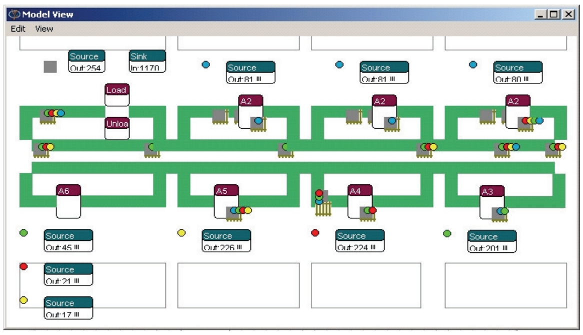
An OO Process - Approach to ARGESIM Comparison C2 Flexible Assembly System with Enterprise Dynamics (Taylor ED)

E. Pazourek, F. Breitenecker, TU Vienna
fbreiten@osiris.tuwien.ac.at

Simulator: Enterprise Dynamics (ED) is an object-oriented software for modelling, simulation and visualisation of dynamic processes. In ED everything is an atom: an application, a model, a library, a product, etc. An atom has four dimensions: location and speed in space and dynamic behaviour. The atoms are hierarchically structured (a main atom contains all others), they can be displayed in tree structures. A model can easily be created by dragging atoms from the Library Tree into the Model Layout window. The programming language 4DScript allows defining more complex atom functionality, to run and analyse models, etc.

Model: This model is built of standard atoms of the ED library. It consists of eight *Composition Container* atoms, which represent the eight subsystems of the. As ED is object-oriented it was possible to define one subsystem and create several instances of this atom. Each subsystem was then adjusted by adding additional 4DScript code and additional atoms.

Every *Composition Container* contains at least five *Fast Accumulating Conveyors*, one *Assembler*, one *Source* and one *Product*. The first subsystem, for example, also contains an *Unpack* atom for unloading finished parts. One additional *Source* was used for putting empty pallets on the conveyors. For easier graphical representation the unprocessed part that is put on the pallet in station A1 is not modified, but additional products of different colours are loaded onto the pallet in each station (see following figure with model):



Task a:Control Strategy/Statistical Evaluation. Whether a pallet enters a station depends on the values of labels ai, which are set to 1 when the product undergoes operation Ai.

The routing is implemented in the *Accumulation Conveyor* atoms, which represent B1. The following 4DScript code is written in the *OnOcReady* eventhandler in the *Atom Editor* for routing pallets to station A5.

```
if(
 <>(content(c),0),
 do(
  setstatus(5,c),
  if(and(label([a1],first(c)),
       not(label([a5],first(c)))),
     if(and(=(content(out(1,c)),0),
            icopen(1,out(1,out(1,c)))),
        moverequest(first(c),1),
        moverequest(first(c),2)),
     moverequest(first(c),2)))
```

Results are collected by using an *Experimentation* atom, which makes it possible to set the number of observations, the run length and the warmup period. When a product is put on a pallet in station A1, it gets a label with the current time, which is readout when the product is unloaded. The throughput time of each product is written into an Excel for further evaluation.

Task b: Simulation Results - Throughput. The following table shows the results for different numbers of pallets. For each number of pallets 100 simulation runs were made, each with a warmup period of 2 hours and a run length of 8 hours.

Number of pallets	Total throughput	Average throughput time in sec.
13	1320	261
14	1440	258
15	1440	278
16	1440	298
17	1440	318
18	1440	338
19	1440	358
20	1440	378
40	1440	774
60	1440	1155

Task c: Simulation Results - optimisation. The total throughput stayed the same for 20, 40 and 60 pallets (results task b). Even with 14 pallets the maximal possible number of processed parts, which is 1440, was reached. A higher number is not possible, because the three stations A2 act as a bottleneck. Great differences can be seen in average throughput time. The lowest average throughput time is reached with 14 pallets and it increases nearly linear with increasing number of pallets. So no optimisation is necessary, and the optimal number of pallets is 14 with maximal throughput and minimal throughput time.

C2 Classification: Object- / Process-oriented Modelling

Simulator: Enterprise Dynamics V.4.5