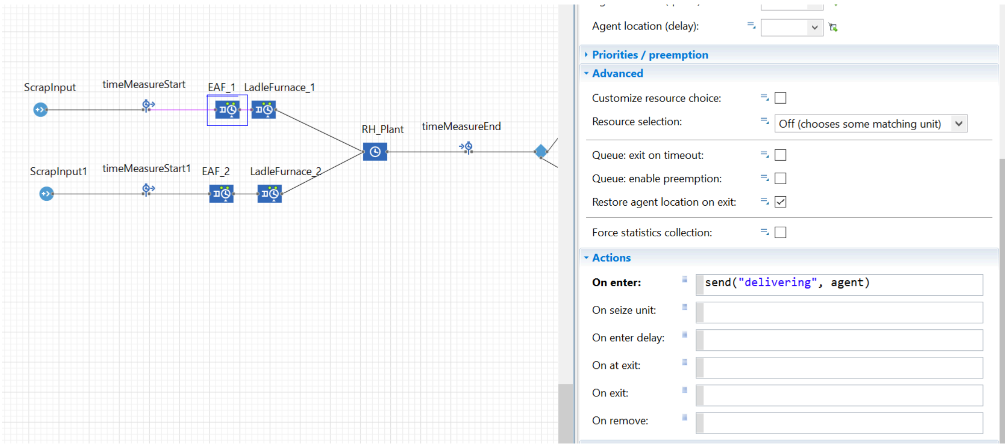


Fig.13 EAF_1 and EAF_2 actions

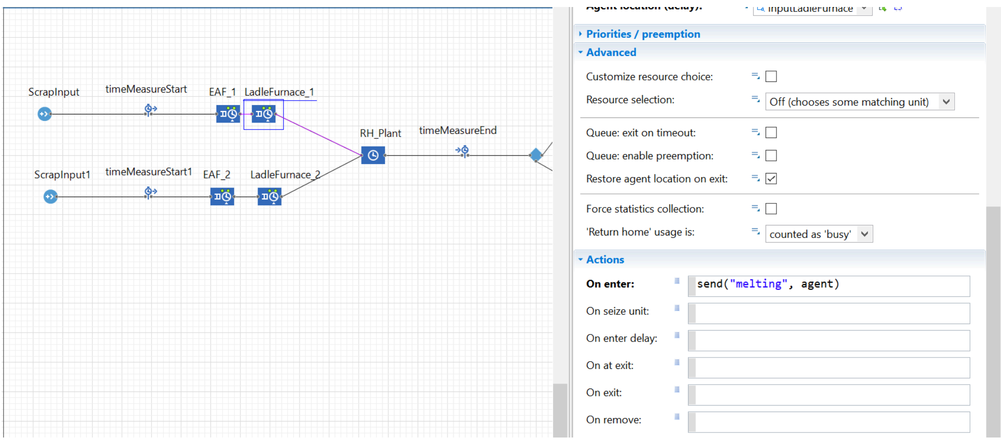


Fig.14 LadleFurnace_1 action

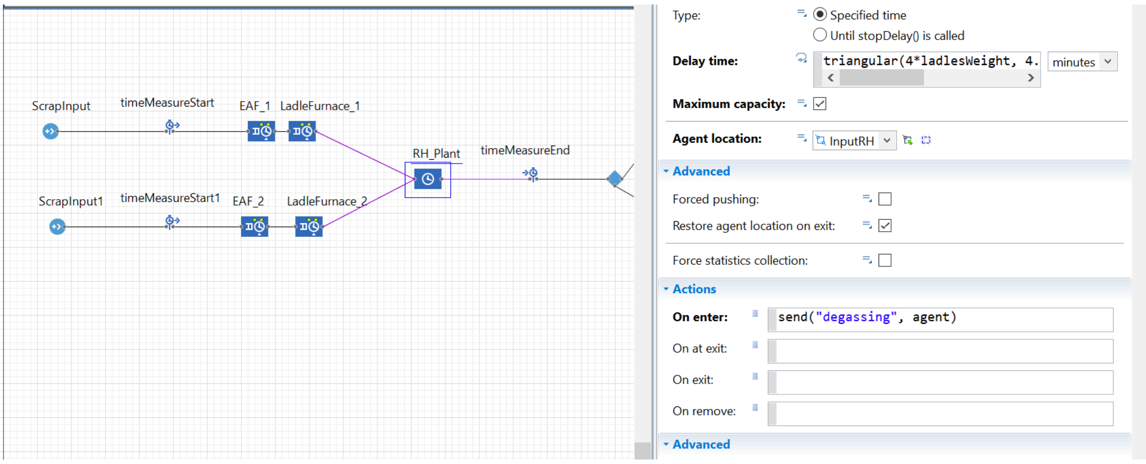


Fig.15 RH_Plant action

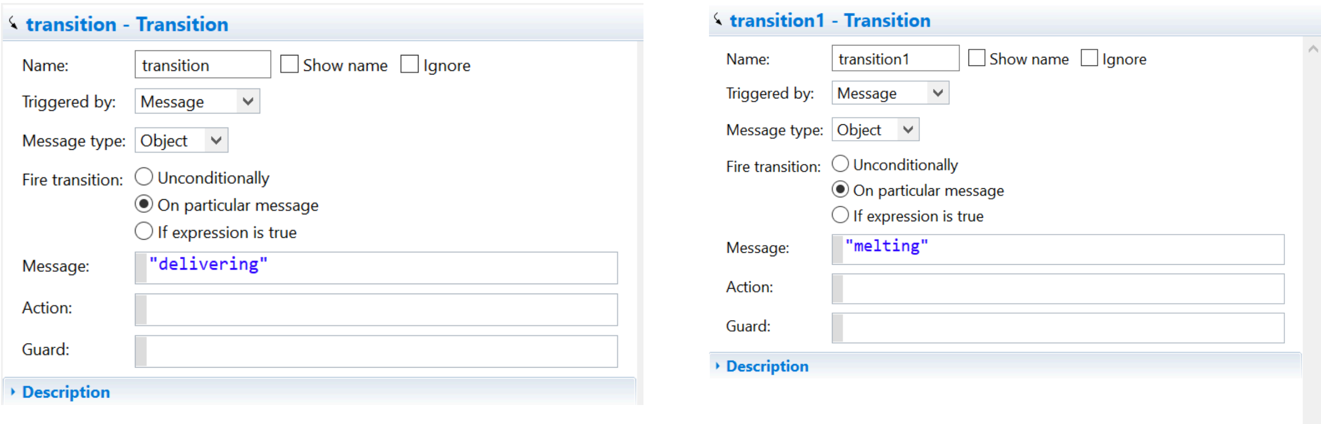


Fig 16+17 Transition

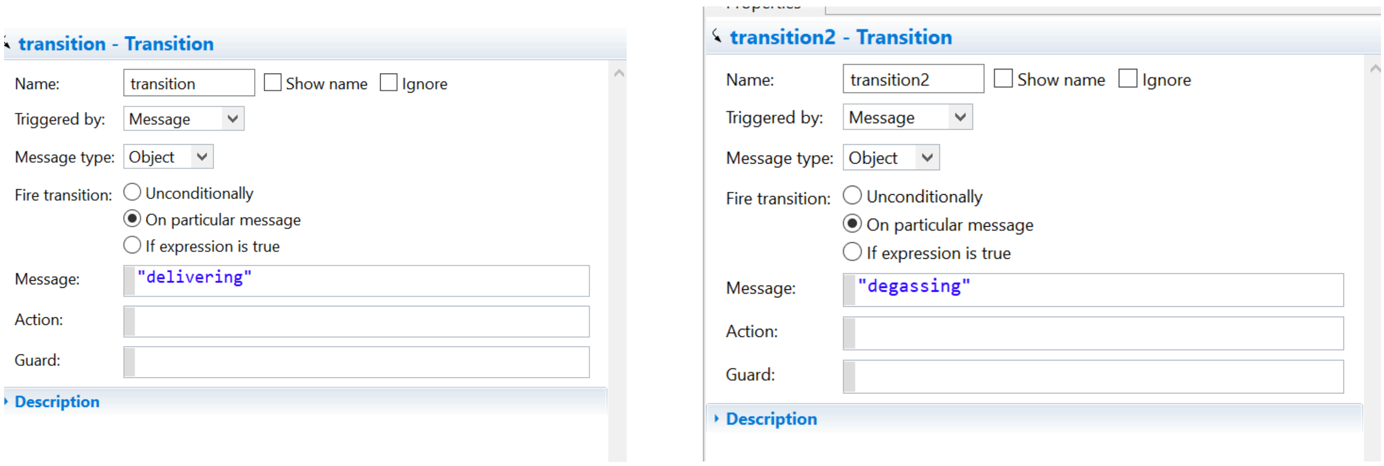


Fig 17+18 Transitions

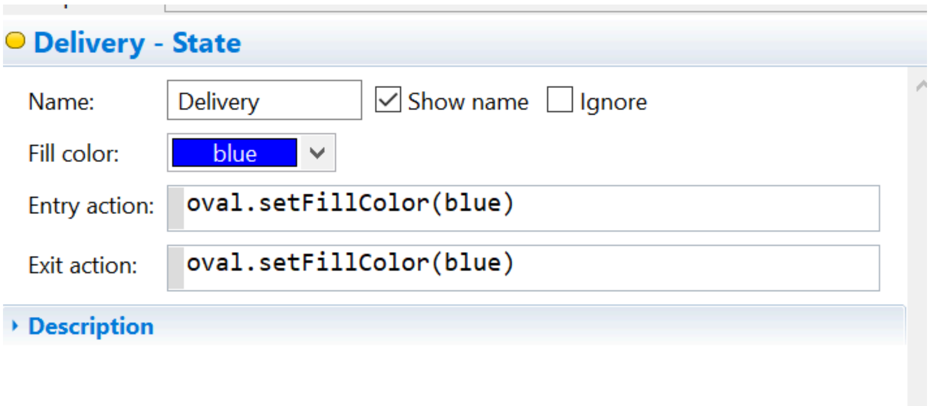


Fig.19 Example of State

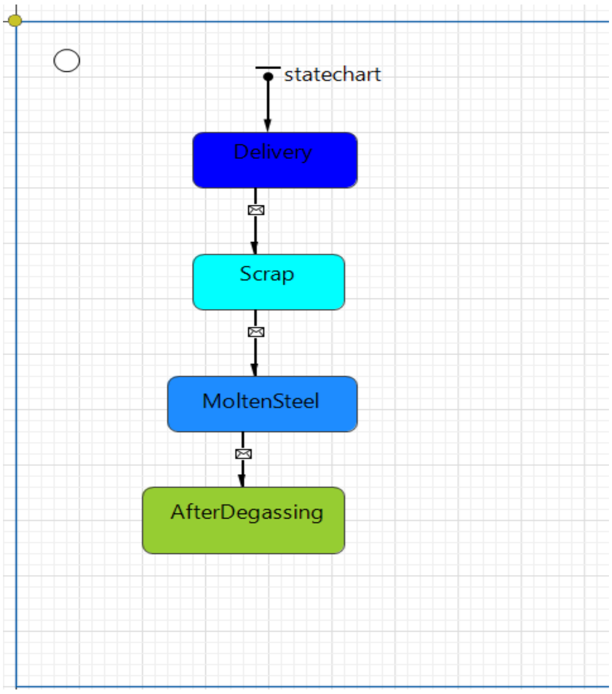


Fig.20 state chart

2. Coil state chart

Here, we are adding colors to the 3D boxes in our model using a statechart with 6 states and 5 transitions. These transitions are activated by 'String' type messages, set up so that the Fire transition responds to specific messages. The statechart begins at the "Coils" state and includes two transitions activated by messages, leading to either the HR or HRP states. The transition between "Coils" and "HR" is a message and in particular message "HR", As for the transition between "Coils" and "HRP" it is also a message and in particular message "HRP". As for the main chart I put "send("HR", agent)" on enter action in storingDelay. We added "send("HRP", agent)" on enter action in PLTCM. From HRP, transitions lead to either GIGA or CRFH states. For the transition between HRP and GIGA it is a message with one particular message "GIGA" and the same logic for the transition between HRP and CRFH

with the message being “CRFH”. As for the main chart I put “send("GIGA", agent)” on enter action in CAL and I added “send("CRFH", agent)” on enter action in BAF. The final transition moves from CRFH to CRFP with a message transition of “CRFP” and “send(“CRFP”, agent)” as an on enter action in SPM.
The statechart uses colors like powderBlue,mediumPurple,orchid,salmon,lightGrey and yellowGreen. We apply these colors using the `shapeBox.setFillColor(Color)` function in each state's Entry action. This setup ensures that the box colors change dynamically, reflecting the current process stage of the steel and differentiating various outputs by color.

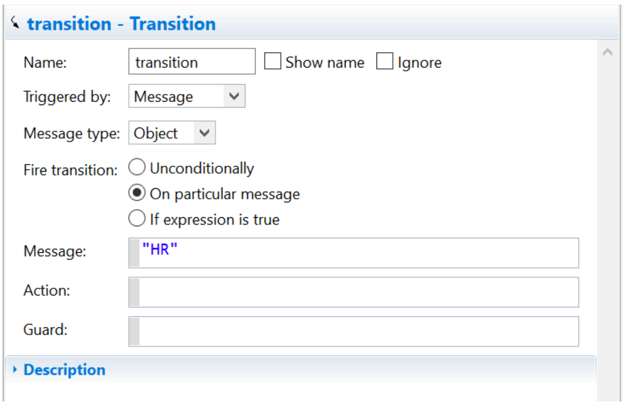
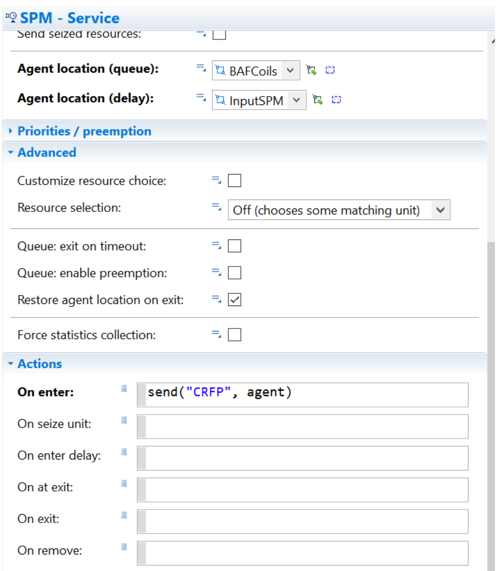
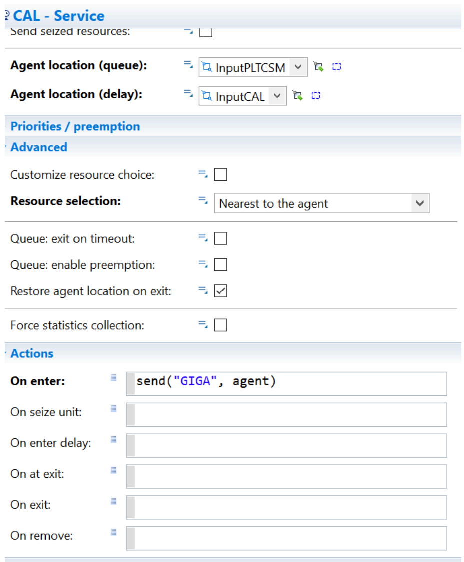
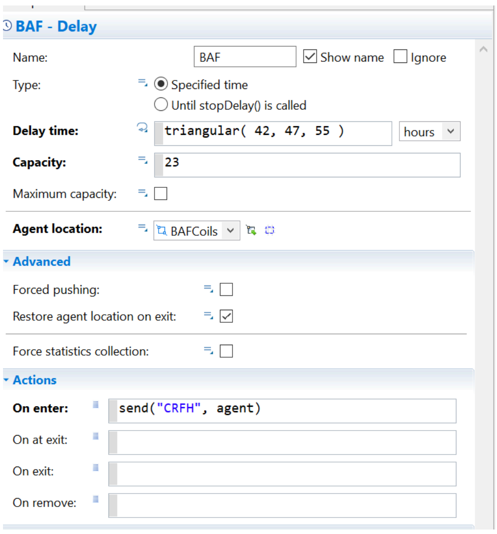
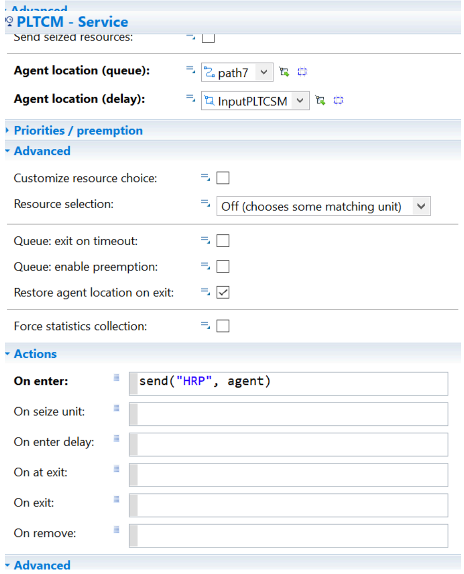


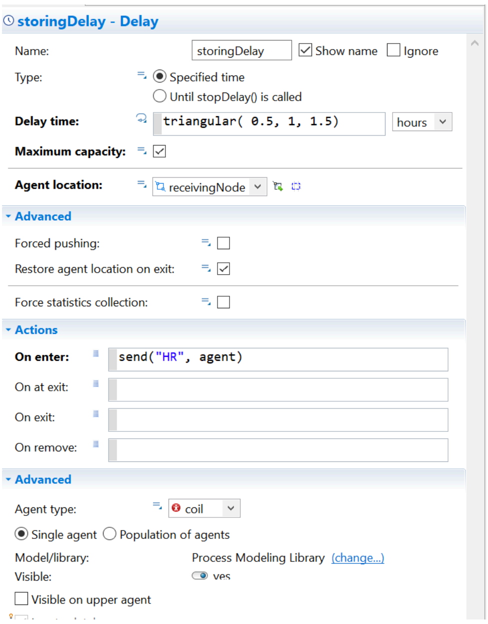
Fig.21 Transition example











Figs.22 till Figs 26 are all on enter actions

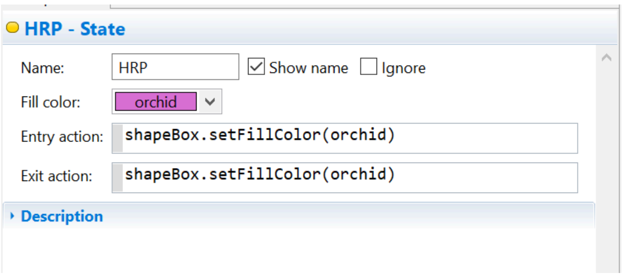


Fig.27 HRP state example

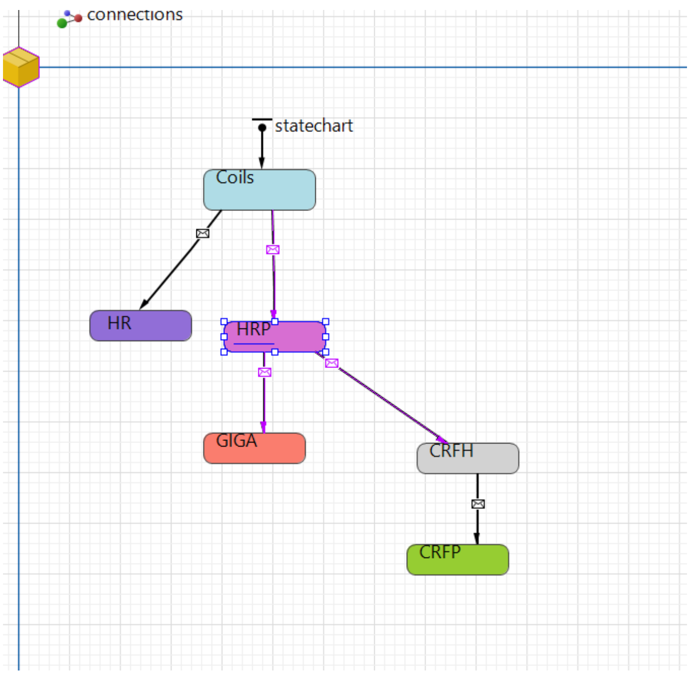


Fig.28 State charts

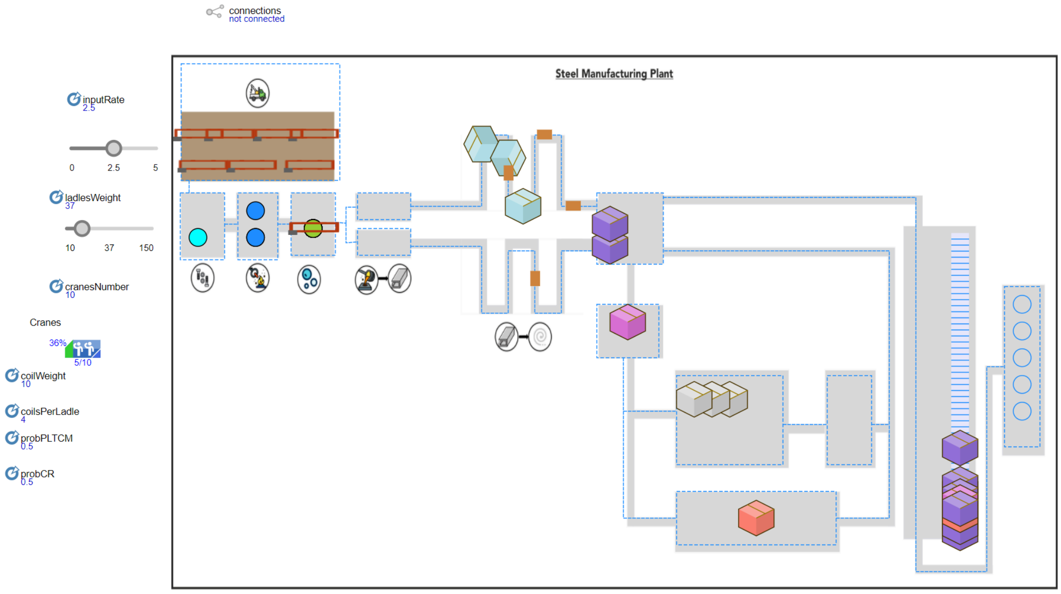


Fig.29 Simulation running

Question 4 (Nada)

There are seven essential parameters in our model. They are all represented in the following pictures.

1. inputRate - with a default value 2.5 and type double. It controls the rate at which the scrap metal is entering the model and is directly connected to the production rate.

2. cranesNumber- represents the total number of cranes in the model, with the value of 10

3. ladleWeight- represents the ladle's weight. The default value is 40.

4. coilWeight- weight of coil has a default value of 10.

5. coilsPerLadle - the amount of coils made per one ladle. The type is double and is connected to ladlesWeight / coilWeight.

6. probPLTCM - represents the probability of going to the production of hot rolled steel coils and the rest goes to cold mill. It is also type double and has 0.5 as a default value.

7. probCR-by default it is 0.5 probability and determines the specific selectOutput which shows which output is going to be produced.

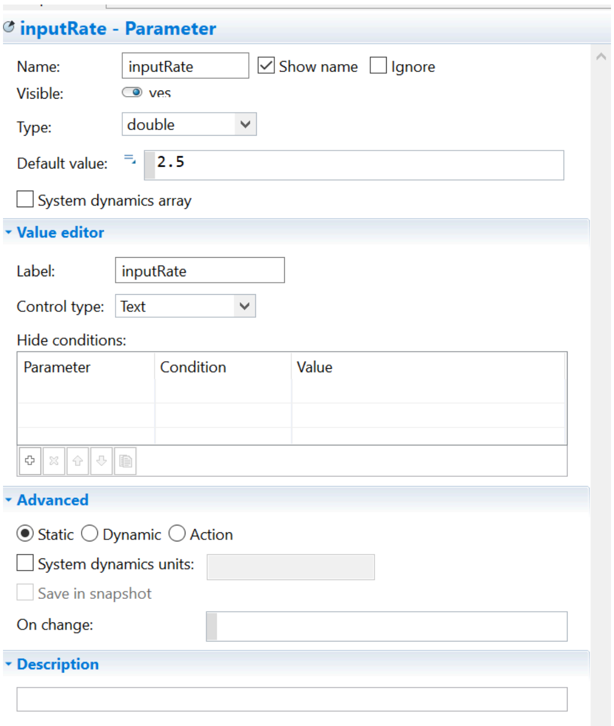


Fig.30 Input Rate

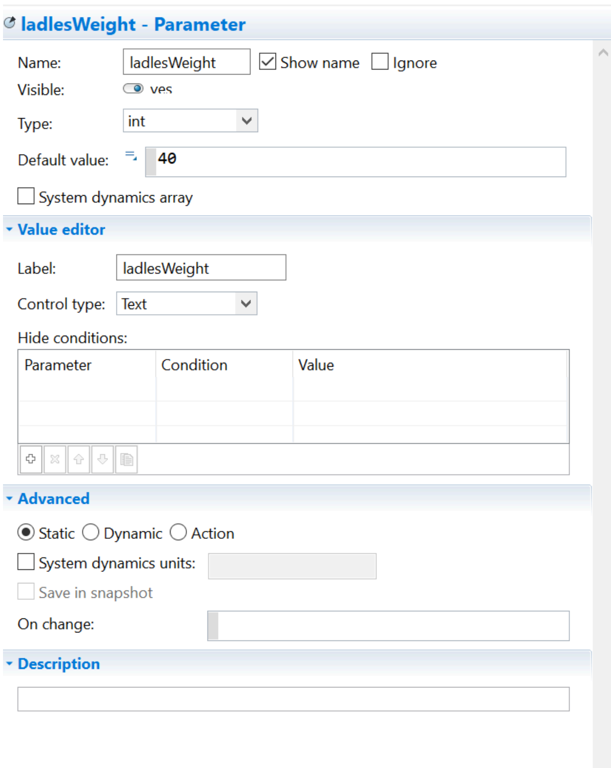


Fig.31 Ladles Weight

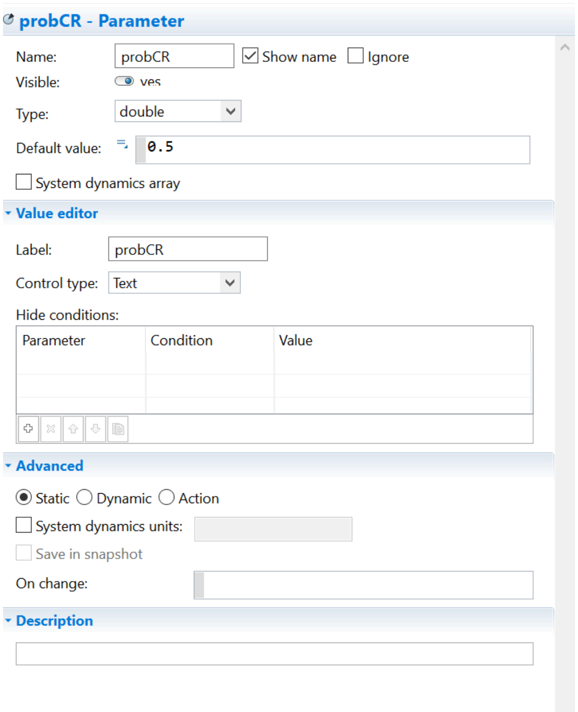


Fig.32 ProbCR

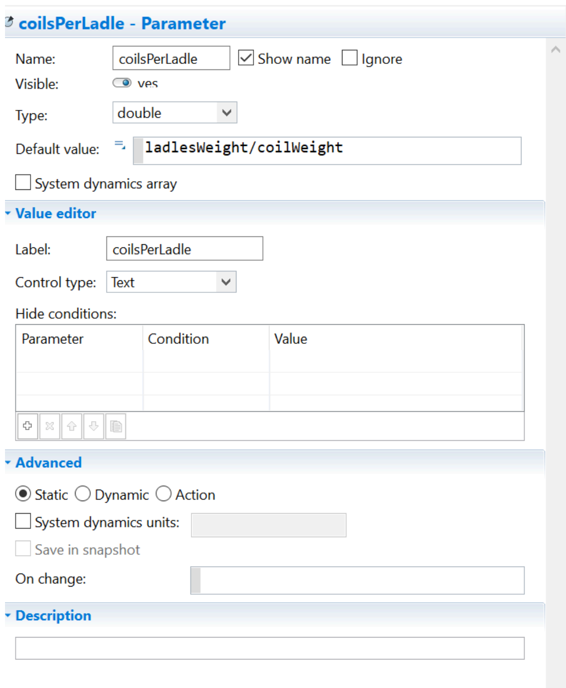


Fig.33 Coils Per Ladle

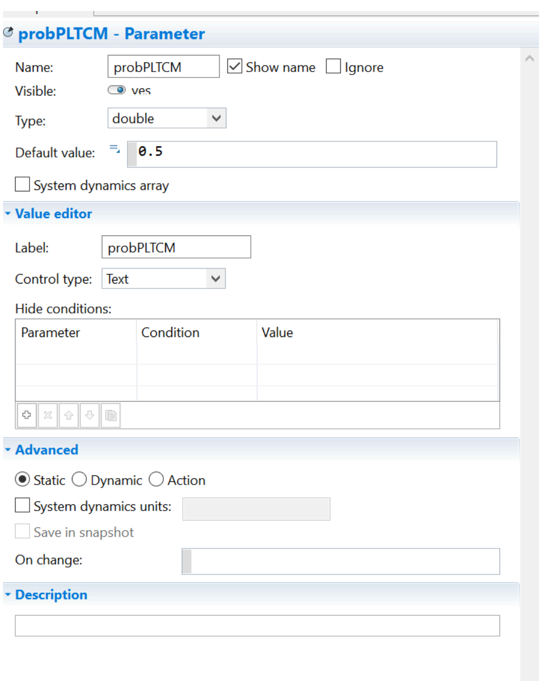


Fig.34 probPLTCM

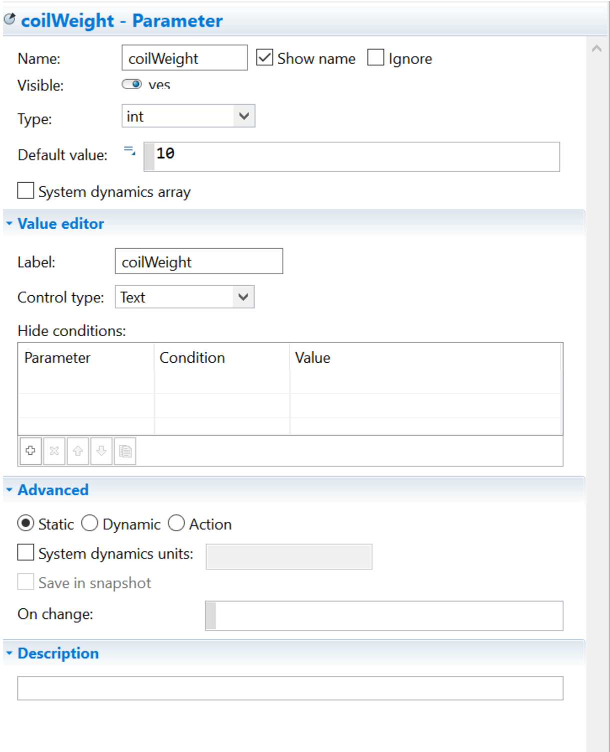


Fig.35 Coil Weight

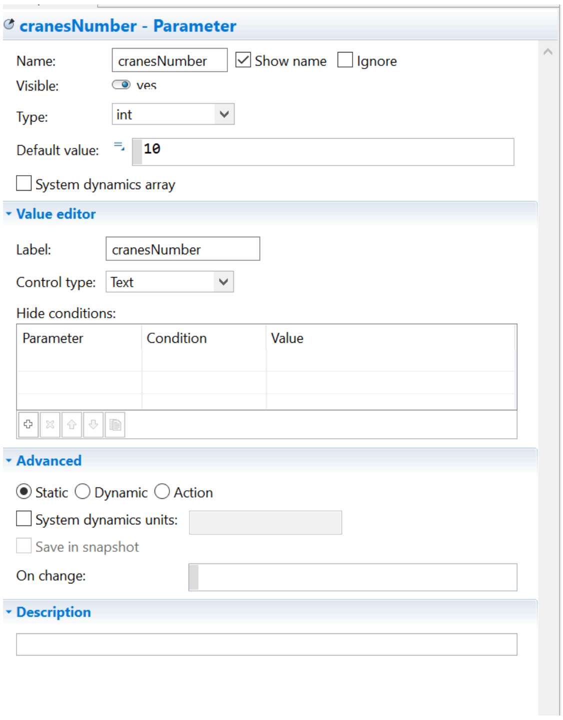
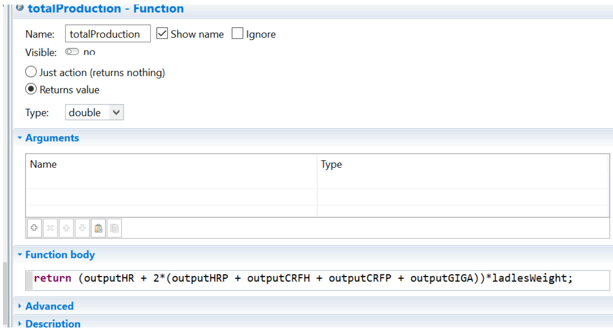


Fig.36 Cranes Number

Question 5 (Alexandra)

For this task, we will try to make an optimization of our model to increase the production rate by changing the input rate. We create a new model- optimization experiment and set the number of iterations to 500. In the main, we create a new variable 'totalProduction' and in the function body we enter: <u>return (outputHR + 2*(outputHRP + outputCRFH + outputCRFP + outputGIGA))*ladlesWeight;</u> Then, in the optimization we set the objective function to <u>root.totalProduction()</u> and select maximize.

Fig.37 totalProduction Function



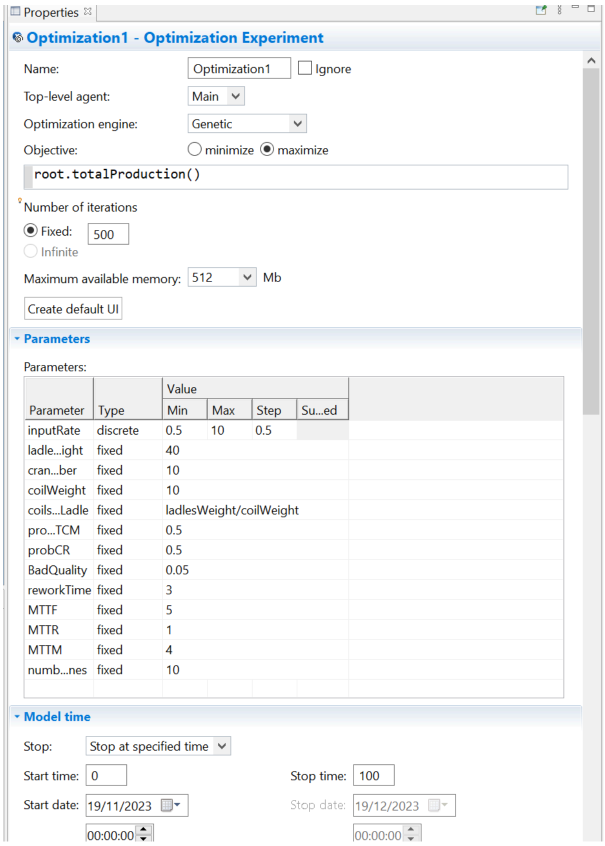


Fig.38 Optimization Parameters

Running the model, we observe that with the best input rate equal to 7, we get a maximum total production of steel equal to 56,480 with 17 iterations.

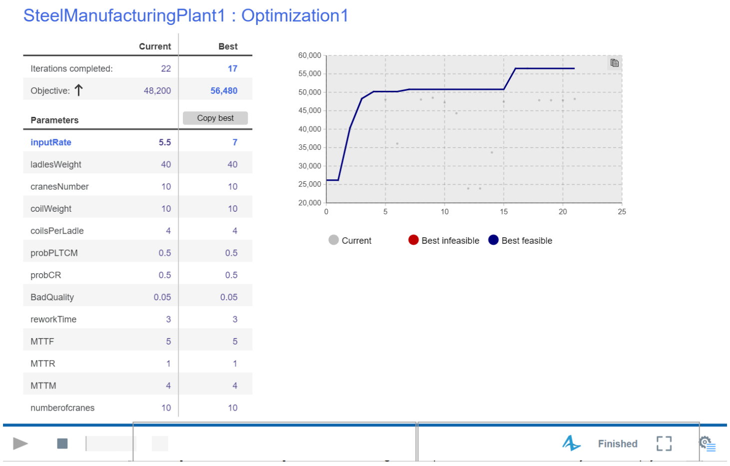


Fig.39 Optimization Result Model

Question 6 (Nada+Alexandra)

We need to simulate a breakdown and maintenance system of only 2 cranes. A breakdown of the crane may happen every 5 days on average and there is maintenance happening regularly every 4 days. We added three states, InOperation, maintenance and failure. We also added the parameters MTTR, MTTM, MTTF and numberofcranes as shown in the figures below. From InOperation to maintenance a transition was added with a rate of 1/MTTM per day and an action of Cranes.set_capacity(2). From maintenance to InOperation it was a rate of 1/MTTR per day. From inOperation to failure a transition with a rate of 1/MTTF per day was added with an action of Cranes.set_capacity(2) and lastly the transition from failure to inoperation it was a rate of 1/MTTR per day. As for the states we put the capacity of In Operation as the total number of cranes by setting an action of Cranes.set_capacity(numberofcranes).

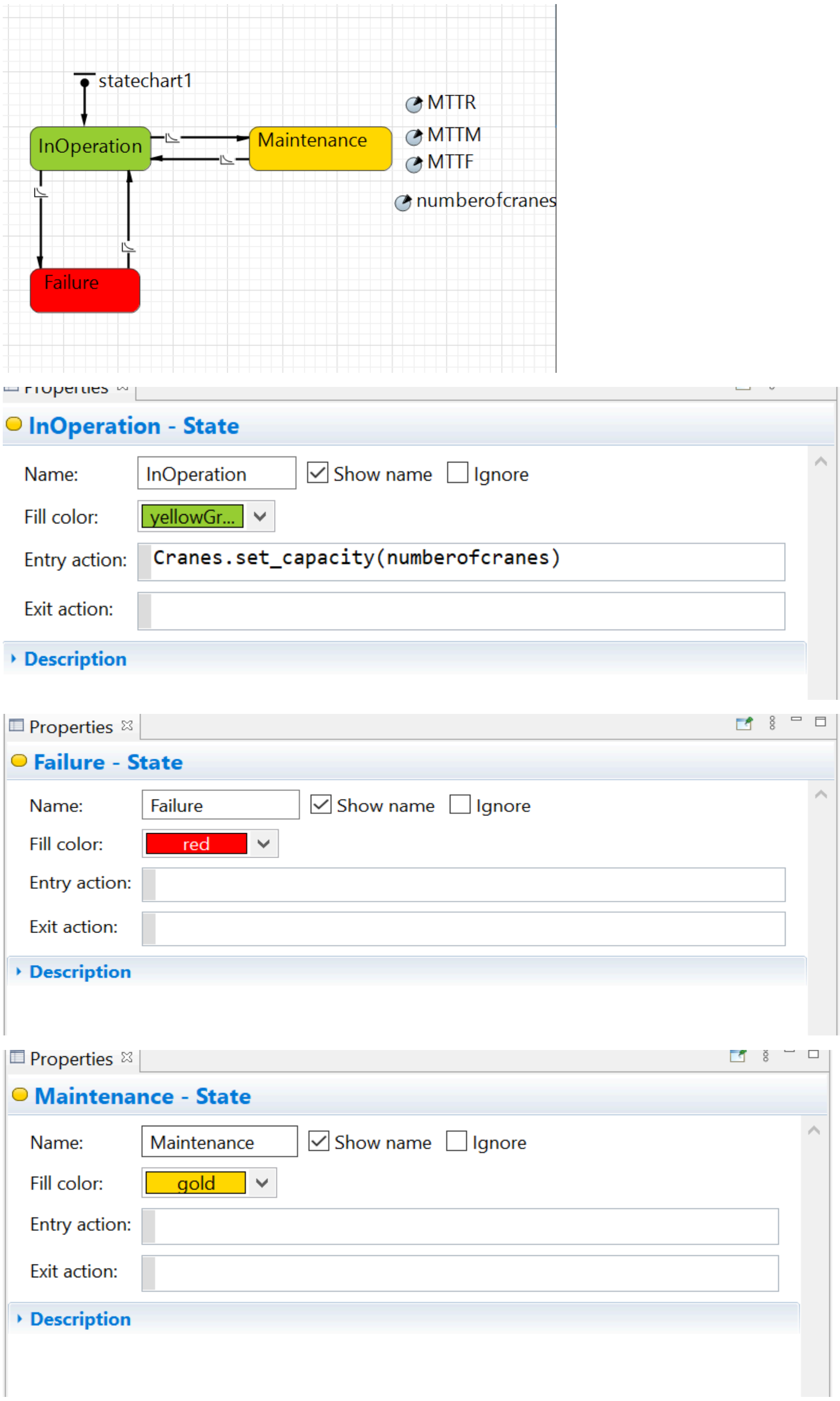


Fig 40 to Fig 43 State chart representing the maintenance system and each of the states with their specific colors

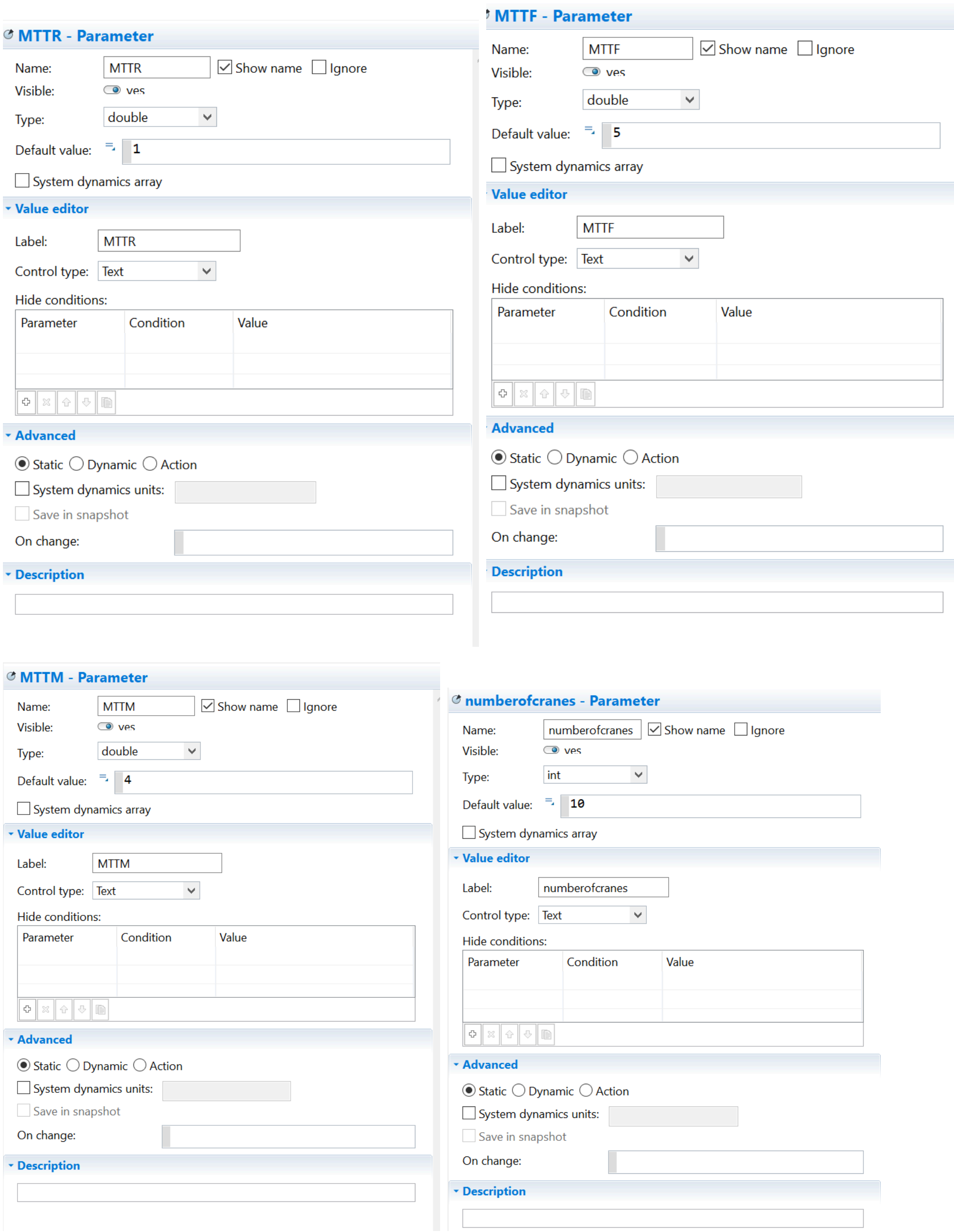


Fig 44 to Fig 47 Parameters

Fig 48 (Transition 3)-From InOperation to Maintenance:

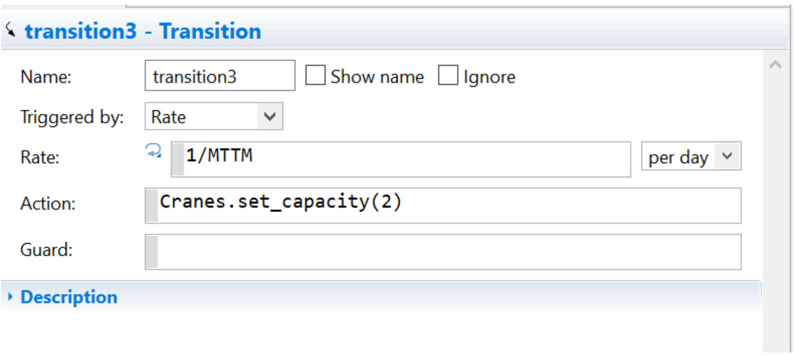


Fig 49 (Transition 4)- From Maintenance to InOperation

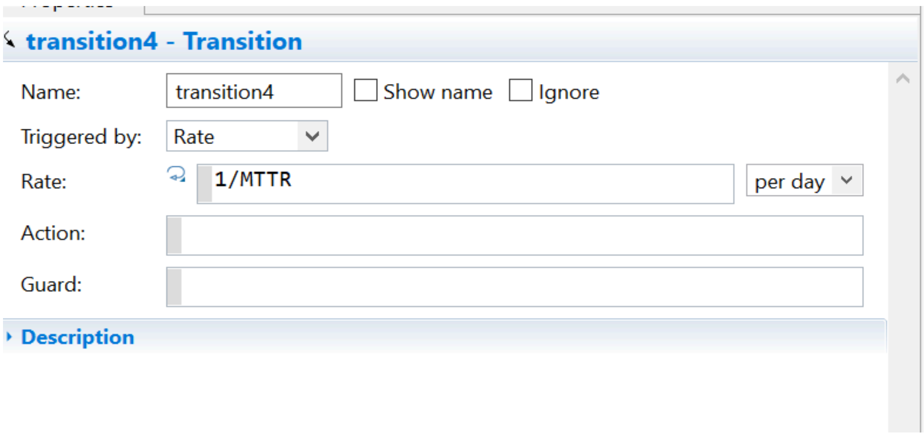


Fig 50 (Transition 2) from InOperation to Failure:

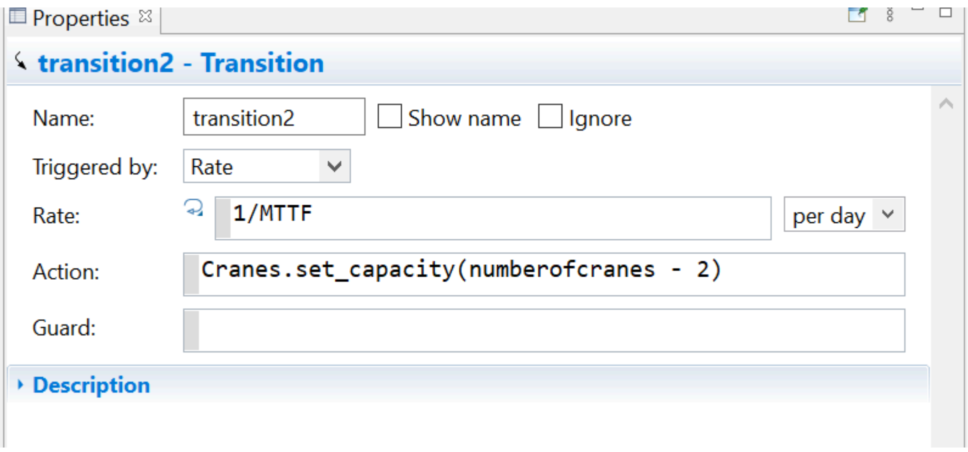
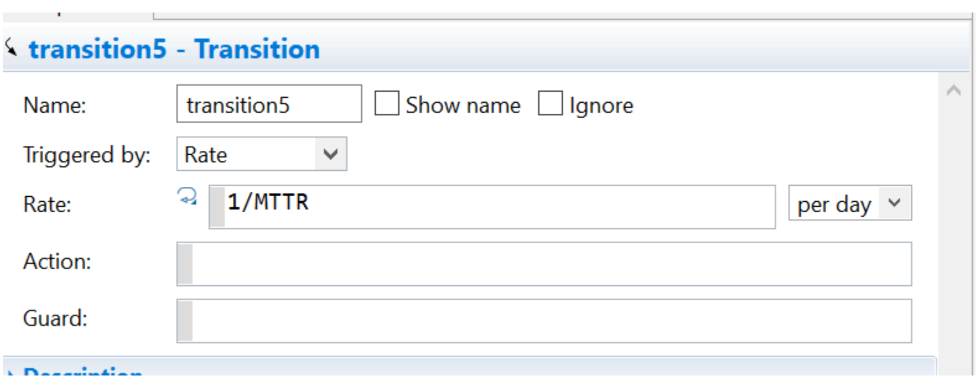


Fig 51 (Transition 5) From Failure to InOperation



Question 7 (Hala)

For this question, we need two flows, one for the quality check and the other for reworking. The total amount of slabs are entered into the flow, we use a variable called “Total Slabs” and give it a function “LadleToSlab.count()”, because we want it to count all slabs produced in the steel plant. Only 5% of the total amount of slabs are counted as bad, therefore we set a parameter called “BadQuality” and we set it to be double, and 0.05. Quality check is the inflow of “total slabs* bad quality”.

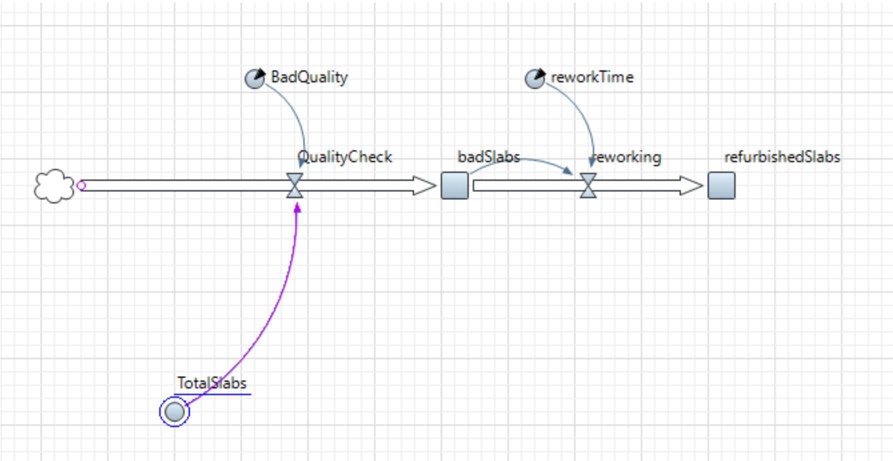


Fig 52 System dynamics model

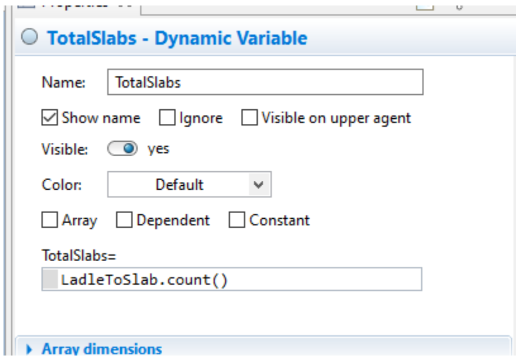
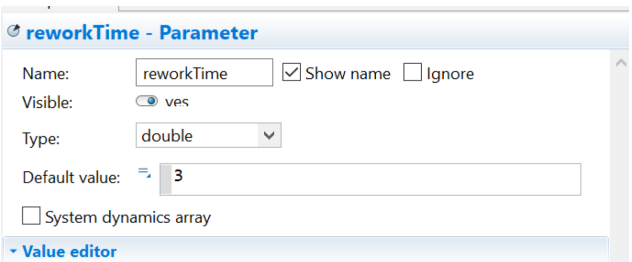


Fig 53 Varibale counting slabs



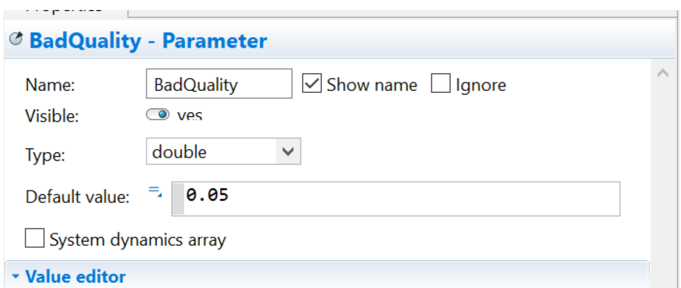


Fig 54 - 55 Fixed Parameters for rework and bad quality

The stock for badSlabs is 0 at the beginning since no bad slabs are entered but as the simulation starts, the stock will start to get filled. The bad slabs are checked and then will go to rework, rework takes 3 hours, so the parameter is set to 3 and our model time units are in hours. Reworking is the inflow of only bad slabs and the time it takes 3 hours to fix them, hence reworking is calculated as "bad slabs/ reworkTime".

All parameters and the variable were joined by links to the flow.

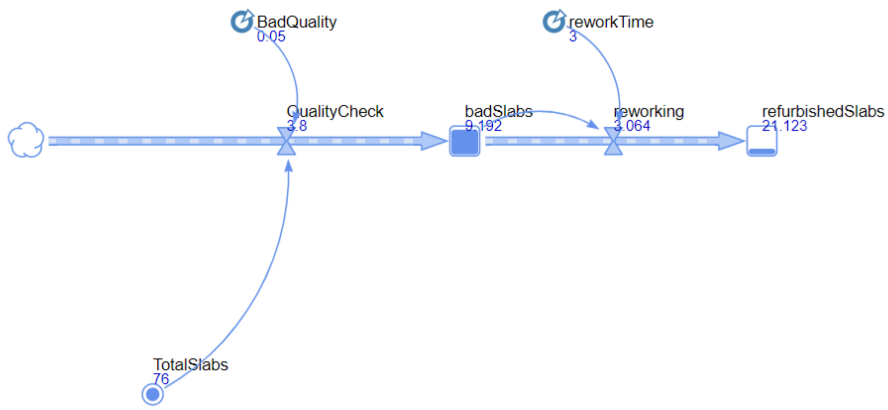


Fig 56
Bonus question (Hala+Joelle):

To extend our model with more parameters and feedback loops, we decided to turn slabs to steel plates. We will only need good slabs for that, therefore we will take 90% of the good slabs produced and we will take 35% of the number of refurbished slabs produced and put them in our stock. Only 50% of the stock will be used to produce the plates.

We add a flow connecting before quality check, it will take the total amount of slabs produced and then moves through a flow that takes the 90% as good slabs produced, keeps them in inventory which is the stock here, and then we added another flow after refurbished slabs, the flow will take only 35% of refurbished slabs, the parameter of "RefurbishedToStock" is set to 0.35. The amount is also added to the stock.
Finally we added a final flow from the stock to produce the steel plates, we are not using our whole stock to produce the steel plates but rather only 50% of it, hence we set the parameter "GoodToPlates" to be 0.50.

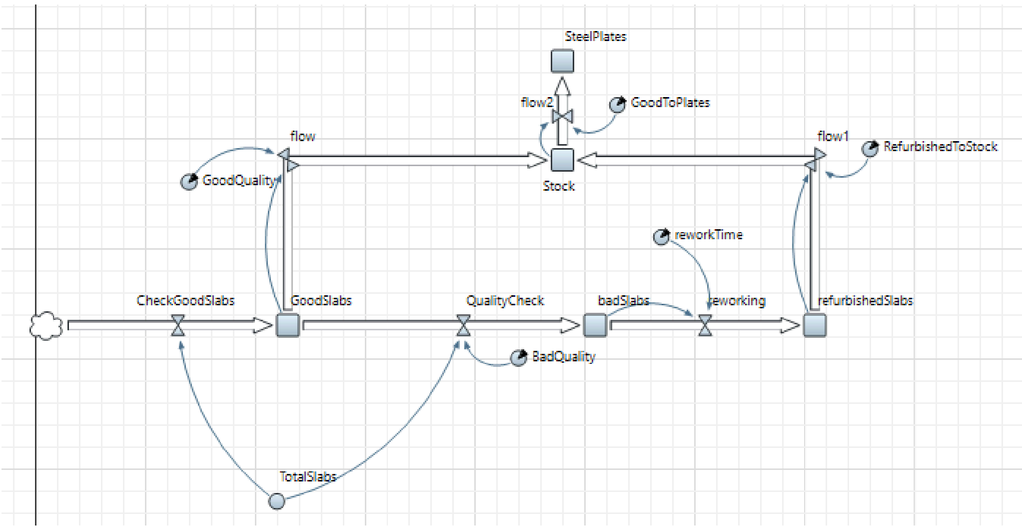


Fig. 57 Extended Model

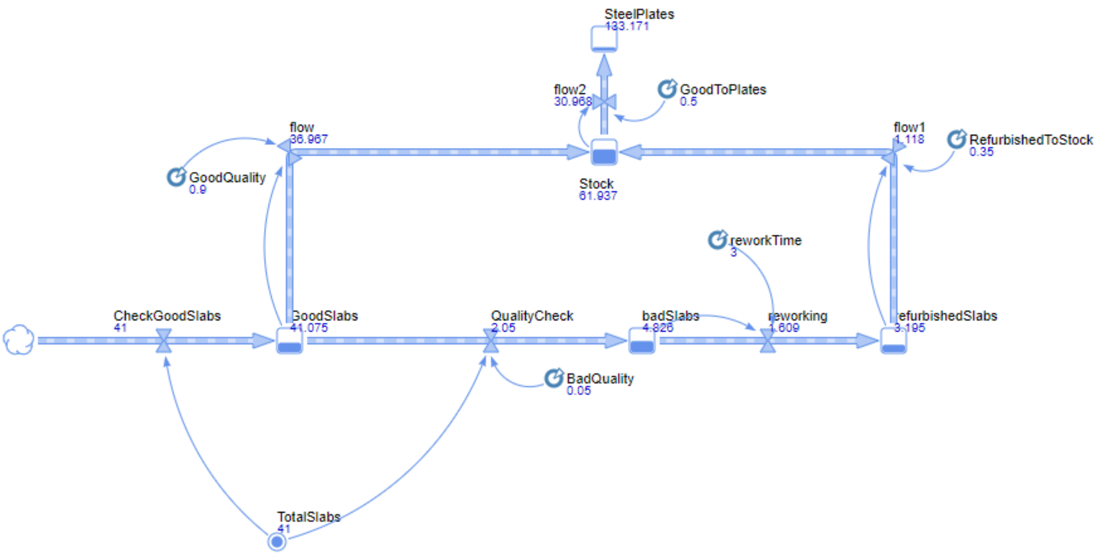


Fig. 58 Simulation of Extended Model

The simulation shows that only 90% of total slabs are considered to be good and will be moved to the stock, the amount of refurbished slabs that have finished with rework and will go to the stock is only 35%. Finally, the model shows that only 50% of the stock will be made into steel plates.

All necessary parameters are equations for the flow of the extended model are provided below

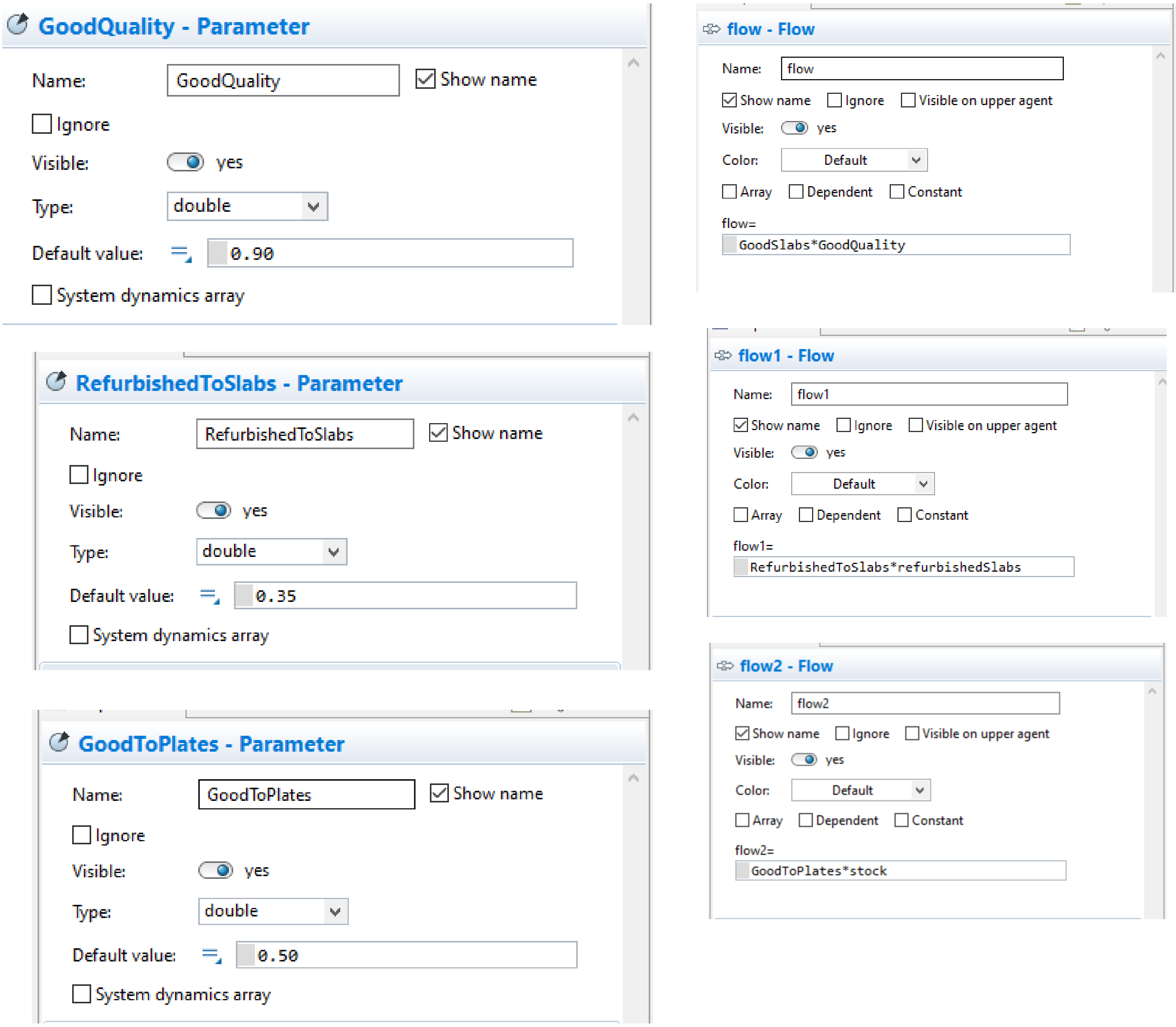


Fig 59-64 parameters and flows added to the extended model

Conclusion (Nada)

In this assignment, we combined all of the previously learned skills in AnyLogic software, from building the model to making state charts. It represents the whole process of steel manufacturing and simulates the organizational process of a steel factory, with all of its main parameters.