*Proceedings of the 1989 Winter Simulation Conference*
E.A. MacNair, K.J. Musselman, P. Heidelberger (eds.)

# JUST-IN-TIME SIMULATION USING ARTIFICIAL INTELLIGENCE

S. Manivannan
Assistant Professor
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

## ABSTRACT

The shortcomings of process oriented simulation languages for modeling various types of kanbans in a Just-in-time (JIT) system are outlined in this paper. To circumvent these modeling difficulties, a rule-based simulator (JITSAI) has been designed and implemented using LISP/VM language on an IBM 4381 computer. The simulator includes efficient user interfaces, static and dynamic databases, primitives for each of the JIT modeling and simulation issues, a domain-specific rulebase interfaced with an event planner, an explanation subsystem and an inference mechanism. Two simulation experiments conducted on a JIT system to illustrate the current features of the JITSAI simulator are discussed.

## 1. INTRODUCTION

Just-in-time manufacturing (JIT) system embraces the concept of producing and/or stocking only the necessary items in necessary quantities at the necessary time. It is also known as a pull system wherein the subsequent production process goes to the preceding process to withdraw the necessary parts whenever there is a demand. Toyota production system discussed in Monden (1983) was designed based on this concept. The primary focus of this paper is limited to modeling kanban systems in a JIT environment. A survey on process interaction simulation languages for modeling JIT systems was recently carried out by Manivannan (1988). The results of the survey has indicated the following:

(a) There is no simulation language that provides the language constructs (blocks) explicitly for the purpose of modeling kanbans.
(b) All the available features of a language should be understood in a general context and then used to model a JIT system.
(c) The unfamiliarity of simulationists with both the techniques and their importance in statistical validity of the simulation output may cause misuse of simulation models as decision tools.
(d) Modeling a JIT system with conventional, general purpose languages such as SIMAN, SLAM, etc., may require making several changes in the model and experiment frames whenever there is a change in the level of abstraction.
(e) Building simulation models for complex and large manufacturing systems with conventional languages makes the models error prone; see Nance (1984).

Huang (1983), Schroer and Black (1985), and Lulu (1986) have developed simulation models for analyzing JIT systems with process interaction simulation (CSL) languages. Modeling JIT systems using a CSL is very cumbersome as the languages do not offer the blocks designed specifically for kanbans; see Pegden (1986). There are no explicit features for representing flow characteristics of different types of kanbans in a JIT system. Further, modeling interactions between kanbans and parts is very difficult using a CSL as the kanbans and parts move in different directions. User-written routines must be interfaced with a CSL to incorporate the rules for creation and movement of each of these kanban types. Hence, there is a need for a new modeling framework and a simulator capable of modeling kanbans in JIT systems. This conclusion has been further supported by the current trend in simulation software design to provide for automatic generation of simulation models through an efficient user interface; see Luker (1986).

## 2. A RULE-BASED SIMULATOR FOR MODELING JIT SYSTEMS

### 2.1. Requirements of Modeling Framework

The requirements of the modeling framework for conducting simulation of a JIT system are as follows:

(a) The modeling framework should offer capabilities to handle two types of entities (parts/kanbans) that flow in opposite directions.
(b) The modeling framework must provide a capability to easily represent various rules used in the creation and the movement of parts and kanbans.
(c) The modeling framework should be easier to understand, implement and use than that of CSL. It should facilitate implementing an integrated software environment which eliminates the need for a JIT user to write computer programs.
(d) The modeling framework should allow the knowledge and the rules designed for simulating various JIT systems to be flexible and distributed in a more natural way so that the simulation can be easily changed and extended.

## 2.2. Proposed Framework for JIT modeling

In order to satisfy the modeling requirements of a JIT system, a combination of rule orientation and event scheduling framework has been selected. The rule base contains rules designed for modeling only the production related issues in a JIT system. The rule base has been integrated with an event planner (time flow mechanism) to obtain the event scheduling framework.

## 2.3. Design and Implementation

Figure 1 indicates the block diagram of various components of the rule-based simulator implemented in LISP on an IBM 4381 computer. The simulator is named as JITSAI (Just-in-time Simulation with Artificial Intelligence). The detailed description of different components and implementation methods of software are provided in Manivannan [4]. The LISP program modules implemented as major components of JITSAI simulator are as follows:

(a) USER: An efficient user interface designed to obtain data related to modeling and simulation from the users of JIT systems.
(b) DATASTR: Various static and dynamic databases are implemeted to store data pertaining to modeling, simulation and experiment phases.
(c) RULEBASE: An associative list implemented in LISP that consists of a domain-specific rulebase. This consists of rules for modeling, configuring the simulation parameters, and running simulation.
(d) EP: An event planner implemented to work in conjunction with the rulebase in identifying the next event.
(e) PRIMIT: This module consists of various language constructs that are designed and implemented to represent the JIT system and simulation concepts.

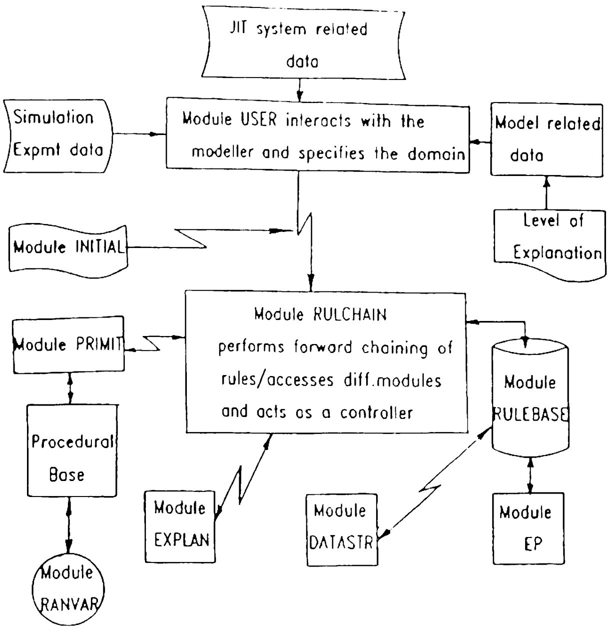

Figure 1. Block Diagram Showing the Various Components of JITSAI Simulator

(f) RULCHAIN: This module is the inference engine that consists of a rule chaining and control mechanism, and uses forward chaining principles.
(g) EXPLAN: An explanation subsystem implemented to provide simulation outputs, and explanations on various actions taken by the simulator.

The simulation models are created online by the computer using the factual information provided by the JIT user. For a given set of user inputs, only a portion of the rule base is invoked thereby depicting a specific JIT system. The rules in the rulebase are grouped into rulesets and each ruleset will invoke a procedure (primitive) stored in the procedural base. The data structures for the rules and rulesets are discussed in Manivannan (1988). The rules are linked by a forward chain inference mechanism. The dynamic databases are modified and updated whenever the rules in a ruleset are triggered and fired. This chaining process (simulation) will be repeated until a user-specified termination condition occurs. The system throughput, machine utilization, the number and type of kanbans at different workstations, and the queue related information are some of the results generated by the simulator.

## 3. SIMULATION EXPERIMENTS USING JITSAI

Two of the most commonly encountered production related issues are selected and modeled using the simulator. These issues are the (i) determination of kanban levels and (ii) effect of demand fluctuation on the performance of the JIT system. The simulation experiments conducted, results and the explanations generated by the simulator in each of these cases are discussed in this section.

### 3.1. Description of a JIT cell

To illustrate the current capabilities of JITSAI simulator, a fictitious manufacturing cell that uses JIT principles has been developed. The cell consists of two types of machining centers, each type has two individual machines. The cell produces and transports finished castings to an assembly (parent). The raw castings are supplied by a foundry (supplier). There are three types of castings produced by the cell whose process sequence and machining times are known. Two types of kanbans are used in the cell. These are withdrawal (WLK) and production order (POK) kanbans. The raw material is supplied by the foundry as and when the kanbans (RMK) are released. The layout of the machining cell is shown in Figure 2.

There is a material handler who goes around the cell and collects the kanbans and parts. The handler always moves around the cell in clockwise direction. The creation times for WLKs and POKs, speed of the handler, the distances between various machines, the demand process for the parts, process sequence and operation times are shown in Table 1. The material handler goes to the preceding machine only after a constant number of WLKs (known as COQ) is accumulated at the subsequent machine.

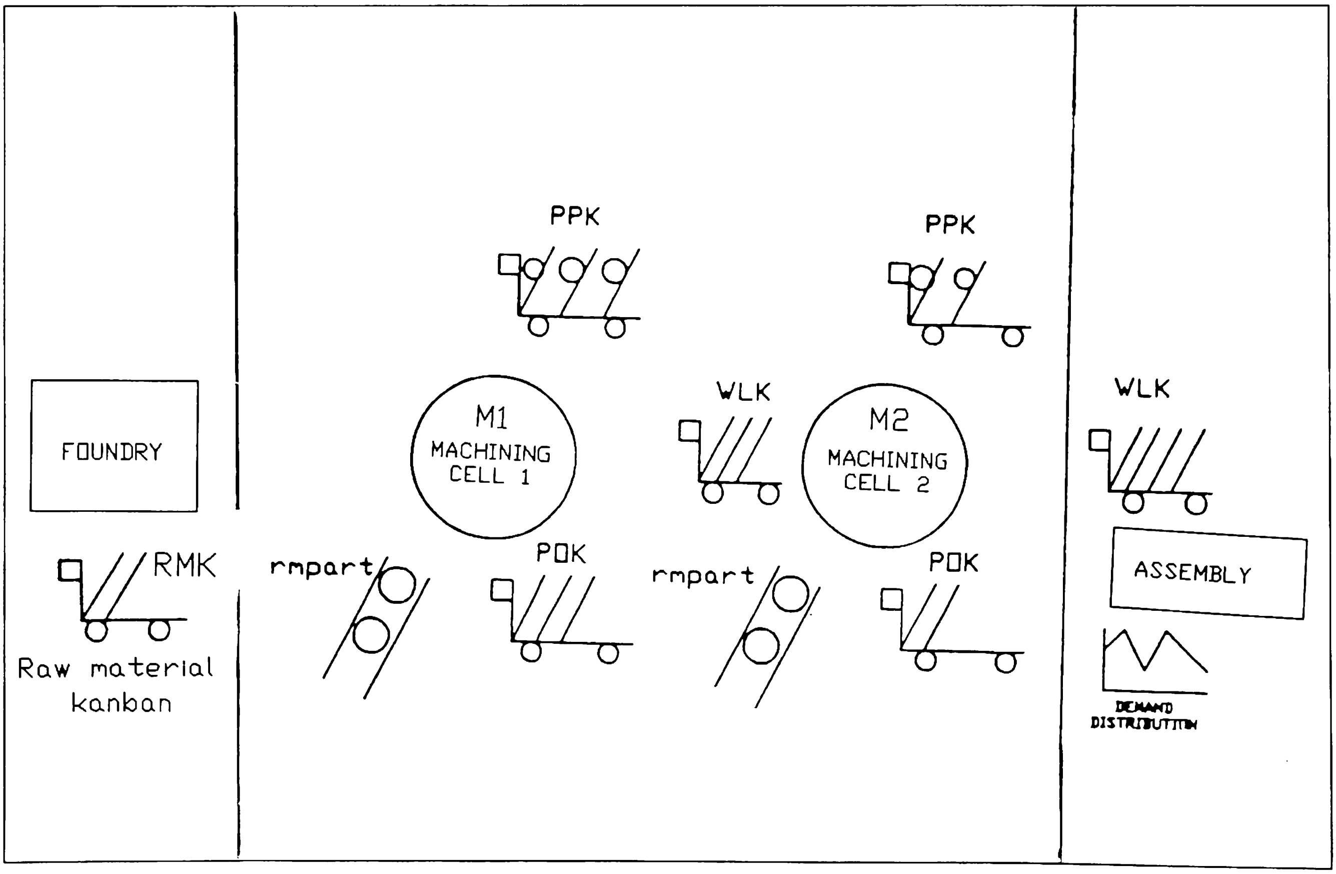

Figure 2. Layout of a JIT Cell

Table 1. Data Inputs for Case(i)

Number of parttypes = 3
Number of machinetypes = 2
Number of kanbans = 2
Kanban types = Withdrawal(WLK) and Production order (POK)
Creation time for WLK at each stage = 2 time units
Creation time for POK at each stage = 1 time units
Movement time between two adjacent stations = 1 time unit

Assumptions:

(i) Demand is known, continuous and it is random
Exponential arrival with mean = 10 and with part types
having the following proportions:
P1=0.3, P2=0.4, P3=0.3
(ii) No setups or breakdowns are considered
(iii) The number of kanbans are determined using Constant
order quantity (COQ) method. The parameters are given
by the user before simulation.
(iv) Number of machines in each machine type = 2

Process Sequence

| | | |
|---|---|---|
| Part 1 | M1 | M2 |
| Part 2 | M2 | M1 |
| Part 3 | M1 | M2 |

Process time

| part \ m/c | M1 | M2 |
|---|---|---|
| P1 | 10 | 12 |
| P2 | 11 | 10 |
| P3 | 10 | 9 |

### 3.2. Modeling and Verification

To model various production related issues using the JITSAI simulator, JIT system related data (shown in Table 1) have been entered with the user interface (USER module). Once the input phase is complete, the rulebase will be invoked and various rulesets needed to simulate the specified JIT system will be chosen. These rulesets and the input data will then be stored in a temporary dynamic database. This is very useful when the user of a JIT system wants to either perform several experiments on the same model or modify the model parameters between the simulation runs. Further these changes are stored each time a change is made and there is no need to keep track of them.

The model verification can be performed using the explanation system (level 1). The EXPLAN module is designed to provide three levels of explanations. Level (1) gives a complete trace for each simulation run. This trace consists of various information such as the current time, the rule triggered in a ruleset, the content of the rule fired, the next ruleset that will be invoked, results of rule evaluation, contents of the queues and the event planner. Once the model verification is complete, the simulation experiments can be performed using the JITSAI simulator by simply modifying the values of the model and experiment parameters. Level (2) provides the current time, and rules fired to perform the rule adequacy. Level (3) provides the final results of each simulation run.

### 3.3. Case(i): Determination of Kanban levels

The purpose of modeling the above JIT system and running experiments in this case is to determine the preferred WLK level (COQ) at each of the machines such that the throughput and the machine utilization are maximized, and the buffer size is minimized. The buffer size is defined as the number of part/POK pairs (PPK) waiting to be withdrawn from the subsequent machines.

Simulation experiments have been performed for 360 time units. The initial COQ value at each of the machines has been kept equal. The COQ value refers to the number of withdrawal kanbans to be accumulated before the material handler checks for finished parts at the preceding machine. The results of simulation include throughput, average machine utilization, and buffer size (maximum number in the PPK queue). Simulation runs were repeated for different COQ values and the simulation results from each run are indicated in Table 2.

Table 2. Case(i) Simulation Results (Fixed COQ)

| Fixed COQ/ stage (Assy., M1, M2) | Total Produced | | | | Ave. Util. | | Buffer Size | |
|---|---|---|---|---|---|---|---|---|
| | P1 | P2 | P3 | Total | M1 | M2 | M1 | M2 |
| (1, 1, 1) | 4 | 7 | 3 | 14 | 0.177 | 0.165 | 2 | 4 |
| (2, 2, 2) | 7 | 11 | 9 | 27 | 0.555 | 0.495 | 5 | 3 |
| (3, 3, 3) | 6 | 9 | 9 | 24 | 0.585 | 0.523 | 6 | 3 |
| (4, 4, 4) | 6 | 7 | 7 | 20 | 0.520 | 0.441 | 6 | 3 |
| (5, 5, 5) | 4 | 7 | 5 | 16 | 0.480 | 0.391 | 6 | 3 |

The results are plotted and indicated in Figures 3 to 6. Figure 3 indicates the total number of parts for each part type and the total system throughput for various COQ values.

From the graph in Figure 3, it can be infered that the throughput keeps increasing up to a certain value and then starts to decrease. The maximum throughput occurs when the COQ value equals two. This means that whenever there are two withdrawal kanbans for a part type at any stage, the material handler goes to the previous stage to check for a finished part. The increase in throughput for higher COQ values is due to the fact that there is more time available for machines to produce more parts (as the frequency of withdrawal is low). But this increases the buffer size. The increase in buffer size for an increase in COQ value can be seen in Figure 4. The decrease in the throughput beyond COQ values greater than two is due to diminishing returns. The same is true for the average machine utilization values.

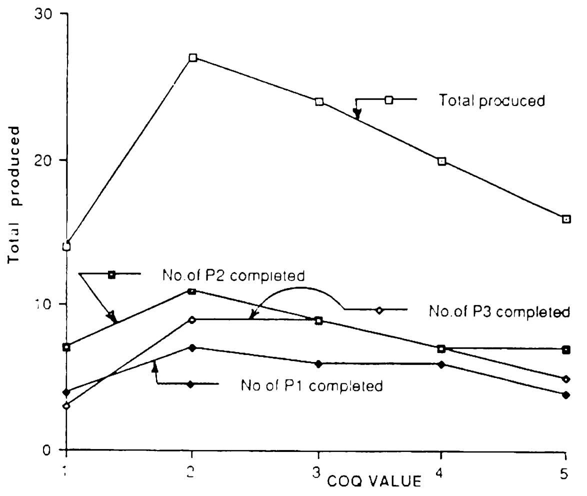

Figure 3. Fixed COQ vs Parts Produced

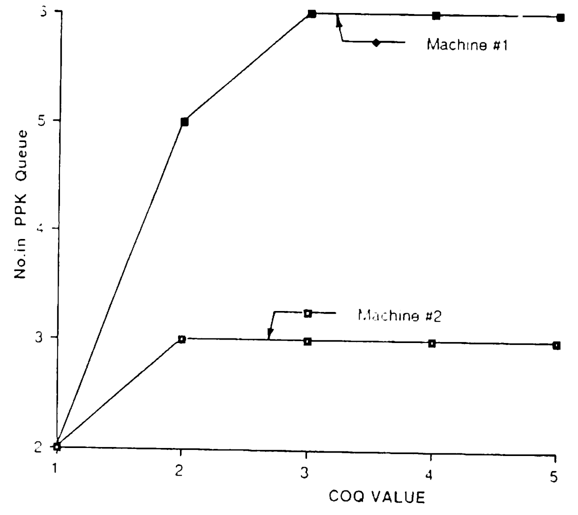

Figure 4. Fixed COQ vs No.in PPK Queue

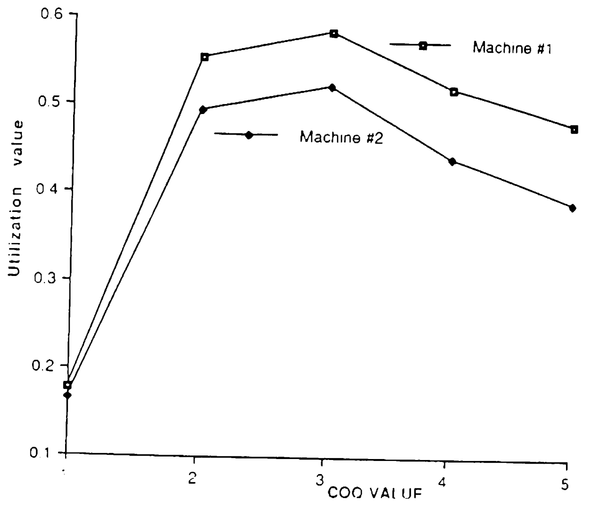

Figure 5. Fixed COQ vs Machine Utilization

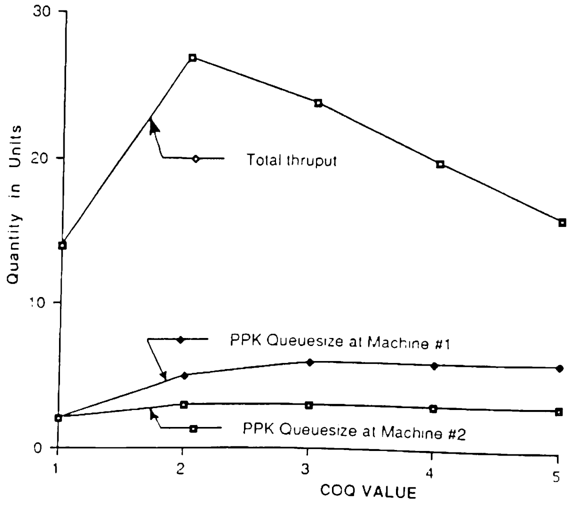

Figure 6. COQ vs Throughput and PPK Queuesize

Figure 5 indicates a graph showing the machine utilization for different COQ values. The average utilization increases with increase in the COQ value upto a certain level and then starts to decrease. Hence, these graphs show that the best preferred COQ value is two for each machine stage. Figure 6 shows a graph with total throughput and the buffer size at the machines M1 and M2. From this graph, the maximum throughput occurs at a COQ value of two. But the buffer size at M1 is higher. The selected COQ values are summarized (without the final stage) as follows:

COQ (M1) = 1; COQ (M2) = 2

To study the effects of withdrawals made at the final stage on the total throughput, machine utilization and the buffer size, another series of simulation experiments was conducted. In this study, for a pre-specified COQ value at the final stage (assembly), the COQ values for M1 and M2 were varied. Simulation has been performed in each case and the results are indicated in Table 3.

Table 3. Case(i) Simulation Results (Variable COQ)

| COQ Value at the Assembly | COQ(M1,M2) | Total Parts Completed | Aver. Mach. Utilization | | Total Buffer Size |
|---|---|---|---|---|---|
| | | | M1 | M2 | |
| 1 | (1, 1) | 14 | 0.177 | 0.165 | 6 |
| | (2, 2) | 18 | 0.333 | 0.234 | 7 |
| | (4, 4) | 9 | 0.278 | 0.227 | 10 |
| 2 | (1, 1) | 19 | 0.394 | 0.354 | 6 |
| | (2, 2) | 21 | 0.555 | 0.495 | 8 |
| | (4, 4) | 24 | 0.603 | 0.513 | 8 |
| 3 | (1, 1) | 16 | 0.440 | 0.372 | 8 |
| | (2, 2) | 18 | 0.520 | 0.441 | 9 |
| | (4, 4) | 20 | 0.500 | 0.422 | 8 |

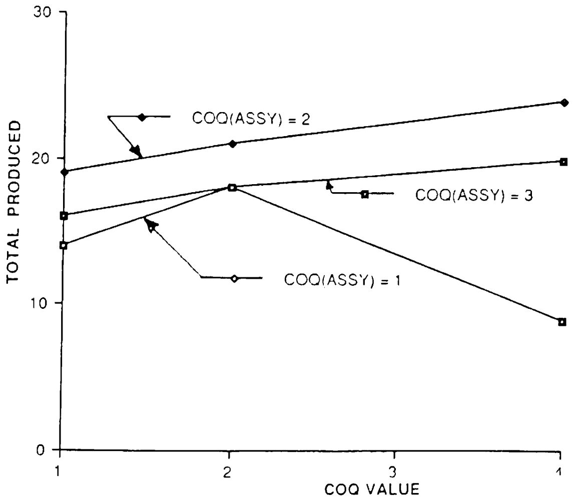

Figure 7. Variable COQ vs Total throughput

Figure 7 shows the plots on system throughput for different COQ (assembly) values. From this graph, it is inferred that the maximum values occurred at a COQ value of two for machines M1 and M2 (as before). At lower COQ (assembly) levels, the throughput is less and starts to increase as the COQ (assembly) value increases. However, the throughput decreases beyond a certain increase in COQ (assembly) value. The reason for this trend is that the COQ value at the final stage determines the demand rate. As this rate increases, the buffer sizes and the machine utilization increase to a certain extent (leading to increase in throughput) but then start to fall beyond a certain increase. Hence, the best value for COQ (assembly) from the simulation results is two. Hence, the COQ values for different stages are summarized (including the final stage) as follows:

COQ(assembly) = 2; COQ(M1) = 1; COQ(M2) = 2

This introduces variable COQ levels (levels vary from one stage to the other). In order to determine the performance of the JIT system with the selected COQ levels, a simulation run has been made and the results are indicated in Table 4. From this table, it can be seen that the buffer size has gone down, the throughput and the machine utilization have gone up.

Table 4. Simulation Results for Best COQ values

COQ VALUE: Assembly = 2
M1 = 1
M2 = 2

| | | |
|---|---|---|
| Throughput | ☒ | 25 |
| M/c Utilization | M1 | 0.534 |
| | M2 | 0.434 |
| Buffer Size | M1 | 4 |
| | M2 | 3 |

## 3.4. Case(ii): Effect of demand fluctuations

The purpose of modeling and simulation of a JIT system in this case is to demonstrate other features of the JITSAI simulator. Simulation experiments were conducted to study the effects of fluctuations in demand on the performance of the JIT system. The JIT system described in Section 3.1 has been utilized for this purpose. The total number of types of machining centers in the JIT cell has been increased to three. Table 5 provides various data inputs pertaining to to this new model.

Table 5. Data Inputs for Case(ii)

Number of parttypes = 3
Number of machinetypes = 3
Number of kanbans = 2
Kanban types = Withdrawal(WLK) and Production order (POK)
Creation time for WLK at each stage = 3 time units
Creation time for POK at each stage = 2 time units
Movement time between two adjacent stations = 2 time units

Assumptions:

(i) Demand is known, continuous and it is random Exponential arrivals:
Case (i): inter-arrival time = 12 units
Case (ii):inter-arrival time = 8 units
and the part types have the following proportions:
P1=0.1, P2=0.5, P3=0.4
(ii) No setups or breakdowns are considered
(iii) The number of kanbans are determined using Constant order quantity (COQ) method. The parameter values varied by the user before simulation.
(iv) Number of machines in each machine type = 2

Process Sequence

| | | | |
|---|---|---|---|
| Part 1 | M1 | M2 | M3 |
| Part 2 | M3 | M1 | M2 |
| Part 3 | M1 | M3 | M2 |

Process time

| part \ m/c | M1 | M2 | M3 |
|---|---|---|---|
| P1 | 11 | 15 | 13 |
| P2 | 12 | 13 | 14 |
| P3 | 9 | 12 | 15 |

The JIT system has been modeled and simulated for 360 time units. Two different demand rates have been used to study the effects of demand variations. Initially, the demand distribution has been assumed to be exponential with a mean value of 12. The COQ values have been varied and in each case, the outputs on the throughput, average machine utilization, and number of parts in the PPK queue (buffer size) are collected. These results are provided in Table 6.

Table 6. Case(ii) Simulation Results (a1 = 12)

| Fixed COQ/ machine (Ass., M1, M2, M3) | Total Produced | Average Utilization | | | Total Buffer Size |
|---|---|---|---|---|---|
| | | M1 | M2 | M3 | |
| (1, 1, 1, 1) | 10 | 0.188 | 0.078 | 0.174 | 1 |
| (2, 2, 2, 2) | 15 | 0.580 | 0.546 | 0.548 | 9 |
| (3, 3, 3, 3) | 13 | 0.556 | 0.560 | 0.460 | 12 |
| (4, 4, 4, 4) | 12 | 0.484 | 0.294 | 0.494 | 20 |

Another set of simulation experiments has been conducted on the same JIT system with a change in the demand rate. The mean value is changed to 8 (higher demand rate). The COQ values were varied as before, and outputs from the simulation runs are collected and shown in Table 7.

Table 7. Case(ii) Simulation Results (a2 = 8)

| Fixed COQC (Ass., M1, M2,M3) | Total Produced | Average Utilization | | | Buffer Size |
|---|---|---|---|---|---|
| | | M1 | M2 | M3 | |
| (1, 1, 1, 1) | 17 | 0.436 | 0.220 | 0.540 | 10 |
| (2, 2, 2, 2) | 38 | 0.800 | 0.900 | 0.870 | 13 |
| (3, 3, 3, 3) | 42 | 0.930 | 0.950 | 0.920 | 12 |
| (4, 4, 4, 4) | 39 | 0.910 | 0.890 | 0.913 | 17 |

The simulation results are plotted for both the demand rates (a1=8) and (a2=12). The throughput is found to increase as the demand rate is increased (Figure 8).

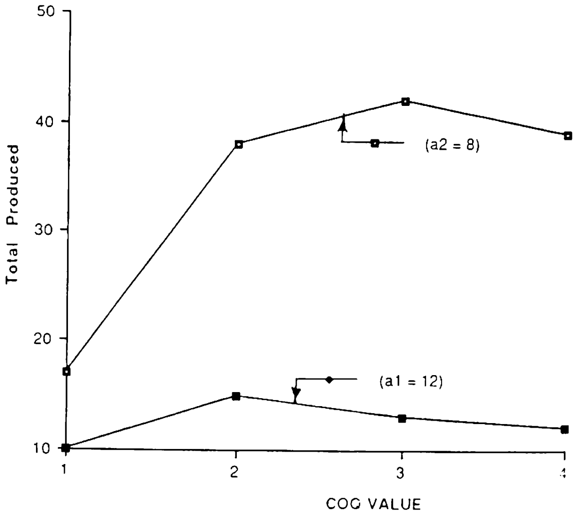

Figure 8. Fixed COQ vs Throughput for a1=8, a2=12

Likewise, the average machine utilization has increased as the demand rate is increased as seen in Figure 9. By using average machine utilization as one of the performance measures, the upper and lower bounds for the demand rates are set for the given JIT system. It should be mentioned here that the purpose of these experiments is to demonstrate the use of the simulator. To study the effects of demand variations in a JIT system, a detailed statistical analysis is required to find the best preferred demand rate.

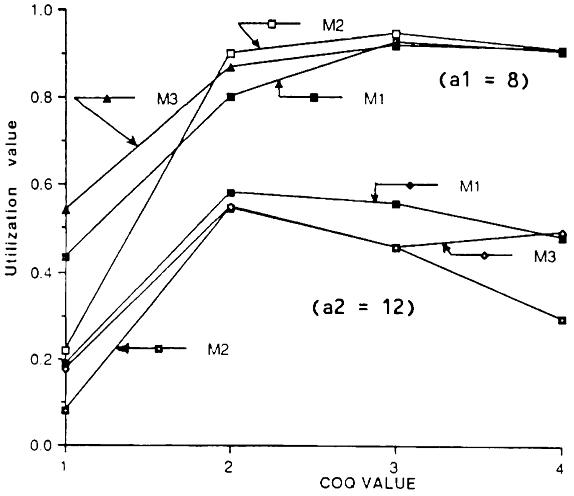

Figure 9. Fixed COQ vs Machine Utilization for a1 = 8 and a2 = 12

## 4. FUTURE RESEARCH

The current capabilities of JITSAI simulator are very limited for modeling and analysis of various JIT systems. This is due to the number of rules and rulesets available in the rulebase of the simulator. Future research will be directed toward extending the rulebase as follows:

(a) At present, the rulebase is restricted to mainly WLK, POK and emergency kanbans (for machine setup and breakdown). More rules will be developed to handle supplier kanbans, express kanbans, quality control related kanbans along with several others.
(b) There is a need for automating the process of adding new rules to the rulebase. Also, when the new rule is added, there is a need for a test to be performed to check the consistency and rule adequacy before using the rulebase.
(c) Rules are restricted to only the COQ method to determine the frequency of withdrawal. Also, the user needs to input these values, and there is no on-line determination of COQ values based on lead time and waiting time.
(d) An output processor for conducting statistical analysis on the simulation results needs to be implemented.
(e) Results of simulation are restricted to system throughput, average machine utilization and queue statistics. Additional rules and procedures will be developed to provide results on various other performance measures related to JIT systems.

## 5. CONCLUSIONS

The main advantages of the new JITSAI simulator are as follows:

(a) The new rule-based simulator helps the user to model JIT systems without learning the intricate details of the simulation software.
(b) It provides an integrated simulation environment to the user for modeling kanban systems and other production related issues such as machine set-up, break-downs and inventory issues in a JIT system.
(c) The process of simulation model creation and program generation phases are automated.
(d) Integration of the rulebase with an event planner reduces CPU times considerably to run simulations as compared to many other rule-based simulators.

In this paper, the difficulties of modeling a JIT system with process oriented simulation software were outlined. A new modeling framework and a rule-based simulator for modeling production related problems in JIT systems were discussed. The JITSAI simulator was utilized to model and simulate several JIT systems and two case studies were presented.

## REFERENCES

Huang, P.Y., Rees, P.L., and B. W. Taylor. "The Simulation Analysis of Japanese JIT for a multi-machine, multi-stage Production System", Decision Sciences, Vol.14, 1983

Luker, P.A., and H.H. Adelsberger. "The Intelligent Simulation Environments," Society for Computer Simulation, Simulation Series, Vol.17, No.1, January 1986

Lulu, M. "Just-in-time (JIT) production and Process Unreliability," Annual Simulation Symposium, 1986

Manivannan, S., and C.D. Pegden. "A Rule-oriented Simulator for Modeling Just-in-time Manufacturing Systems," Submitted to Simulation, 1989

Manivannan, S. "A new rule-oriented simulator for modeling Just-in-time manufacturing systems," Ph.D dissertation, The Penn. State University, PA 16802, August 1988

Monden, Y. "Toyota Production System," Institute of Industrial Engineering Press, 1983

Nance, R.E. "Model Development Revisited," Proc. of Winter Simulation Conference, November 1984

Pegden,C.D. "Introduction to SIMAN," Systems Modeling Corporation, State College, PA 16801, 1986

Pritsker, A.A.B. "Introduction to Simulation and SLAM II," Third Edition, John Wiley & Sons, New York, 1986

Schroer, B.J., Black, J.T and S.X. Zhang. "Just-in-time (JIT) with kanban, Manufacturing System Simulation on Micro-computers," Simulation, 1985

Shannon, R.E., and Mayer, R and H.H. Adelsberger. "Expert Systems and Simulation," Simulation, 1985

## AUTHOR'S'BIOGRAPHY

S. Manivannan is an Assistant Professor in School of Industrial and Systems Engineering, Georgia Institute of Technology. He received his Ph.D in Industrial and Management Systems Engineering from Pennsylvania State University. His current research interests include Intelligent Manfacturing Systems, Knowledge-based Simulation and Database design. He is involved in extending two of his recently developed knowledge-based systems capable of (i) simulating JIT systems, and (ii) incorporating the manufacturing tolerance information into a computer aided process planning system. He is a member of Alpha Pi Mu, IIE, ORSA/TIMS, and SME.