

An Object-Oriented Solution to ARGESIM Benchmark C2 'Flexible Assembly System' using TaylorED

Thomas Löscher, Yilin Huang, Voin Legourski, Ondrej Cevan, TU Wien, Austria
thomas@loescher.at, yilin@gmx.net, legourski@yahoo.com, ondrejcevan@yahoo.com

Simulator: Taylor Enterprise Dynamics (TaylorED) is an object-oriented comprehensive tool for modelling, simulating, visualizing, and monitoring dynamic-flow systems. Atoms (comparable to classes in object-oriented programming languages) are TaylorED's reusable "smart objects" that inherit behaviours from their parent atoms. TaylorED's standard atom library offers over 170 predefined atoms (new atoms are available monthly to download). Users can create and customize their own atoms and building-blocks with "four dimensions": location, speed, rotation and dynamic behaviour over time. Models can be viewed dynamically and simultaneously in 2D, 3D, and Virtual-Reality animations. With "Atom Concept", the soft-ware is powerful, flexible, and easy-to-use for model building and simulation analysis. Simulation results may also be exported to external software programs.

Model: The Flexible Assembly System model is composed of eight almost identical submodels. Each of them contains seven standard atoms as shown in Figure 1.



Figure 1: Screenshot of a submodel

The assembly-station (A1), shifter1 and shifter2 are represented by Server atoms, which have attributes Cycletime among others to configure the durations that pallets being processed by the Server.

The conveyor belt (in dark green) leading pallets to the assembly-station is represented by an Accumulating Conveyor atom. The belt leading pallets away from the assembly-station and the one direct connecting shifter1 and shifter2 are represented by Non-Accumulation Conveyor atoms. The length, speed, capacity, etc. of a conveyor belt can be configured. A Queue atom is placed after shifter2 to perform priority checking of pallets from the two different belts. Additionally, the pallets being processed in the sys-

tem are Product atoms. Loading the pallets into the system and unloading of them are realized by Source and Sink atoms. Lock and Unlock atoms are responsible for controlling the amount of pallets circulating in the system.

More advanced strategy controls, such as input/output channels, triggering-events, label-settings, exporting time-stamps to Excel files, as well as other simulation constrains can be defined by using 4Dscript, which will be described as following.

Shifter1 decides whether it shall shift a pallet towards an assembly-station through the upper belt (by output channel 1), or else directly shift the pallet to shifter2 through the lower belt (by output channel 2). The decision is made according to a pallet's label value (which shows if the pallet has been in the station) and to the fact if the belt (in dark green) towards the station is full. The 4Dscript of shifter1, e.g. in front of station A2, is listed below.

```
1 if(label([step1], rank(1,c)) = 2, 2,
2 if(or(label([step2], rank(1,c)) = 0,
3     label([step2], rank(1,c)) = 12),
4 if(content(out(1,c)) =
5     att([Capacity], out(1,c)), 2, 1), 2))
```

Each pallet will get an updated label-value after being processed and before leaving an assembly-station. The 4Dscript of setting labels, e.g. in station 5, is shown below. Station 3, 4 and 5 will add a value of 3, 4 or 5 correspondingly to label step2. Station 2 will have the label step1 value set to 2.

```
6 setlabel([step2], label([step2], i)+5, i)
```

The pallets' colors will be changed after each assembly processing for visualization purposes. For example, if all steps of station 2, 3, 4 and 5 have been completed, the icon of a pallet will be set to a green circle.

```
7 if(
8 Label([step1], i)=2,
9 if(
10 label([step2], i)=0,
11 set(icon(i), 27),
12 if(
13 label([step2], i)<6,
14 set(icon(i), 29),
```



```

15  set(icon(i),24)
16  )
17  ),
18  if(
19  label([step2],i)=0,
20  set(icon(i),47),
21  if(
22  label([step2],i)<6,
23  set(icon(i),50),
24  set(icon(i),48)
25  )
26  )
27  )

```

A-Task: The timestamps that each pallet entering and exiting the assembly-system are exported to an Excel file for statistic purpose. Entering time is recorded by Source atom using

```
28 ExcelWrite(output(c), 1, time),
```

whereas entering time is recorded by Sink atom using

```
29 ExcelWrite(value(name(i)), 2, time).
```

B-Task: The system model was simulated for eight hours with different amounts of pallets (from 15 to 60) circulating in it. The throughputs were counted by the *Monitor* facility of the simulation tool. The throughput per hour can be summed dynamically by the tool as well. The throughput time of each pallet was calculated by Excel.

	Throughput	Average Throughput time	Min Throughput time	Max Throughput time
15	1255	343.98	213.67	406.33
16	1365	337.30	213.67	407.33
17	1440	339.79	240.00	400.00
18	1440	339.71	240.00	400.00
19	1440	380.00	340.00	400.00
20	1440	399.96	340.00	450.00
40	1440	790.77	524.00	17730.00
60	1440	1133.78	402.00	5740.00

Table 1: Throughput and throughput time of pallets

C-Task: The data in Table 1 shows that the increased amount of pallets in the system doesn't necessarily result in an improved performance; on the contrary, it may downgrade the performance. With 17 pallets on the conveyors, the throughput already reaches its peak, and the throughput time is considerably efficient. Above the number of 17, the throughputs stagnate at 1440, and the throughput time start to rise gradually. Specially when there're more than 20 pallets in the system, the level of throughput

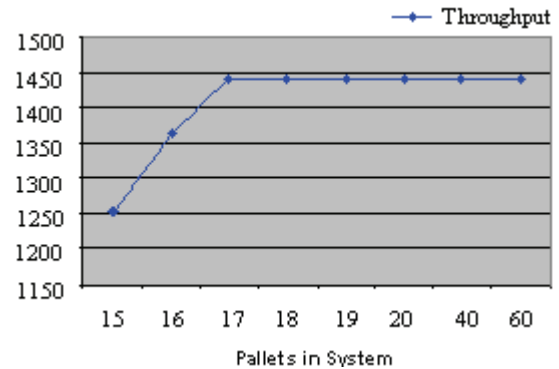


Figure 2: Throughput of 20 ~ 60 Pallets

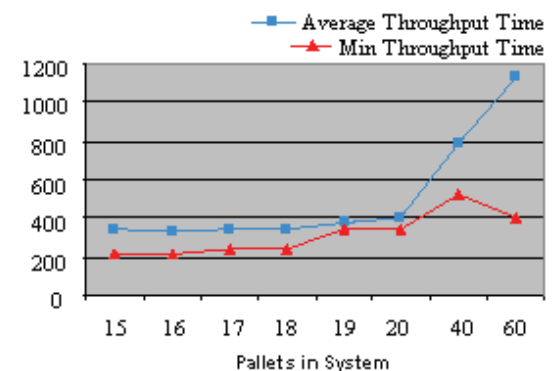


Figure 3: Throughput Time of 20 ~ 60 Pallets

time surge dramatically, as the pallets remained substantially longer in the system because of the overcrowded traffic.

Resumé: This work shows a classical object oriented modelling approach using Taylor Enterprise Dynamics. The statistical evaluation is done by the use of an external Excel file where the shown figures could be easily created.

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

Received: June 20, 2007

Revised: October 10, 2007

Accepted: October 20, 2007