Simulation Notes Europe SNE 20(1), 2010, 45-46, doi: 10.11128/sne.20.bn06.09969

A MATLAB-based Solution to ARGESIM Benchmark C6 'Emergency Department' using SimEvents

Gašper Mušič, University of Ljubljana, Slovenia; *gasper.music@fe.uni-lj.si*

Simulator: The popular Simulink time-driven simulation software environment was recently enhanced by SimEvents, an event-driven blockset. The blockset consists of a number of block libraries, supported by a general purpose discrete event simulation engine, but which is embedded in Simulink. SimEvents blocks simulate passing of entities through a network of queues, servers, gates, and switches based on events. Within SimEvents block scheme, entities and events may be generated, and blocks are components that process entities, events, and signals. Communication across blocks is based on both signals and entities. Attributes may be attached to entities, forming a set of entity properties, which may be used to distinguish various types of entities as well as their state during the simulation run. With SimEvents, Simulink grows into a general purpose simulation tool for continuous, discrete-event and hybrid simulation. More information about SimEvents can be found at *http://www.mathworks.com/products/simevents/*.

Model: An emergency department is modelled, where four kinds of patients are admitted. The department comprises a registration, two casualty wards with two doctors each, X-ray room with two X-ray units, and a room where plaster casts are applied or removed. All patients enter the registration, after that their way through the department depends on the severity of their wounds.

Since the model in SimEvents is designed within the graphical user interface, the topological layout of the patient's way through the department is directly mapped into a simulation block scheme. Figure 1 shows a screenshot of the Simulink GUI with SimEvents library. An initial phase of the model development is also shown in the Figure, i.e. the generation of entities that correspond to arriving patients and model of their way through registration facility. The *Start Timer* block is used to attach a stopwatch timer

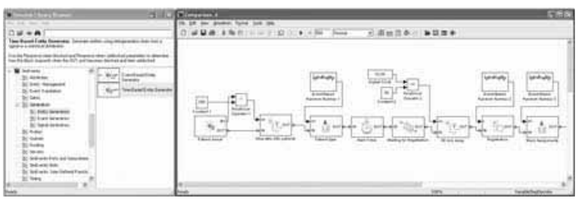

Figure 1. Screenshot of SimEvents.

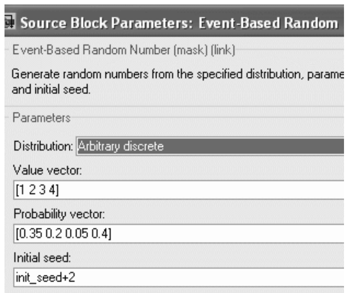

Figure 2. Percent distribution of patients over four types.

to every patient entering the department. The figure shows how the entity generation is stopped after 250 patients and also how the admittance to the registration facility is only allowed after 30 minutes. This way, the requirement about a half hour difference between the time when patients start arriving and the time when doctors start working is modelled. An *Enable Gate* block is employed in both cases.

Statistical parameters are assigned in two ways. Depending on the block and parameter needed, one kind of parameters can be assigned directly via block dialogs, such as the distribution of time between arrivals of patients. Another kind of statistical parameters can be assigned using a random generator block connected to a signal input of an entity processing block. Figure 1 shows several examples of such an assignment, e.g., a patient type is attached as an attribute to the entity representing the patient. The value of the attribute is determined by a random generator, parameters of which are defined through the dialog as shown in Figure 2. In the given case, an arbitrary discrete distribution is used with the values that comply with the problem definition. The `initial_seed` variable is used to enable a number of simulation runs with variable initial conditions of all random generators in the model.

The routing of entities through the model is achieved by the *Output Switch* blocks, where attribute based switching rule is chosen. Path Combiner blocks are used when necessary to merge entities entering a block from different sources. This way the travel path of a specific type of patients through the simulation scheme can be easily followed - illustrated in Fig. 3 showing model part with X-ray Room and Plaster Room.

To improve the readability of the model, a subsystem generation feature of Simulink is used as an abstraction mechanism. This way, individual department facilities are mapped to subsystems, such as X-ray

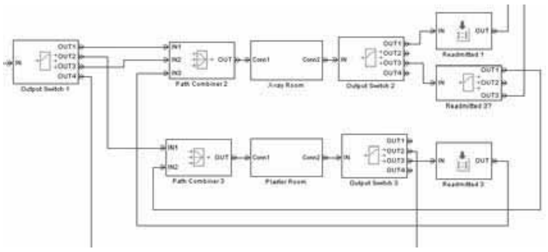

Figure 3. Routing of patients.

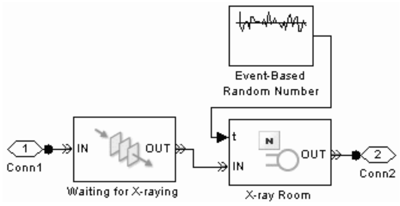

Figure 4. X-ray Room subsystem.

Room and Plaster Room subsystems. The contents of such a subsystem are shown in Figure 4.

To enable statistical evaluation, required quantities are collected in the simulation model by the use of Read Timer and Discrete Event Signal to Workspace blocks. The statistical analysis is then performed by the use of a simple m-script in MATLAB.

A-Task: In SimEvents, a single simulation run is started in a standard Simuling fashion, i.e., within the GUI by pressing the "Start simulation" button, by choosing "Start" in the Simulation menu or by presing Ctrl+T. To obtain results that are representative, however, a number of simulation runs must be initiated with varying initial seed of random generators. This is automated within the MATLAB script that is sketched here:

```
system_name='Comparison_6';
load_system(system_name);
nruns = 100;
w = zeros(nruns,6);
h = waitbar(0,'Please wait...');
seedarray = ceil(rand(nruns,1)*999999)*2+1;
for k = 1:nruns
   waitbar(k/nruns,h);
   init_seed = seedarray(k);
   sim(system_name);
   w(k,1) = mean(...)
          ...
   w(k,5) = std(...);
   w(k,6) = max(...);
end
close(h);
result = mean(w);
```

The average results of 100 simulation runs are shown in Table 1.

Mean Time	Task A	Task B	Task C
Patient 1	223	237	166
Patient 2	146	155	164
Patient 3	242	248	177
Patient 4	135	144	159
Std. Dev.	82	94	75
Ov. treat.time	382	420	381
Close hour	13:52	14:30	13:51

Table 1. Results for tasks a - c: mean of treatment times, standard deviation and closing time.

B-Task: The Casualty Ward model is changed in a way which enables switching between two alternative configurations. Additional *N-Server* and *Output Switch* blocks are inserted to each Casualty Ward subsystem and the switching signal depends on a number of entities waiting in the queue before Casualty Ward 2. A *Discrete Event Subsystem* block implementing simple set/reset signal logic is used to generate the switching signal.

This strategy yields an increase of treatment times for all types of patients, for the standard deviation and also for the overall treatment time (see Table 1).

C-Task: Priority ranking is achieved by inserting separate queues for re-admitted patients before each server in Casualty Ward subsystems. The Path Combiner block, which accepts entities from both queues and outputs them to the server is then configured to accept entities from "re-admitted" queue with higher priority. The Results (Table 1) show a decrease in treatment time for patients of type 1 and 3, and an increase for the others. The standard deviation also decreases while the overall treatment time does not change significantly.

Resumé: The emergency department problem is well suited to simulation with the SimEvents blockset. Patients passing through the department are naturally mapped to flow of entities through the blocks of simulation scheme. Decisions are easily modelled by the use of *Output Switch* blocks where the path of the entity is chosen in dependence of an attribute value. The integration into MATLAB allows for advanced statistical evaluation of the simulation results.

Corresponding Author: Gašper Mušič,
Fac.Electrical Engineering, University of Ljubljana
Tržaška 25, 1000 Ljubljana, Slovenia
gasper.music@fe.uni-lj.si

Received: March 14, 2009 (Rev.Dec.10,2009)
Accepted: January 10, 2010