Guido Fioretti
Gian Paolo Jesi

schedule_sec.xlsx

The file schedule_sec.xlsx provides the simulator with orders. It entails 3 sheets:

1. schedule;
2. production;
3. recipes.

1. schedule

In the sheet *schedule*, the symbol # has the following meaning if it is followed by a number X in the following cell: “At time X do what the ensuing line says”. If the symbol # is followed by anything but a number, then it is simply a comment.

Lines that do not begin with a # say something like this: “macro production every=400.” This expression means: execute the macro entailed in sheet *production* every 400 time units. The sheet production issues a sequence of orders: this sequence is repeated every 400 time units.

Pay attention that the field “every” may eventually be saved as “Every”, which causes the model not to take its value.

2. production

In this sheet, anything that is followed by a # is a comment.

Column A identifies those agents who are able to generate orders. These agents can be either identified with the tags of the orders to be generated, or with the agents’ class, or with agents’ names. If these agents are identified by means of tags, then the contents of columns A and C coincide.

In practice, at the current stage of development only one agent is able to generate orders: *orderDistiller*. This agent is endowed with tags corresponding with orders names. These tags identify *orderDistiller* as the only recipient.

Column B entails the method of *orderDistiller* that is eventually called. In the current version, *doOrder* is the only method that can be called.

Column C entails the sequence orders that must be issued by *orderDistiller*. These orders are identified by their tags.

Column D specifies how many times an order is repeated before switching to the next one in the list.

3. recipes

Anything following a # is a comment. Comments are used as titles of the lines underneath.

Column A entails the tags of orders.

Column B entails a “p”. This specifies that the ensuing cells are used for procuring materials.

For orders implementing recipes that start with a single step, procurement is as follows. The first cell after “p” entails the first step in the recipe (e.g., “papercut”). The subsequent cell specifies how many inputs this step requires (e.g., “1”). This is followed by a cell specifying what sort of input is needed (e.g., “paper”). A cell specifying which agent provides inputs concludes (e.g., “buffer”).

For orders implementing recipes that start with ORs or ANDs, procurement is as follows. The first cell after “p” entails the first step in the recipe (e.g., “papercut”). The subsequent cell specifies how many inputs this step requires (e.g., “2”). The ensuing two cells specify the first input (e.g., “paper”) and which agent provides it (e.g., “buffer”). These are followed by two other cells specifying the second input (e.g., “alu”) and which agent provides it (e.g., “buffer”).

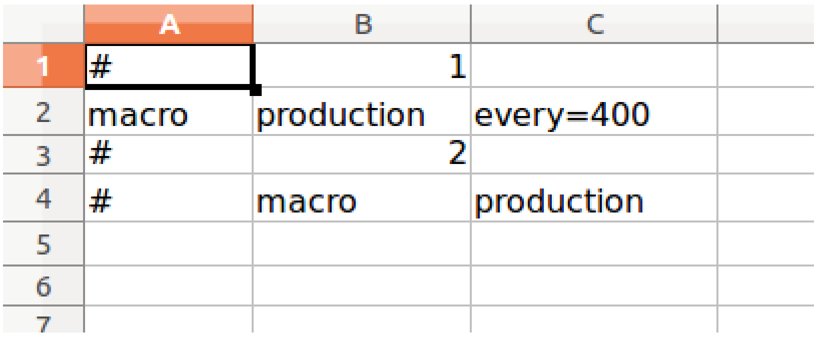
Recipes either entail single steps, or OR steps, or AND steps.

Linear steps are written as follows. In the first cell comes a “,”. The second cell entails the name (i.e., the tag) of the step. The third cell entails an “s” which stays for “seconds” and represents the time unit. Finally, the fourth cell entails a number that specifies how long that step lasts.

Steps entailing either an OR or an AND are written as follows. A cell with either “and” or “or” separates five cells on its left and on its right, respectively. These two sets of five cells represent the steps to be either combined with an OR or with an AND, respectively. Each set of five cells is organized as follows. The two cells on the left and right extremes entail a “(“ and a “)”, respectively. The three remaining cells in the middle entail a linear step. For instance: “(papercut s 1)” is eventually united to “(clean s 3)” by means of an AND. This results with: “(papercut s 1) and (clean s 3)”. Steps entailing either Ors or ANDs do not require commas.

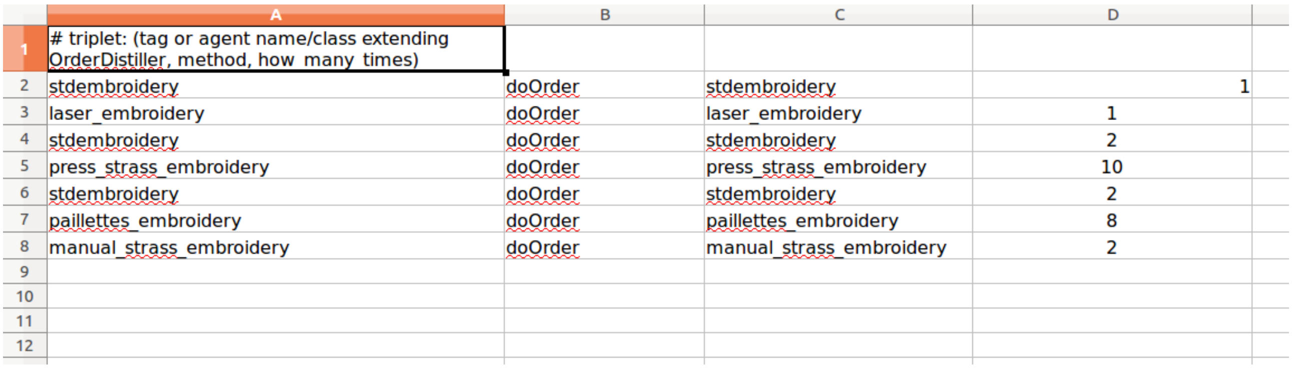
Appendix: The schedule_sec.xlsx of the example Embroidery

1. Example “schedule”



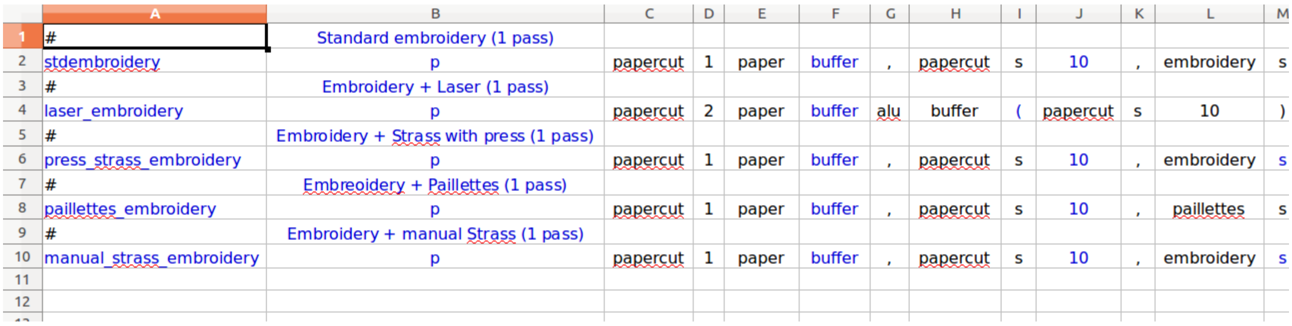
	A	B	C	
1	#	1		
2	macro	production	every=400	
3	#	2		
4	#	macro	production	
5				
6				
7				

2. Example “production”



	A	B	C	D
1	# triplet: (tag or agent name/class extending OrderDistiller, method, how many times)			
2	stdembroidery	doOrder	stdembroidery	1
3	laser_embroidery	doOrder	laser_embroidery	1
4	stdembroidery	doOrder	stdembroidery	2
5	press_strass_embroidery	doOrder	press_strass_embroidery	10
6	stdembroidery	doOrder	stdembroidery	2
7	paillettes_embroidery	doOrder	paillettes_embroidery	8
8	manual_strass_embroidery	doOrder	manual_strass_embroidery	2
9				
10				
11				
12				

3. Example “recipes”



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	#	Standard embroidery (1 pass)											
2	stdembroidery	p	papercut	1	paper	buffer	,	papercut	s	10	,	embroidery	s
3	#	Embroidery + Laser (1 pass)											
4	laser_embroidery	p	papercut	2	paper	buffer	alu	buffer	(	papercut	s	10	)
5	#	Embroidery + Strass with press (1 pass)											
6	press_strass_embroidery	p	papercut	1	paper	buffer	,	papercut	s	10	,	embroidery	s
7	#	Embreoidery + Paillettes (1 pass)											
8	paillettes_embroidery	p	papercut	1	paper	buffer	,	papercut	s	10	,	paillettes	s
9	#	Embroidery + manual Strass (1 pass)											
10	manual_strass_embroidery	p	papercut	1	paper	buffer	,	papercut	s	10	,	embroidery	s
11													
12													