

A DEV Approach to ARGESIM Benchmark C6 'Emergency Department' using Enterprise Dynamics

Thomas Löscher, Joachim Aichinger, Wolfgang Ebner, TU Wien, Austria

thomas@loescher.at, e0025538@student.tuwien.ac.at, e0225538@student.tuwien.ac.at

SNE 18/1, April 2008

37

Simulator: Enterprise Dynamics (ED) from In-control Enterprise Dynamics is a software program for discrete event simulation. Furthermore, Enterprise Dynamics is an object-oriented software program for modelling, simulation, visualization, and control of dynamic processes. The users can pick up elements—called atoms—from standard libraries in order to build their own model. ED is based on this concept of atoms as modelling objects in each model. An atom can represent a machine, a counter or a product but can also have a non-physical character like a graph. Thanks to the open structure of ED, the advanced user can build and use its own atoms, for example to model a machine with very specific characteristics. At this point, ED includes 100 standard atoms, but this number is ever increasing. The beginner may typically only need to select from around 30 frequently used atoms to have enough material for his or her applications. Atoms thus are pre-defined modelling objects used to build models quickly and to carry out studies. ED also has a built-in programming language called 4DScript, which can be used for processing specific conditions from reality in the model.

Model: Casualties from accidents are admitted to an emergency department for dressing of wounds. Broken limbs are put in plaster. After a few days a follow-up examination must be performed to monitor the healing process. If necessary, additional treatment will be administered. Follow-up treatment in the emergency department of a hospital is the discrete process to be investigated in this comparison.

In this case a process orientet approach is chosen whereas all treatment points for the patients are modeled by the use of server and queue atoms. Figure 1 shows a screenshot of the model containing the used atoms and their connections. The patient's way starts in the Source atom (S) where they are created. To define the certain types of patients an empirical distribution is used on creation in the Source atom. The possible ways through the system are represented by the atom connections and they are more or less pre-defined for each type at the beginning. Next follows the

registration (R) which is built up by one queue and one server. In the registration server the patients will be admitted to the two casualty wards (CW) with the distribution given in the definition of comparison 6. Each casualty ward is modeled by one queue and two servers representing the waiting room and the two doctors. Also for plaster (P) and x-ray (X) rooms servers and queues are used. After treatment the patients are leaving the emergency department through the sink atom (Sk). To consider that the doctors will start half an hour later a time schedule availability control (T) has to be placed.

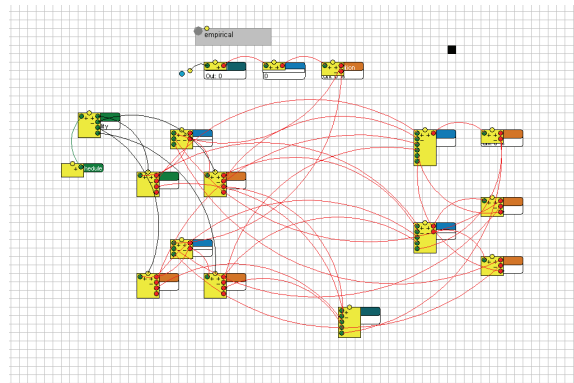


Figure 1: Model with Connections

In addition to the correct linking it is also necessary to program the atoms with certain queries. Several labels are set and used to realize the correct routing of the patients. For example the label called (xray) is defined at the beginning and initialized with the value 1. This label is necessary to know if the patients have already been to the x-ray station. Because after entering this station the value of the label will be increased to 2. So if the patients will be sent to casualty ward again the query of the doctor server will notice the increased value and therefore the output will be controlled.

Such query can look like that:

```
{ If already been to x-ray (xray=2) send to
output 3 (sink) }
if (ddb ([xray], rank(1,c))=2,3,...
```

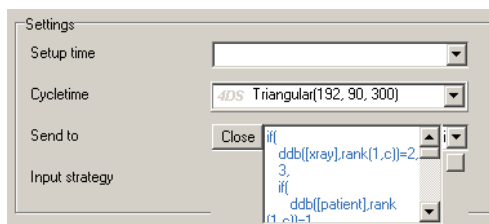
**Figure 2:** Settings for doctor server

Figure 2 shows the screenshot of the parameter window for the doctor server atom. The “send to” parameter is changed depending on the different label values and the patient is routed to the next station corresponding to the number of the output channel.

A-Task: A single simulation run is proceeded in task a. This means that the simulation runs until the stopping criterion is reached. Figure 3 shows the correct setting for the trigger on entry parameter of the sink for 250 patients. If the last patient is leaving the emergency department through the sink after treatment the system directly stops the simulation.

**Figure 3:** Trigger on entry – Sink

Depending on the type of patient the treatment time takes between 94 and 220 minutes (further results see Table 1).

B-Task: As soon as the queue before causality ward 2 contains more than 20 patients the more experienced doctor should take over. Each doctor is modelled as separate server and therefore the physical exchange can be easily implemented by the change of the parameters of the used distribution functions. To change the doctor from an unexperienced to an experienced one the following expression must be set on the cycle time parameter of the doctor server representing the unex-perienced doctor (see Figure 2):

```
1 if(
2   Content(Rank(7, Model))>20,
3   Triangular(246, 168, 378),
4   Triangular(192, 90, 300))
```

The simulation is controlled as in task A and the results and effects can be seen in table 1. Because of the 60 % admission of patients to casualty ward 1, task B (changing Doctors) is not an effective way to lower

mean time	task a	task b	task c
patient 1	274.2	262.2	185.8
patient 2	164.7	182.5	202.1
patient 3	290.5	297.9	233.1
patient 4	154.0	146.3	193.3
standard deriv.	93.3	94.5	92.5
overall treatm. time	372	382	367
close hour	14:12	14:22	14:07

Table 1: Results for task a-c: mean for

the overall treatment time even the mean treatment time of patient 3 is very high.

C-Task: On creation a consecutive number is assigned as label id to the patient atom. In the queues of the casualty wards only patients are arriving with consecutive numbers. Therefore a priority ranking can be established in the casualty ward queues. If a patient is set back from x-ray or plaster with a lower consecutive number he will be ranked in the queue corresponding to his label id. The standard queuing strategy FIFO (first in first out) usually represents the consecutive sequence. On this account a queuing strategy can be used which is based on the consecutive numbers. To do a priority ranking in the casualty ward queues the queue discipline parameter must be set in the following way:

```
t-findqueuepos([id],2)
```

Sort by Label Ascending: The atoms with the lowest value of the label named id are put in front.

But task c (priority ranking) can reduce treatment time a bit, in average of about 289 seconds and also all types of patients have more or less the same treatment time. The results (table 1) show a decrease in treatment time for patients of type 1 and 3 and an increase for the two other types of patients. The standard deviation of the overall treatment time decreases.

Résumé: This work contains a classical event oriented modelling approach using the simulator Enterprise Dynamics.

Corresponding author: Thomas Löscher,
Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria
thomas@loescher.at

Received: June 28, 2007
Revised: November 12, 2007
Accepted: November 20, 2007