A MATLAB-based Solution to ARGESIM Benchmark C2 'Flexible Assembly System' using SimEvents

Tobias Pingel[*], Sven Pawletta

Wismar University of Applied Sciences, Computational Engineering and Automation, , Philipp-Mueller-Strasse, 23952 Wismar, Germany; *tobias.pingel@hs-wismar.de

Simulator. SimEvents, part of the MATLAB product family since 2006, extends Simulink by a blockset for general purpose transaction-oriented discrete-event simulation based on entities. With libraries of predefined blocks, such as queues, servers, and switches, SimEvents can be used for simulating manufacturing processes to determine resource requirements, throughputs and to identify potential bottlenecks, for example. Statistical evaluation can be performed during simulation with the use of the block's statistics signal output port and appropriate signal scopes. These signals can e.g. be the number of departed entities or average utilization of a server.

For implementing the desired control strategy, to control the flow of entities, SimEvents offers several possibilities. Here, the blocks' control strategy can be parameterized directly, can be based up on attributes for e.g. the routing path or service time or based on Stateflow state charts via the signal input ports if more complex control is desired. The simulation of the entity flow itself is based on an event list. If an entity is advanced from one block to the next, events are generated which cause subsequent entities to be requested, in return.

Modeling. As illustrated in Figure 1, the *C2 Flexible Assembly System* consists of eight nearly identical assembly subsystems (A1-A6) as well as a subsystem for entity generation. These subsystems are connected together for realizing the material flow. The pallets are represented by entities.

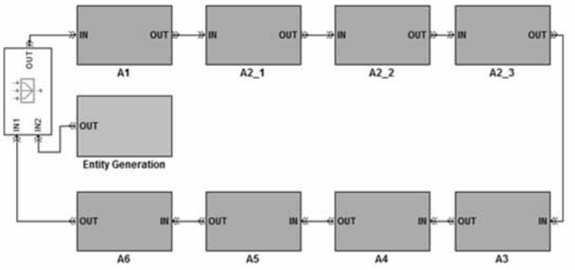

Figure 1. Flexible Assembly System.

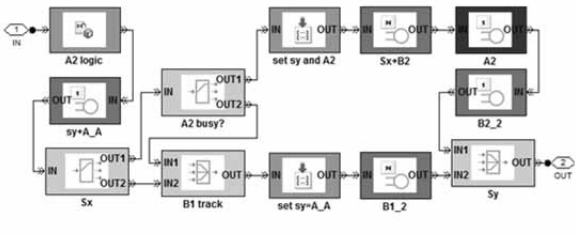

Figure 2. A2_1 Assembly Subsystem

Each subsystem (see Figure 2) is modelled by using standard blocks of the SimEvents library. The conveyor belts, the assembly station as well as the shifting process are represented by a *Single Server* or *N-Server* with appropriate service times. For entity routing *Output Switches* and *Path Combiners* are used. Attribute manipulators (*Set Attribute* and *User-defined Attribute Function*) are used for implementing the desired control behaviour.

By entering the subsystem the *A2 logic* block determines – based on the *ServiceCompletion*-flag – whether the pallet is routed to B1 or B2 (Listing 1).

The pallet then passes the conveyor belt *A_A*, where the service time – in the attribute *sy* – with a value of either $t_{transport} + t_{shift}$ or just $t_{transport}$ depending of whether the pallet need to be shifted back to the main track in the previous assembly subsystem. This design is required since *Sy* has no storage capacity.

Attribute	Purpose
A1 … A5	ServiceCompletion-flag
p	Path for pallet at *Sx* (*B1* or *B2*)
sy	Transportation time at *Sy*
rt	Read timer (only *A1*)
T1	Time stamp (used by *Timer* blocks)

Table 1. Entity attributes.

```
function out_p  = fcn(A1, A2, A3, A4, A5)
out_p = 2;
if A1 && ~A2 && ((~A3&&~A4&&~A5)||(A3&&A4&&A5)),
    out_p = 1; end
```

Listing 1. A2 logic (User-defined Matlab Function)

In the next step, the pallet is routed based on *p* to *B1* or *B2* at *Sx*. If the *B2* conveyor belt is busy the pallet is routed back to B1. The routing strategy of *A2 busy?* is set to ‘First port that is not blocked’. On the *B2* track the entity attributes *A2* and *sy* are modified, where on the *B1* track *sy* is just set to $t_{transport}$ with a service time of 4/3 seconds (0.4m / 0.3m/s).

For realizing prioritization at *Sy* the control strategy is set to ‘IN1 port precedence’ and a higher *Service completion event priority* is defined for *B2_2* than for *B1_2* to guarantee accurate processing in the case of simultaneous event occurrence.

A-Task: Control Strategy and Evaluation. The control strategy is implemented by using entity attributes, user-defined code as well as predefined block strategies. The attributes are modified by manipulator blocks while traversing through the whole assembly system and are routed and delayed based up on these.

As illustrated in Figure 3, time measurement and counting of finished workpieces are realized using *Timer* blocks within the *A1* subsystem. Here, based on the attribute *rt* the pallet is routed to the *Start Timer* block first, equipped with a time stamp attribute *T1* and then evaluated by the *Read Timer* block, if the workpiece is unloaded. By again passing the *Start Timer* block *T1* is being reset.

The recorded data is displayed using Signal Scope blocks and sent to the MATLAB workspace using appropriate blocks for data preparation.

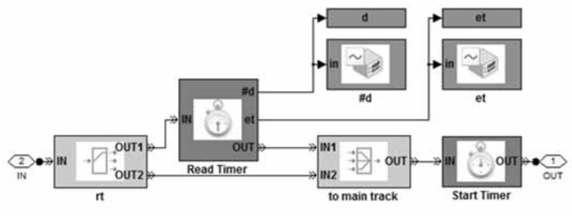

Figure 3. Time measurement and workpiece counting.

B-Task: Simulation-Throughput. The simulation study had been performed using a MATLAB script as sketched in Listing 2. Here, ten to just 49 pallets were used, since a deadlock occurred by using 50 or more entities. Then, the relevant data from the 120th to the 600th minute (eight hours shift excluding the warm-up period) was extracted from the recorded signals *d* and *et*. Based on this data the total throughput as well as the average throughput time per simulation could be computed (Listing 2). The results of the simulation study using 10 to 49 pallets are listed in Table 2.

```
for np = 10:49,
    sim('C2_FAS.mdl', 36000); k = np-10+1;
    [tp(k) at(k)] = evaluate_sim(d, et);
end
```

Listing 2: Simulation study with 10 to 49 pallet.

Number of pallets (np)	**Total through-put (tp)**	**Avg. throughput time in sec. (at)**
10	959	285
11	1072	280
12	1185	276
13	1298	273
14	1439	265
15	1439	285
20	1439	385
25	1439	396
30	1440	586
35	1438	682
40	1437	787
45	1440	882
50…60	(deadlock)	(deadlock)

Table 2. Simulation results.

C-Task C: Pallet Number Optimisation. For automatical optimization a user-defined parameter optimization function (Listing 3) had been implemented in MATLAB. Therefore, an objective function $|tp-at|\cdot tp$ with respects to minimum average throughput time and overall throughput was used. The optimum is 14 pallets, yielding into a throughput of 1439 and an average throughput time of 265 seconds.

```
[opt idx] = min(abs(tp-at).*tp);
tp_opt = tp(idx); at_opt = at(idx);
np_opt = idx-1+10;
```

Listing 3: Simulation study with 10 to 49 pallets.

Resume. SimEvents – as a simulation tool for general purpose discrete-event simulation – is well-suited for simulating manufacturing systems. The model can be build up individually and at any level of detail using a variety of simple standard blocks. With predefined strategies the process flow can be controlled locally. If more complex control is required, SimEvents can be extended by Stateflow using state charts.

Submitted: September 2010
Accepted: June 30, 2011