Proceedings of the 2000 Winter Simulation Conference
J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds.

USE OF DISCRETE EVENT SIMULATION TO VALIDATE AN AGENT BASED SCHEDULING ENGINE

Shubhabrata Biswas
Sara Merchawi

DELMIA Corporation
5500 New King Street
Troy, MI 48098, U.S.A.

ABSTRACT

This paper discusses the use of simulation in a new context. Most often QUEST is viewed as a stand-alone simulation tool to analyze and understand shop floor behavior. It has rarely been used in conjunction with other proprietary software. This paper attempts to demonstrate how QUEST is being used in conjunction with an agent based scheduling software (henceforth referred to as Scheduler) developed using Java. The Scheduler is a multi-threaded application using software agents. It can run in a distributed manner over a LAN. This paper deals with the agent based architecture of the Scheduler as well as the important role of simulation in validating the Scheduler.

1 OVERVIEW

Most factory scheduling systems seek the optimum schedule for an assumed stable configuration and cannot react rapidly to changes in resources. Moreover, most are not integrated into a unified system for task monitoring, scheduling, simulation, and execution. DELMIA Corp. proposes to improve on existing systems by validating a new task scheduling and execution system in which software agents represent factory resources, systems, and jobs.

A software agent is an autonomous software code that takes independent action in response to local conditions. Software agents generate flexible, responsive system behavior.

The proposed Agent Network for Task Scheduling (ANTS) system would continuously adapt to new conditions rather than simply finding an optimum schedule, which is meaningless in a dynamic environment.

The ANTS system would adjust schedules in line with resources, assist in recovery from faults in the factory and management interruptions in the supply chain, and dispatch work against the schedule, thereby streamlining the flow of material and services from an integrated supply chain. The agents can be upgraded independently as new functions are needed or new algorithms are discovered. ANTS integrates four prototype technologies: an agent-based symmetrical scheduling architecture (in which all aspects of the factory are represented as agents), market-based mechanisms for agent coordination, algorithms for response to varying demand, and independent scheduling for different parts of the factory.

DELMIA Corp., has developed an infrastructure that can support large agent communities, algorithms for scheduling and execution, mechanisms for agent-factory interactions, and user interfaces. The system will be validated at a large shipyard. The ANTS system is expected to improve factory throughput, lead time and agility while also reducing production costs. Potential annual productivity savings for the strategically important shipbuilding industry could be very significant.

2 THE QUEST MODEL

A simulation model of a typical shop floor is being used to substitute for a “Factory Information System” or an MES system which is planned to be used in the future. The advantage of using simulation for this validation process is two fold.

1) It allows us to control the conditions being created in the shop floor, thus we can control when, where and how frequently resource failures occur. This allows us to check our scheduling algorithms under various combination of factors.
2) It allows us the option of racing ahead in time and performing “What If” type of analysis to judge the relative merits or demerits of our scheduling algorithms.

The Scheduler is the driver of the simulation model. The simulation model executes the instructions issued to it by the Scheduler. After it executes the command, it updates the Scheduler about its new status. Then the Scheduler sends it its next set of instructions and this process continues on iterating. The Scheduler and the QUEST model always stay in sync.

The key inputs to the QUEST validation model are

i) Resource Failure distribution
ii) Part Failure distribution
iii) Delay to diagnose data
iv) Shift data

The behavior of the simulation model can be controlled through an input file that is used to specify the above operating parameters.

The QUEST model opens a socket for communication with a "Message Broker". It handles all messages going between the simulation model and the Scheduler. This set up allows us some unique advantages.

We do not need to know the exact destination of the server or servers running the Scheduler. Both QUEST as well as the Scheduler register with the Message Broker using unique registration ID's. When QUEST has to send a message to the Scheduler, it just adds the Registration ID of the Scheduler to the message and posts it to the Message Broker. The Message Broker locates the server running the Scheduler and passes on the message to it. This design also allows us to run multiple QUEST models with a Scheduler. The Scheduler may itself be running on multiple servers. Please refer to Figure 1 for a pictorial representation of the set up.

3 AGENT BASED SCHEDULER

ANTS uses agent technology because of the promise that large populations of agents have shown in addressing complex problems. Agents generate rich system behavior which is less brittle than centralized code when the system is modified and which adapts to the changing conditions on the plant floor.

In ANTS, the schedule *emerges* from the behavior of the system. The goal in ANTS is not to find an optimal schedule, which is meaningless in a dynamic environment, but rather to continuously adapt in the best manner possible to the changing conditions of the plant floor. Rather than just alerting the management to problems, ANTS provides management with better information about delivery windows and cost of production using an activity-based costing model.

Each Agent makes decisions based on its own individual priorities and rules. When done in large numbers this individualistic behavior leads to the emergence of a collective behavior which is highly adaptive and responsive to shop floor changes.

3.1 Distributed Architecture

The implementation of the Scheduler has been done in JAVA. The strictly object oriented nature of JAVA allows for good modularization and re-use of code. Once a particular agent behavior is defined in a class, it is easy to create multiple instances of that agent by simply creating multiple objects from the agent class. Thus the object oriented architecture of JAVA ties in well with the agent based approach used for the Scheduler.

The design of the Scheduler allows for agents to be distributed over multiple servers. An Object Request Broker (ORB) takes care of maintaining information on all agents and forwarding messages across the Local Area Network (LAN). Please refer to Figure 1 for details.

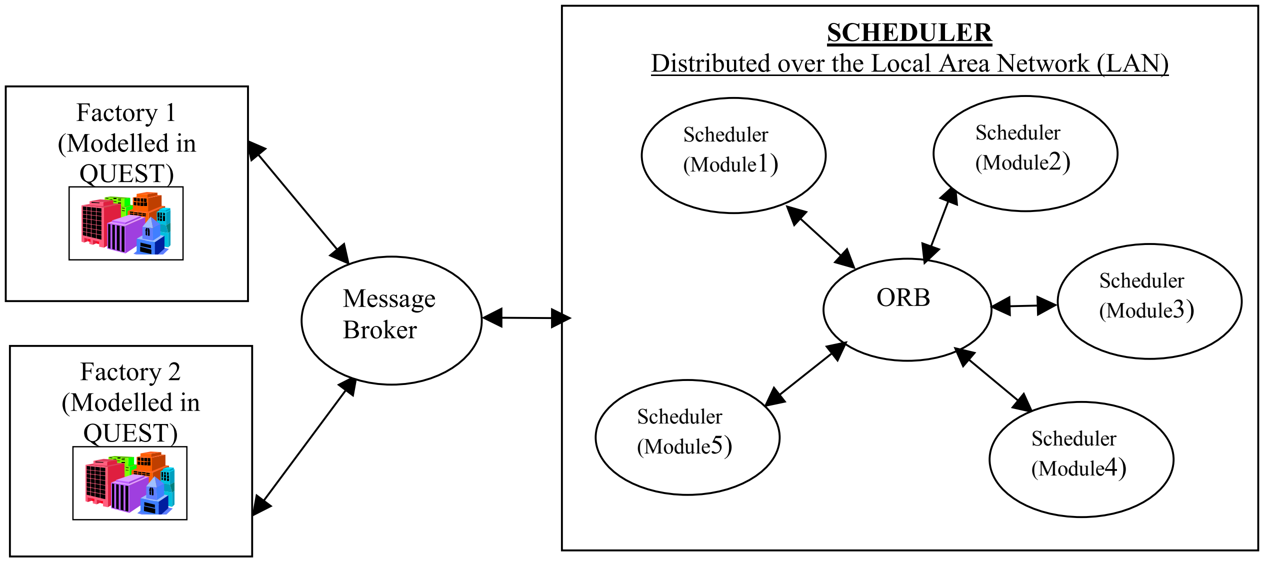

Figure 1: Distributed Architecture of Simulation Model and Scheduler

3.2 Scaleable

The Scheduler is easily scaleable to accommodate changes in the types and counts of resources as well as addition or deletion of part types to the existing portfolio. Scaling up simply requires more agents to be created while scaling down requires existing agents to be made invalid.

3.3 Reconfigureable

The Agent based architecture is quite conducive to absorbing on-going changes to work practices, a critical requirement for any typical manufacturing facility. Changes to existing process plans can be made quickly by changing the operating features of corresponding agents.

3.4 Adaptive

The Scheduler has specific algorithms that allow it to take corrective action when the situation demands. Such actions may be necessary whenever there are deviations from expected behaviour. Some examples would be

i) Primary or Secondary Resource Failure
ii) Part to be reworked/repaired after processing
iii) Part to be scrapped, alternate part required
iv) Worker absenteeism etc.

3.5 Input to Scheduler

The key inputs to the scheduler are

i) Process Plans
ii) Resource data
iii) Shift data
iv) Purchased Part data
v) Orders with required due dates

A typical Process Plan consists of a series of Process Steps that need to be completed in a certain order. Each Process Step in turn consists of a set of Resources and the time duration for each required resource. Now, most Process Steps do not state the exact resource required, more often it states a generic requirement. For example: Process Step 2 requires a Milling Machine, a Welder and a Fitter. There might be multiple instances of each type in a facility. Namely there may be 3 Welders, 3 Fitters and 3 Milling Machines to choose from.

Figure 2, illustrates a Process Plan with multiple Process Steps. Process Step 2 has been blown up to show its components. For this scenario, there can be potentially 27 (3×3×3) combinations of resources or "Resource Clusters" that can be used to perform Process Step 2.

The Scheduler picks the best "Resource Cluster" based on the costs quoted by each component of a cluster. The resources which have greater available capacity and hence lower costs are picked over other resources which are already booked and hence have higher costs. This leads to balancing out of work load between resources of similar skill set. Using this selection criteria, the scheduler picks the least costly cluster and awards the task to its component resources.

The functionality of the Scheduler can be divided into 2 categories

1) Task Allocation- Schedule Generation
2) Task Re-Allocation – Schedule Adaptation

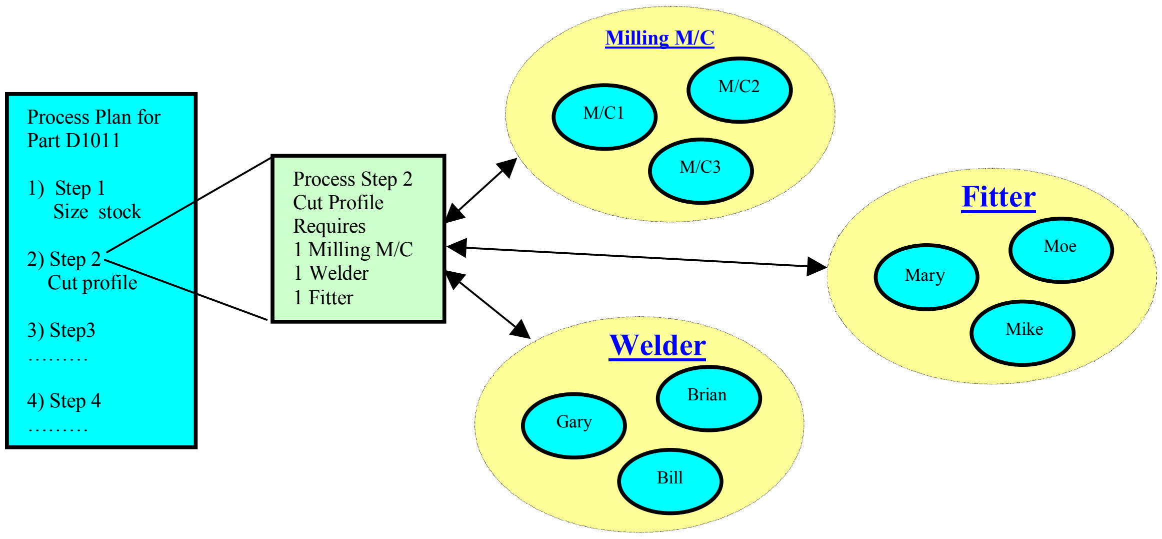

Figure 2: Details of a Typical Process Plan and its Component Resources

3.6 Task Allocation-Schedule Generation

This is the process of taking an input order, splitting it into its component tasks based on applicable process plans and then finding the most suitable candidates to perform the task. While the first task is fairly routine, it is the second task, namely the task of choosing the right candidates, that allows us to leverage advantages of Agent technology to the utmost.

This choice is done on an activity based cost approach. When an order comes into the Scheduler, a bid request is sent to all Agents that can potentially contribute to its processing. All of these Agents respond by quoting individual cost figures to do the task. The bid value that each Agent returns is based on the "level of committed capacity" it has at the time the new task is requested to be performed. Selection of the appropriate "Resource Cluster" can then be done. This kind of decision is usually based on the level of urgency of the order versus the amount of extra cost that would be incurred to get the order done fast. Algorithms falling under the "Task Allocation" create a schedule that is based on future available capacity.

3.7 Task Re-Allocation-Schedule Adaptation

While task allocation deals with the allotment of tasks to individual resources based on a predicted level of available capacity, task re-allocation is a reactive strategy that senses a problem in the schedule and tries to resolve it by re-allocating tasks. The need for this may arise whenever deviations are made from the predefined schedule because of factors like resource failure. The difference between these two approaches is that, task allocation is based more on an advanced perception of available capacity. No change to the schedule would be required if everything performed as predicted. But since unpredictable factors like resource failures, cycle time overruns etc. are endemic to any production facility, it is necessary to fine tune a schedule on a real time basis based on status updates from the shop floor. This real time update to a pre existing schedule is handled by protocols falling under the Task-Reallocation category.

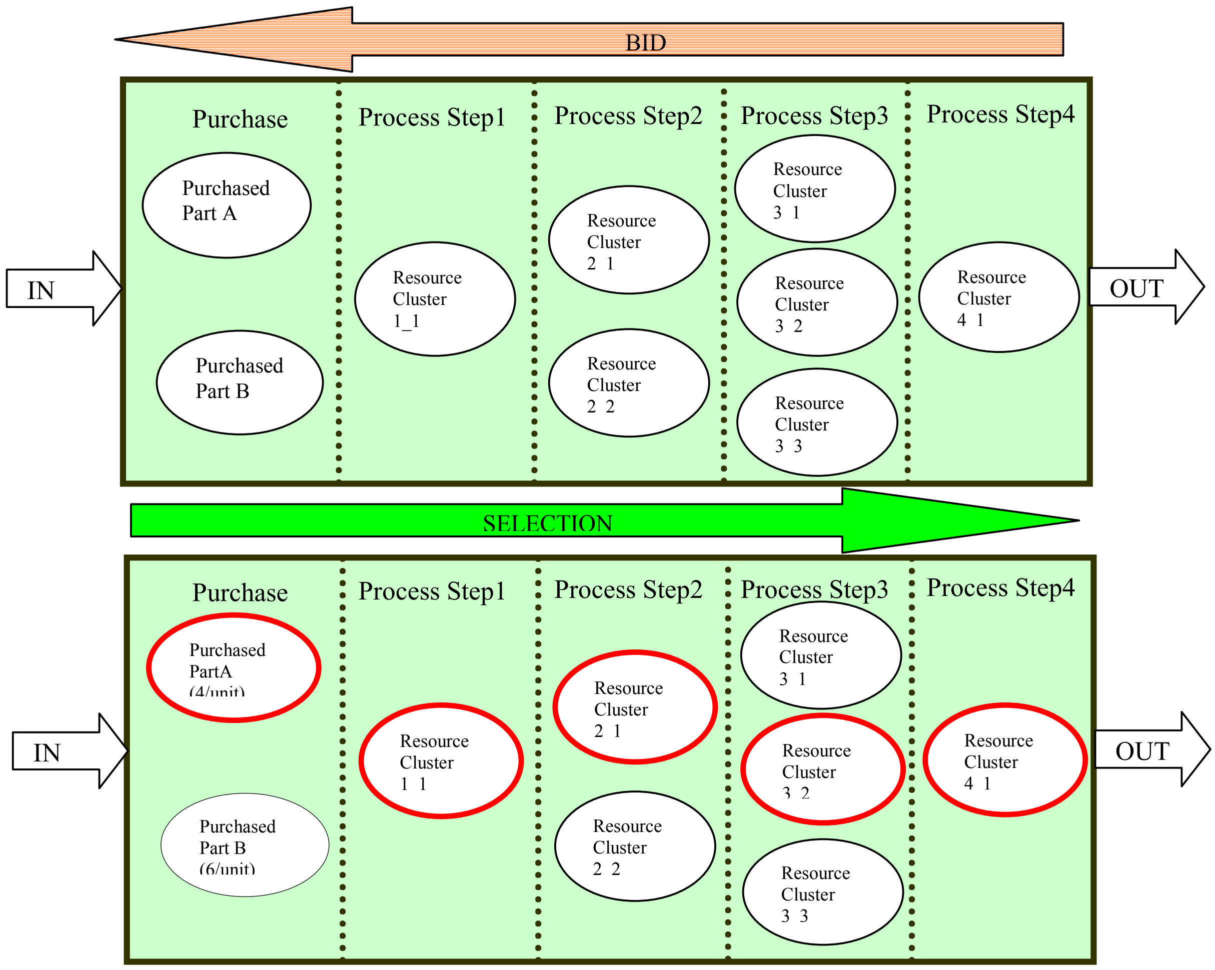

Figure 3: Bidding Process and Selection Resources

4 CONCLUSION

The solutions traditionally applied to scheduling have been either too computationally intensive or cumbersome because of centralized decision making. While such solutions have done a good job at "look forward schedulers i.e. scheduling for the next shift or the next day, they fall short when it comes to performing "adaptive scheduling", which is scheduling for a dynamically changing and unpredictable system on a real time basis.

We feel that the methodology and approach outlined here has inherent advantages that allow us to truly address and solve this problem. In summary this paper demonstrates

1) Agent based technology, the use of Agents in the field of Scheduling is unique and unprecedented.
2) It discusses a Scheduler developed in Java. As a result it has all the benefits of portability, scalability, distributed computing and ability to be run over a LAN or WAN.
3) It demonstrates a new application of simulation, i.e. application of discrete event simulation to validate dynamically changing schedules.
4) The discrete event simulation software linked to a scheduling system, versus the feedback from a live MES system, provides an excellent testing and development environment.

ACKNOWLEDGMENTS

This project is partially funded by the NIST ATP program awarded to DELMIA Corp. (Naresh Raja). This project was developed with participation from ERIM and other group members from DELMIA Corp.

REFERENCES

Sauter, J.A., Parunak, H.V.D, Goic, J. 1999. ANTS in the Supply Chain. Presented at the Workshop on Agents for Electronic Commerce at Agents'99, Seattle, WA, May (1-5).

Smith, S.F. 1994. OPIS: a methodology and architecture for reactive scheduling, in *Intelligent Scheduling*, M.Zweben and M.S. Fox, Eds. San Francisco, CA: Morgan Kaufmann Publishers, pp.29-66.

Walsh, W.E., Wellman, M.P., Wurman, P.R., and MacKie Mason J.R. 1998. Some economics of market-based distributed scheduling. In *Proceedings of 18th International Conference on Distributed Computing Systems*.

AUTHOR BIOGRAPHIES

SHUBHABRATA BISWAS is a Software Engineer working on the design and development of the Agent Network for Task Scheduling project at DELMIA Corp. He has over 6 years experience in the application of simulation to process improvements for major manufacturers like Ford, GM, DaimlerChrysler, Mercedes Benz and Delphi Corp. His other interests include Supply Chain Management, Object Oriented Design Concepts, B2B E-Commerce, Distributed Computing, Robotics and Intelligent Manufacturing. He has a B.S in Mechanical Engineering and an M.S in Industrial and Manufacturing Systems Engineering from the University of Windsor, Canada. His graduate research work was on "Dynamic Scheduling". He has held memberships at the SME chapter of Detroit and the CSIE. His email address is `<shubhabrata_biswas@delmia.com>`.

SARA MERCHAWI is a Senior Software Engineer and Product Manager of the Agent Network for Task Scheduling project at DELMIA Corp. She has a B.S and M.S in Computer Engineering and a Ph.D in Industrial Engineering from Penn State University. Her other interests include Agent Technology, Neural Networks and Artificial Intelligence. Her e-mail address is `<sara_merchawi@delmia.com>`.