C10 Dining Philosophers II –
Enterprise Dynamics (Taylor ED)

Simulation Approach

Simulator: Taylor ED (Enterprise Dynamics) is a family of software products for modelling, visualisation and control business processes. The 4Dscript Language is the interface through which all Taylor ED functionality is controlled. It's used to to define editing fields, to define atom functionality, to create run and analyse models, to define model logic, to control Taylor ED from outside, etc.

Model: For the implementation of this problem in ED the event based approach was chosen, using only main atoms already available in the ED atom library. The five philosophers are *Servers*, the chopsticks are *Products* to be processed. A *cleaning process* was introduced to manage certain properties.

After having satisfied his hunger the philosopher puts the chopstick back on the table. After a uniformly distributed cleaning time of (1,2) it is now free to be seized on need. In the simulation the chopstick is stored in a queue until being requested again.

Each philosopher may be in one of three status: meditating, hungry or eating. He stays in this status for a uniformly distributed time period in the interval (1,10).

Task a: Simulation/Analysis of behaviour: As expected the philosopher's habit of picking up the left chopstick first leads to their doom: if all five of them get hungry at approximately the same time they will all seize the left chopstick to wait for the right one. Therefore all resources are blocked an the system is caught in a deadlock. The following tables summarize the results for the different status:

	P1	P2	P3	P4	P5
mean	5.55	5.25	5.42	5.18	6.00
St. Dev.	2.84	2.99	2.43	1.60	2.28

Table 1: eating

	P1	P2	P3	P4	P5
mean	6.25	5.54	5.62	4.92	5.75
St. Dev.	2.63	2.54	2.84	2.57	2.60

Table 2 : hungry

	P1	P2	P3	P4	P5
mean	17.09	15.17	13.75	17.36	17.18
st. Dev.	11.56	12.12	12.40	11.53	12.51

Table 3 : waiting

	eating	thinking	waiting
mean	5.47	5.61	16.05
st. Dev.	2.29	2.48	11.05

Table 4: average times for all

ch1	ch2	ch3	ch4	ch5
88	86	86	86	87

Table 5: Chopstick Utilisation

For executing the 50 simulation runs the Experiment Atom was used. Here only the number of runs to be done needs to be set.

Additionally a task to be executed at the end of each simulation run can be defined. This was used to write the needed data into an Excel file for evaluation.

	Time till Deadlock	Percentage Eating	Percentage Meditating	Percentage Waiting
Mean	143.2	21.71145313	27.43	50.86
Stand. Dev.	123.6	10.57447291	9.048	14.1

Table 6: Results for 50 simulation runs

Task b : Deadlock Detection: ED does not recognise a deadlock for what it is- the simulation continues until no more events are listed. Then a time overflow occurs.

To prevent this from happening a query is made each time a philosopher seizes his left chopstick: if a global variable containing the number of philosophers holding a chopstick is equal to five the simulation run is stopped.

```
if(ic(c)=1,do(
inc(model.deadlock),
if(model.deadlock=5,do(createevent(0,atombyna
me([Experiment],model),2),stop)),
```

It may happen that two philosophers try to seize a chopstick at the same time. According to the way ED handles this, the philosopher that comes first in the event list would benefit from being computed first and therefore receive the chopstick.

In the simulation this was blocked by implementing the decision that the philosopher sitting on the right would be given the chopstick.

Shabnam Michèle Tauböck, TU Vienna
shaby@osiris.tuwien.ac.at