Using Resources to Model CPU and Memory

This example shows how MLDesigner can be used to model memory and CPU resources using Discrete Event abstraction techniques.

Packets compete for both memory and CPU resources during processing. Packets are loaded into memory and then presented to the CPU for processing. When processing is complete, the memory is released. Memory is represented with quantity resources and the CPU is represented with server resources. A data structure represents the packet and carries information between the blocks.

Packets are generated by the PacketSource blocks and assigned fields that define memory and CPU requirements. Packets are passed to the AllocateBasic block, where they are assigned memory (from a quantity resource block.) Once memory has been allocated, the packet is passed to the CPU block where it is processed, based on priority and CPU requirements. (Preemptive is allowed and switching overhead is set at 0.1.) Once CPU processing is complete, the packet is passed to the FreeBasic block, which frees the memory allocated to that packet and returns it to the memory pool so it is available for subsequent packets. The figure below shows the base model.

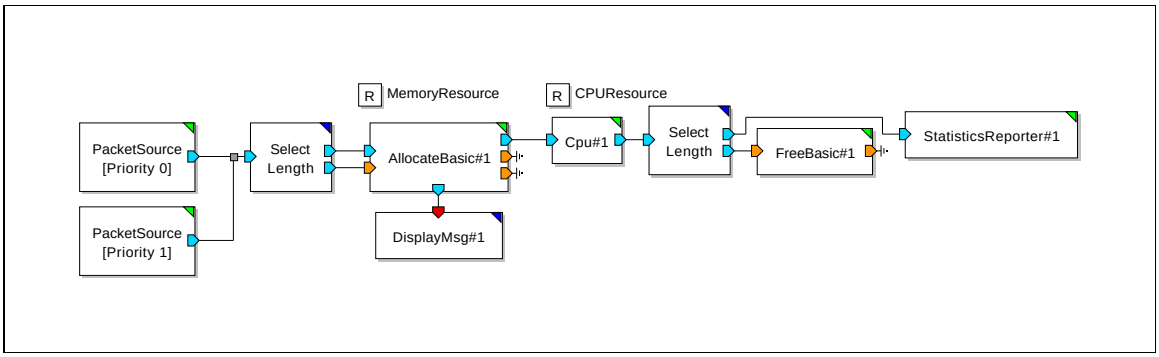
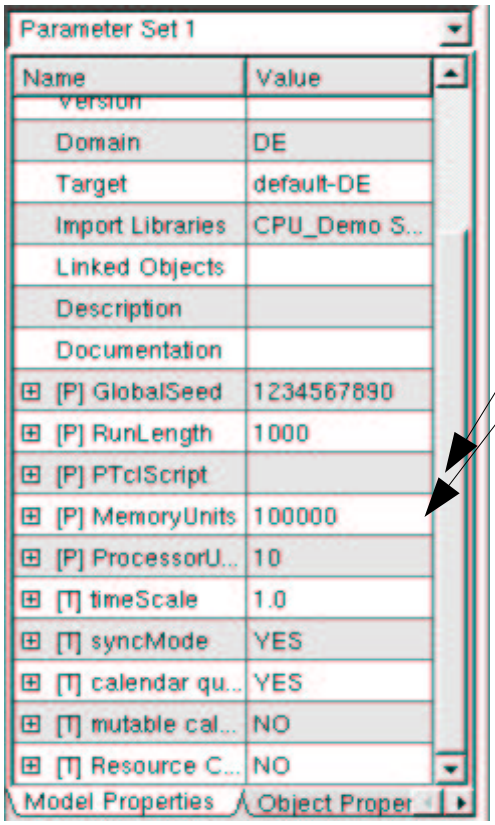


Figure 1: Top-level CPU and Memory model

The model has four primary parts: create packets, allocate memory, process packets and free memory. There are two identical PacketSource blocks to the left side of the model: one for priority 0 packets and one for priority 1 packets. The AllocateBasic block, located to the right of the PacketSource blocks, allocates memory to store the packets based on packet size. The CPU block in the center processes the packets using CPU resources. The Free Basic block frees the memory used to store the packets once they have been processed by the CPU block.



Parameters size the memory units and the number of processor units in the CPU.

Details of the Packet Source Block are shown below.

Figure 2: Parameter menu

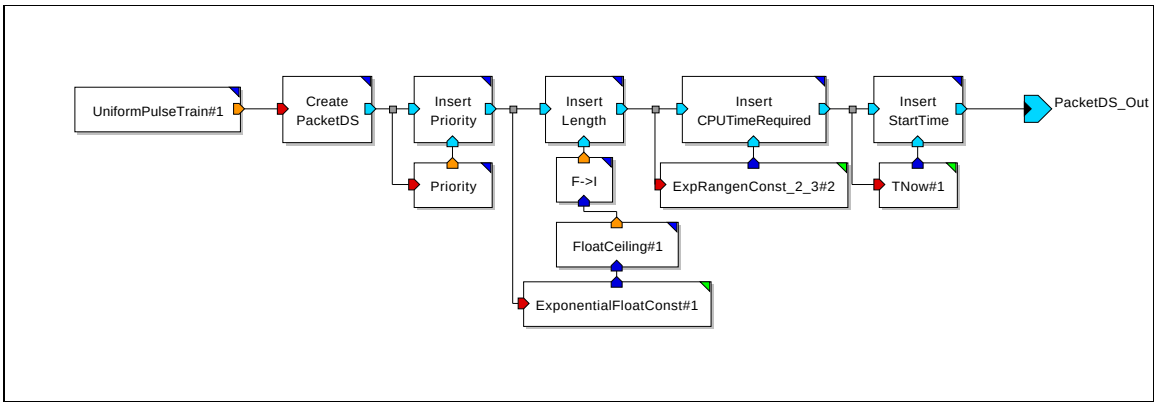


Figure 3: Packet souce block

Details of the CPU Block are shown below.

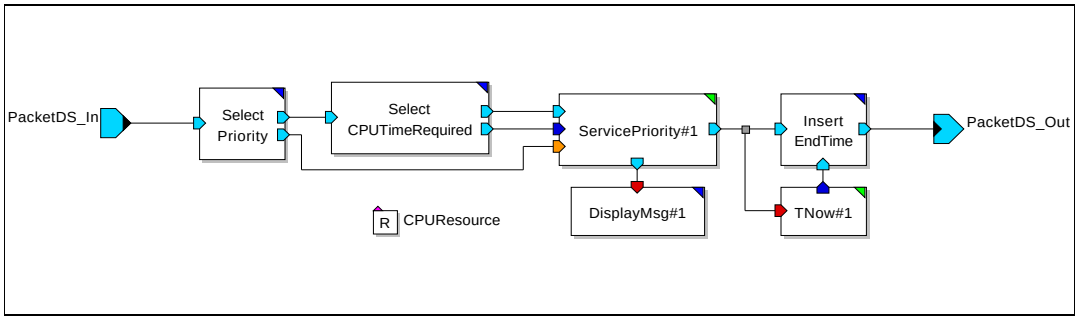


Figure 4: CPU block

Details of the AllocateBasic block are shown below.

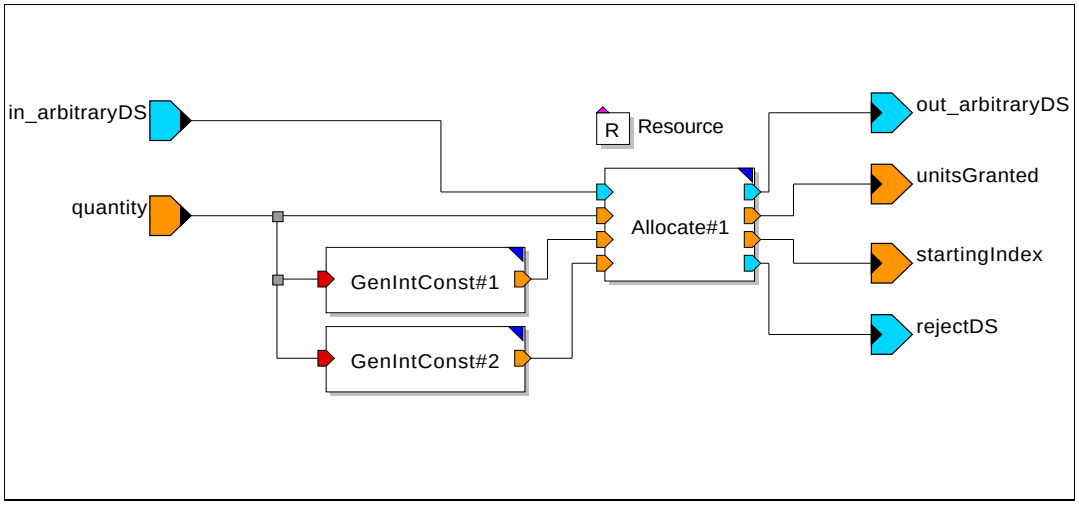


Figure 5: AllocateBasic memory block

Details of the free memory block are shown below.

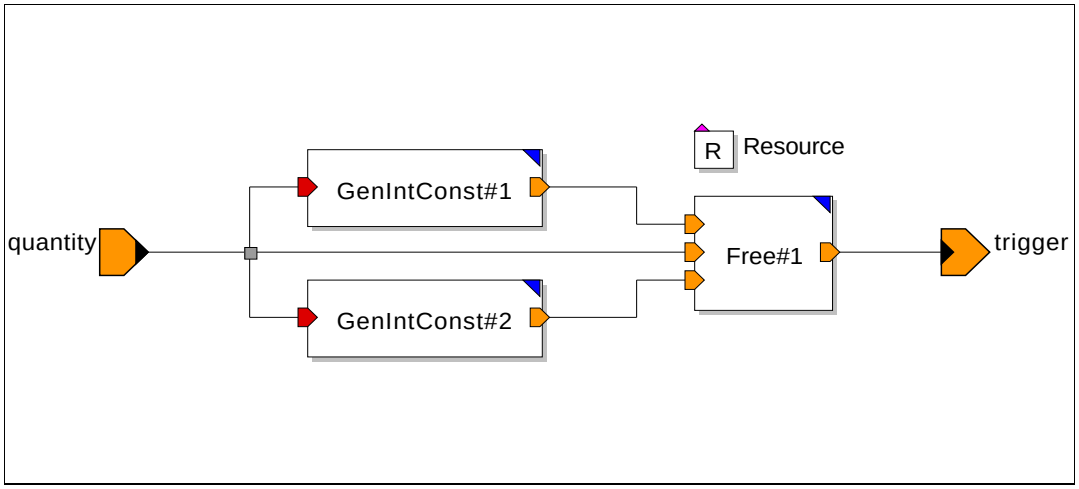


Figure 6: FreeBasic memory block

Details of the statistics block are shown below.

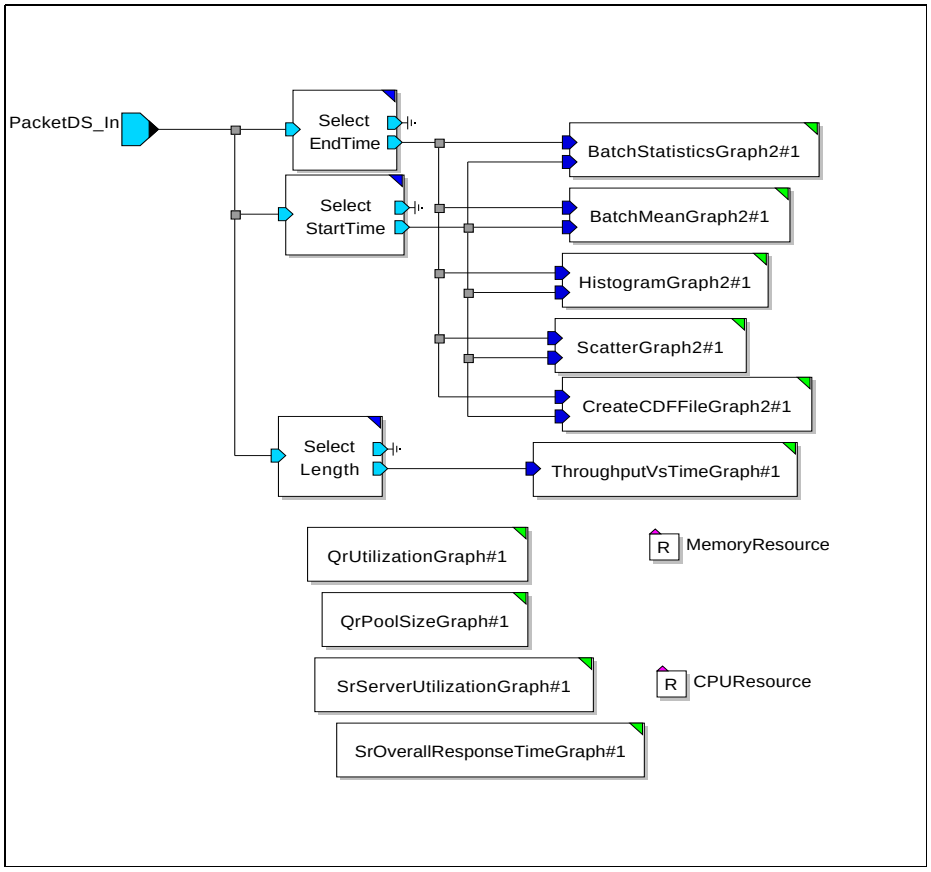


Figure 7: Statistics block

Three statistical reports from the simulation model are shown below.

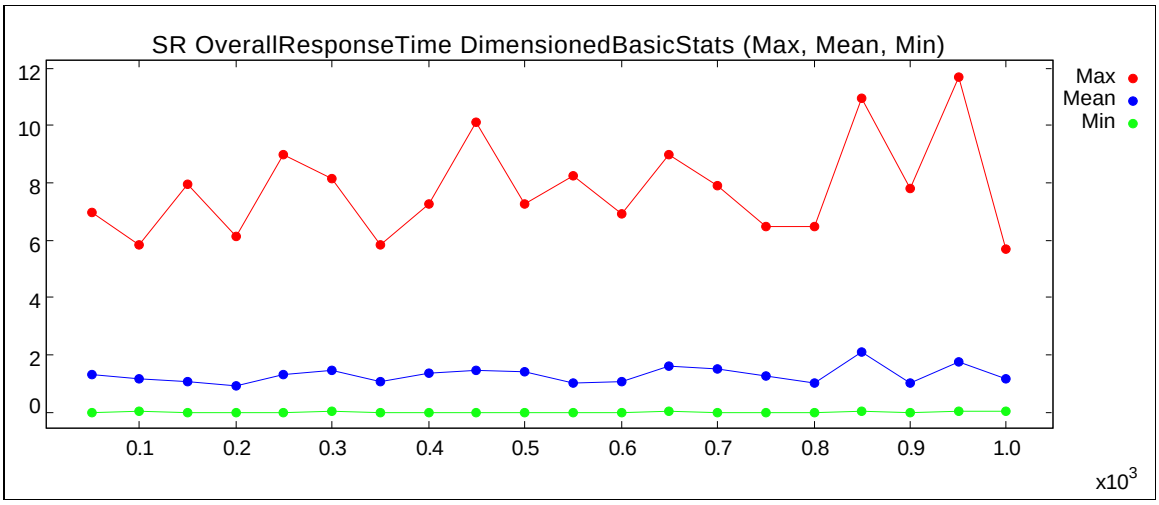


Figure 8: CPU overall response time

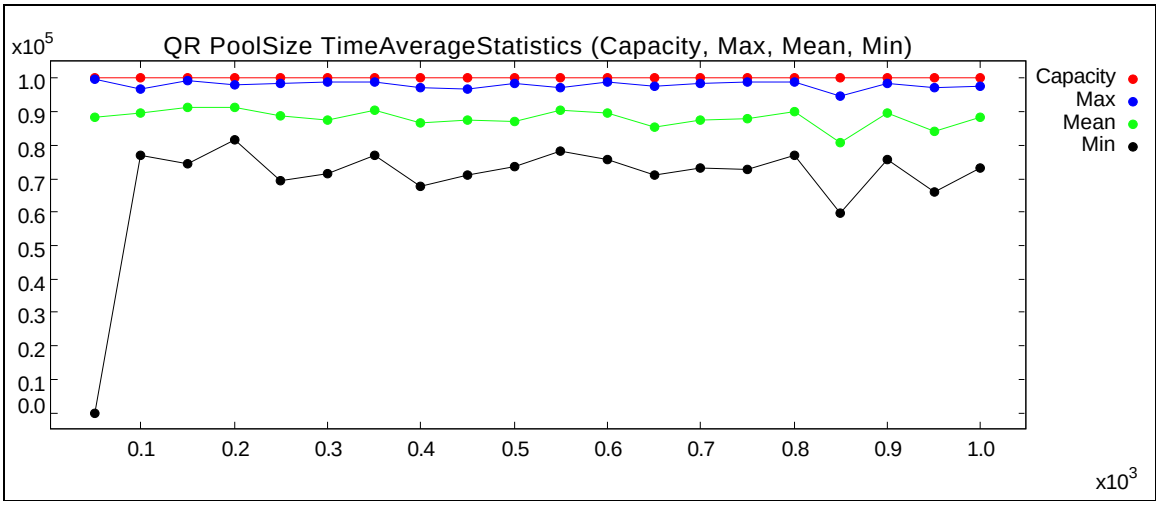


Figure 9: Memory pool size

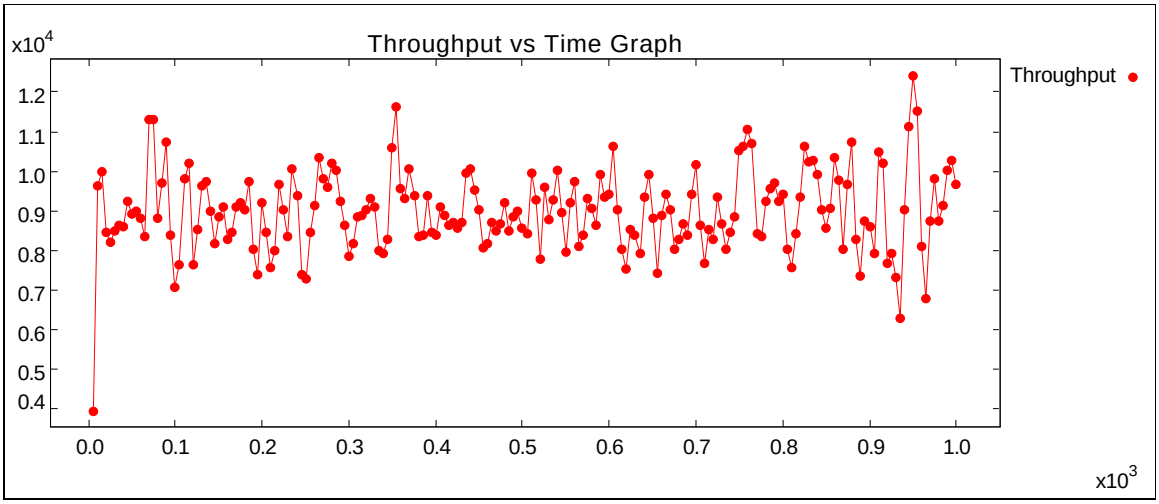


Figure 10: Throughput vs time

Note: Model developed by Dr. Keyvan Farhangian of KVON Technologies.