

## Assignment 4: Definition of Supply Table and Production Table for the production of different batches of products in the same section

Álvaro Morales Sánchez – 18240

### 1. VSM

We added the following VSM as a scheme of the production model.

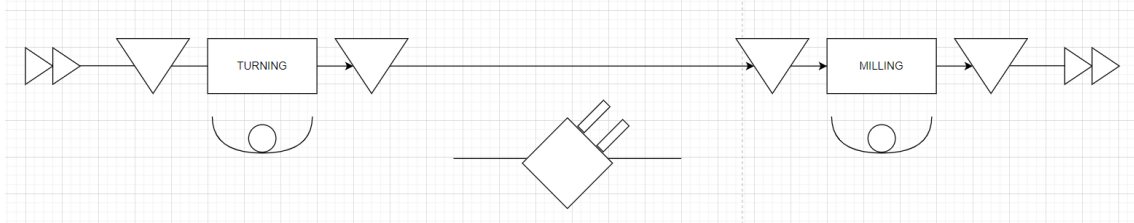


Figure 1. VSM of the model.

### 2. FXS Model

Figure of the structure of the model.

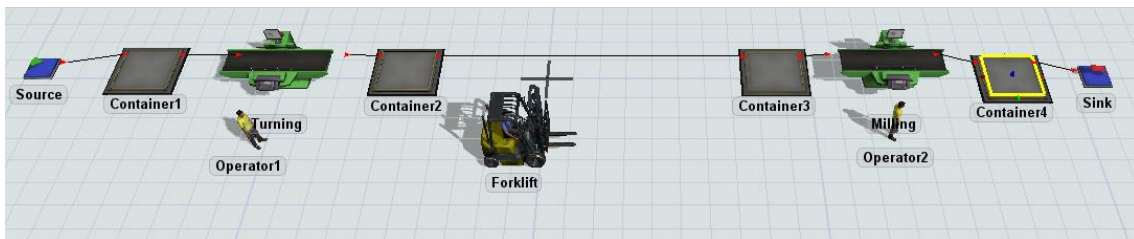


Figure 2. Model in FXS

### 3. Source

We need to create an arrival sequence that achieve the table of the instructions.

	Batch	Setup-Turning	Turning	Setup-Milling	Milling
1	7.00	15.00	7.00	12.00	8.00
2	4.00	10.00	8.00	8.00	6.00
3	10.00	8.00	5.00	6.00	9.00
4	8.00	10.00	10.00	10.00	4.00

Figure 3. Production table of the instructions

Arrivals - Source

Arrivals	4	Labels	1
	ItemName	Quantity	Type
Arrival1	Product	7	1
Arrival2	Product	4	2
Arrival3	Product	10	3
Arrival4	Product	8	4

Figure 4. FXS Arrival Sequence

Also, we create a trigger on creation>set color by case for the different four products.

#### 4. Global Table

We need to create a global table to define the different times of the different operations and other specifications, so that we can read on the different parts of the production line and work as the instructions describe.

	Item	Batch Size	SetUp-Turnin	Turning	SetUp-Milling	Milling
Row 1	1	7	15	7	12	8
Row 2	2	4	10	8	8	6
Row 3	3	10	8	5	6	9
Row 4	4	8	10	10	10	4

Figure 5. Global Table with the production details

#### 5. Turning and milling times

In order to choose the different set up times and operation times, both operations need to read the data from the GlobalTable created before with the get function. In the following pictures it is defined and specified the use of the function.

For the setup time we have to define the batch processing in the same way.

The screenshot shows the 'Processor' configuration window. The 'Max Content' is set to 1, and 'Animate Items' is checked. Under 'Setup Time', 'Batch Processing' is selected. The 'Process Time' field contains the formula 'balTable",getitemtype(item),4)' with a unit of 's'. 'Use Operator(s)' is checked and set to 1. The 'Operator' field contains 'current.centerObjects[1]'. 'Priority' is 0.00 and 'Preemption' is set to 'no preempt'.

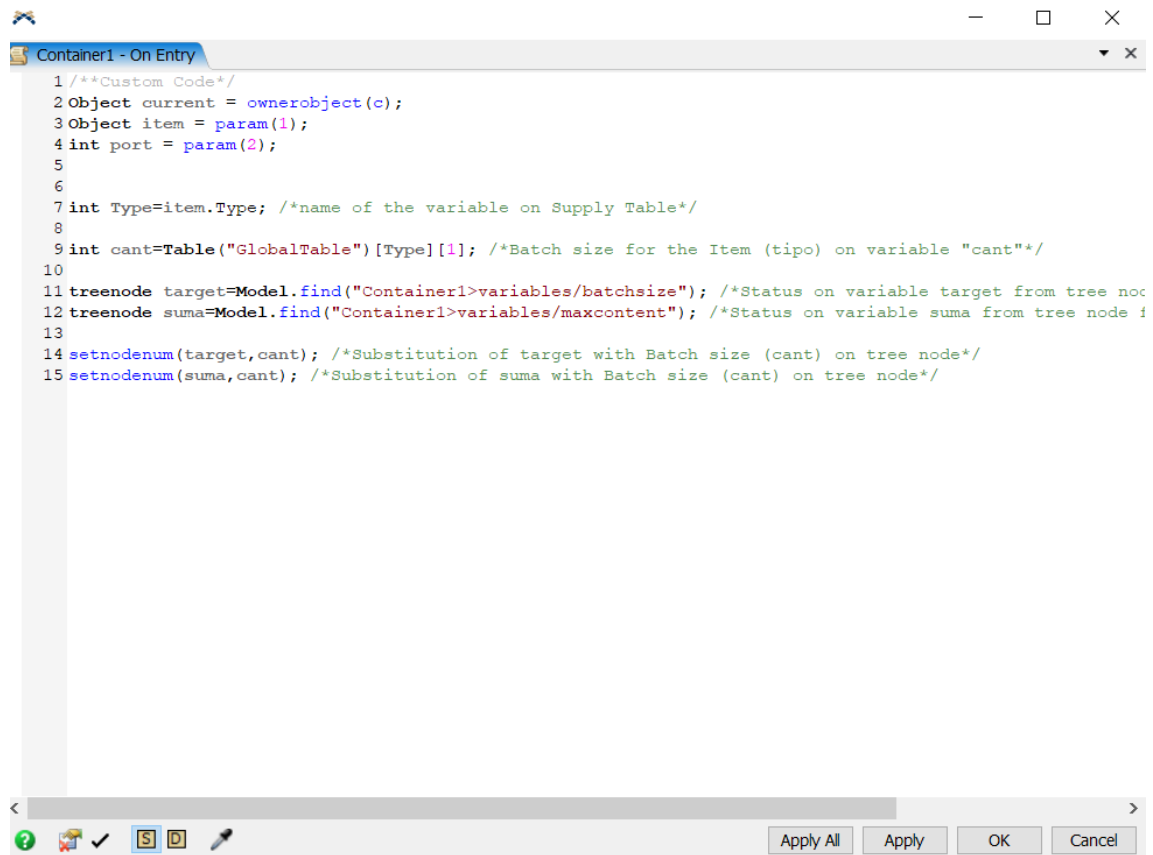
Figure 6. Turning times.

The screenshot shows the 'Processor' configuration window. The 'Max Content' is set to 1, and 'Animate Items' is checked. Under 'Setup Time', 'Batch Processing' is selected. The 'Process Time' field contains the formula 'um("GlobalTable",getitemtype(item),6)' with a unit of 's'. 'Use Operator(s)' is checked and set to 1. The 'Operator' field contains 'current.centerObjects[1]'. 'Priority' is 0.00 and 'Preemption' is set to 'no preempt'.

Figure 7. Milling times

#### 6. Containers

To adapt the size of the container to the patch size we create a trigger on entry with the code that the teacher had supplied.



```
1 /**Custom Code*/
2 Object current = ownerobject(c);
3 Object item = param(1);
4 int port = param(2);
5
6
7 int Type=item.Type; /*name of the variable on Supply Table*/
8
9 int cant=Table("GlobalTable")[Type][1]; /*Batch size for the Item (tipo) on variable "cant"*/
10
11 treenode target=Model.find("Container1>variables/batchsize"); /*Status on variable target from tree node*/
12 treenode suma=Model.find("Container1>variables/maxcontent"); /*Status on variable suma from tree node*/
13
14 setnodenum(target,cant); /*Substitution of target with Batch size (cant) on tree node*/
15 setnodenum(suma,cant); /*Substitution of suma with Batch size (cant) on tree node*/
```

Figure 8. Code personalized