Proceedings of the 1990 Winter Simulation Conference
Osman Balci, Randall P. Sadowski, Richard E. Nance (eds.)

USING THE MAXX SIMULATOR FOR ELECTRONIC CARD MANUFACTURING

Stephen Buckley
Chae An
James Korein

IBM T. J. Watson Research Center
P.O. Box 218
Yorktown Heights, New York 10598

Darcy Palmer

IBM Almaden Research Center
650 Harry Road
San Jose, California 95120-6099

Javad Ahmadi
Peng Si Ow

IBM
11400 Burnet Road
Austin, Texas 78758

Urban Brunner

ETH-Zentrum
Swiss Federal Institute of Technology
CH-8092, Zurich, Switzerland

ABSTRACT

MAXX is a manufacturing oriented simulator designed for detailed modelling and analysis of a manufacturing line. It is capable of evaluating both steady state long term performance and transient performance based on short term production schedules for many types of manufacturing systems. It requires no programming by a user. Instead, users interact with MAXX through menus, tables, forms, and a graphical routing editor. In this paper, we discuss the use of MAXX for electronic card manufacturing.

1. INTRODUCTION

In the highly competitive electronics industry, it is crucial for manufacturing to react quickly to changing requirements. It is important to achieve short manufacturing lead times while making efficient use of resources such as machines and people. Since product life cycles are relatively short, a plant will almost always have a number of products being introduced or phased out. To cope effectively in such an environment, careful design of the manufacturing line is required. To ensure efficient day-to-day operation of the line, a plant needs to implement coherent rules for scheduling production resources.

In a typical card manufacturing plant, the number of products is large and production volumes vary widely from week to week and month to month. Demand forecasts are fairly unreliable and product life cycles are short (typically 1 to 2 years). Many of the machines require frequent setup changes (changes in machine configuration) when switching from one product to another. Operator skills vary and there may be a shortage of operators depending on the product mix as some products may require larger number of operators with particular skills. Electronic component shortages occur frequently and may halt production in midstream.

Card manufacturing plants differ in their plant control philosophies. Some are traditional "push" plants, where production is triggered by releasing a shop order to the beginning of the line. Others follow a "pull" philosophy, also known as just-in-time, where production is triggered by a requirement at shipping for some quantity of a particular product and fixed capacity buffers are used to control the flow throughout the line. Some lines have rules for deciding which and how many to make when multiple products are available.

Extensive tests are conducted at the end of the line. At this stage, defective cards may be sent back into the line for rework. Cards may also be repaired at designated repair stations and returned to the line to continue normal processing.

The characteristics described above place a large number of requirements on a simulator. The MAXX Version 4 simulator is designed to allow manufacturing engineers to investigate these types of issues without programming. MAXX is based on RESQ (Research Queueing Package) [Chow et al. 1985; MacNair 1985], a general purpose queueing package. MAXX Version 4 adds key enhancements to earlier versions of MAXX which were primarily intended for evaluating the steady-state performance of a manufacturing line.

The MAXX Version 4 user interface consists of menus, tables, forms, and a graphical routing editor. The model data are stored in a database. Once a model has been entered, a translator program automatically creates a RESQ input program. The actual simulation is then performed by the RESQ system.

In Section 2 of this paper, we describe how the above requirements are addressed by MAXX and how MAXX can be used to model an electronic card manufacturing line. In Section 3, we present some simulation results for an example model. We conclude with a summary in Section 4. A companion article [An et al. 1990] presents additional information about MAXX.

2. MODELLING AN ELECTRONIC CARD MANUFACTURING PLANT WITH MAXX

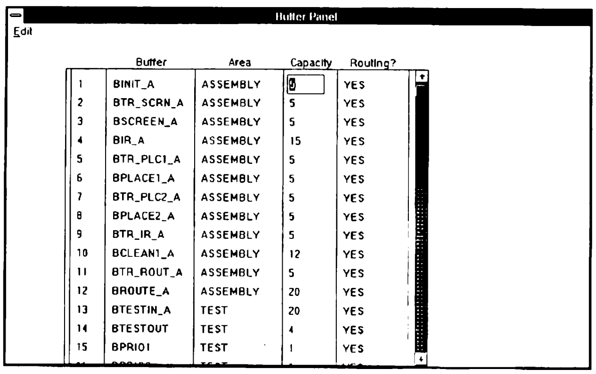
This section describes how we used MAXX to build a simulation model of a fictitious card manufacturing plant. Although the names and parameters used in this example do not represent an actual plant, the model resembles a typical card plant. For the sake of brevity, we will only be able to describe a subset of the model features. Where appropriate, we include screen images for some of the MAXX panels to illustrate the model building process.

2.1 Basic Line Configuration Parameters

The line configuration parameters that are common in card plants include products, areas, sources, shipments, buffers, machines, and routings. For our purposes, the word product refers to card type. As discussed earlier, an electronic card assembly line may handle several different products, each of which may require different machine setups, processing times, routings, and schedules. For this example, we model two products, A and B. For both of these products, raw printed circuit boards come into the line through the sources, components are assembled onto the boards, the boards are tested, and then they exit at the shipments.

Typically, a manufacturing plant is divided into areas. An area is a logical grouping of machines, buffers, sources and shipments. For the example card plant, there are three areas: ASSEMBLY, TEST, and SHIPPING.

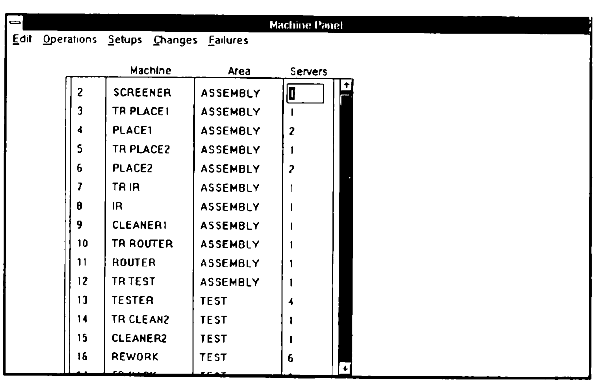
Figure 1 shows the buffer panel for the example card plant. Buffers are where work in process (WIP) can be stored between operations. For our example, we are modelling pull system rules for operating the line. Therefore, buffers with user specified capacities are used to control the flow of products through the line. If a buffer becomes full, it blocks the preceding operation. A push system can be modelled by specifying very large buffers, so that blocking does not occur.



	Buffer	Area	Capacity	Routing?
1	BINIT_A	ASSEMBLY	0	YES
2	BTR_SCRN_A	ASSEMBLY	5	YES
3	BSCREEN_A	ASSEMBLY	5	YES
4	BIR_A	ASSEMBLY	15	YES
5	BTR_PLC1_A	ASSEMBLY	5	YES
6	BPLACE1_A	ASSEMBLY	5	YES
7	BTR_PLC2_A	ASSEMBLY	5	YES
8	BPLACE2_A	ASSEMBLY	5	YES
9	BTR_IR_A	ASSEMBLY	5	YES
10	BCLEAN1_A	ASSEMBLY	12	YES
11	BTR_ROUT_A	ASSEMBLY	5	YES
12	BROUTE_A	ASSEMBLY	20	YES
13	BTESTIN_A	TEST	20	YES
14	BTESTOUT	TEST	4	YES
15	BPRIO1	TEST	1	YES

Figure 1. Buffer Panel

Figure 2 shows the machine panel for the example card plant. For each machine, the user can input information about operations that are performed on the machine, machine failures, setups on the machine, and times to change between setups. Figure 3 shows the operation panel for the SCREENER machine where solder paste is applied to each card.



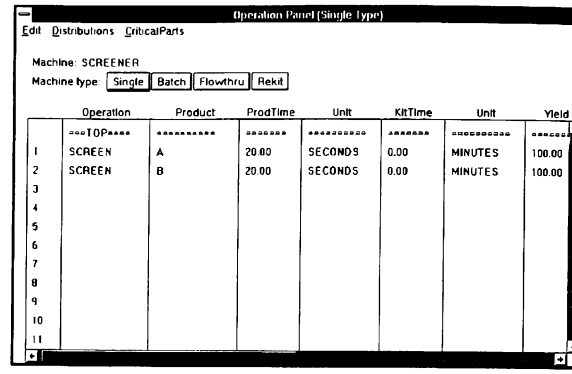
	Machine	Area	Servers
2	SCREENER	ASSEMBLY	0
3	TR PLACE1	ASSEMBLY	1
4	PLACE1	ASSEMBLY	2
5	TR PLACE2	ASSEMBLY	1
6	PLACE2	ASSEMBLY	2
7	TR IR	ASSEMBLY	1
8	IR	ASSEMBLY	1
9	CLEANER1	ASSEMBLY	1
10	TR ROUTER	ASSEMBLY	1
11	ROUTER	ASSEMBLY	1
12	TR TEST	ASSEMBLY	1
13	TESTER	TEST	4
14	TR CLEAN2	TEST	1
15	CLEANER2	TEST	1
16	REWORK	TEST	6

Figure 2. Machine Panel

The operation panel in Figure 3 shows that the SCREEN operation is performed on a single type machine. In card plants, there are four types of machines:

Single Cards are processed one at a time. A typical single type machine is a machine which performs electronic tests on a card.

Batch Kits of cards are accumulated, then processed simultaneously. A transport vehicle can be an example of a batch machine if several kits are collected at the vehicle before it is moved to the next operation.



	Operation	Product	ProdTime	Unit	KitTime	Unit	Yield
	===TOP====						
1	SCREEN	A	20.00	SECONDS	0.00	MINUTES	100.00
2	SCREEN	B	20.00	SECONDS	0.00	MINUTES	100.00
3							
4							
5							
6							
7							
8							
9							
10							
11							

Figure 3. Operation Panel

Flowthru Kits flow through in a pipelined fashion. Several kits can occupy a flowthru machine at the same time. One example is a cleaner in which there is a conveyor transporting boards through a solution bath.

Rekit The size of kits is changed by the machine. One example of a rekit machine is a machine which cuts a printed circuit board into several individual cards.

Note that a kit can represent different things. In the ASSEMBLY area of this example, a kit represents one printed circuit board. After each board has been cut into four cards, a kit represents one card.

Sources are where raw printed circuit boards come into the line and shipments are where the finished products exit the line. They can be scheduled periodically or according to a calendar schedule. A source can also be modelled as Always Available, as in this model, so that the production is triggered by shipping requirements according to pull system rules.

A routing depicts the logical network of sources, operations, buffers, and shipments that kits must go through. The routings for the ASSEMBLY, TEST, and SHIPPING areas in the example model are shown in Figures 4-6 respectively. Rectangles in the routings represent operations and octagons represent buffers. For an operation, the top row gives the name of the machine, the middle row gives the name of the operation and the bottom row gives the name of the product. Sources and shipments are represented by six sided polygons, with the narrow edge on the right side for sources and on the left side for shipments.

In the ASSEMBLY area each kit passes through a sequence of operations and buffers without branching. Transport operations, indicated by the name TRANSFER, move batches of five boards from one place to another in 60 seconds. Boards, each containing four cards, enter at SOURCE_A. They are first transported to SCREENER, a single type machine where solder paste is applied to each board. They are then transported to PLACE1, a component placement machine, and then to PLACE2, another component placement machine. Both of these component placement machines are single type machines, but they place different types of components. Next, the boards are transported to IR, a flowthru type machine where the solder is reflowed. After being cleaned by the flowthru machine CLEANER1, the boards are transported to ROUTER, a rekit type machine where each board is cut into four cards.

The first machine in the TEST area is TESTER, which performs electronic tests on each card, one at a time. Following TESTER, the routing contains a rework loop. In the

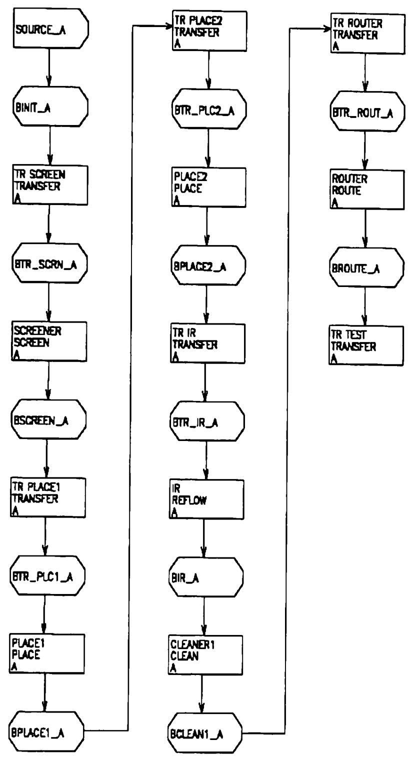


Figure 4. Routing for the ASSEMBLY Area for Product A

rework loop, branching conditions control the number of rework passes in two ways:

1. Branching conditions determine the rework level of each card by checking the card's priority. Since card priorities start at 1 at the source and are incremented by 1 at the rework stations, each card will exit after at most three passes. In this model, all cards are assumed to be good after three rework passes.
2. Branching conditions determine the percentage of cards that fail the test and require rework. Note that the failure percentage depends on the rework level.

The TESTER machine works on higher priority cards first. This avoids deadlocks that can occur when one of the rework loops reaches capacity. Following rework, cards are cleaned by CLEANER2, a flowthru machine. Cards which have not been reworked do not need to be cleaned.

Cards which have completed testing and cleaning are transported to the PACK machine in the SHIPPING area, where they are packaged and sent to the shipment SHIP_A. At SHIP_A, the cards wait to be shipped from the plant according to the shipment schedule.

The buffer capacities for this example are set to the minimum levels possible to minimize WIP and cycle time. For example, the transport vehicle TR_SCREEN in the ASSEMBLY area is a batch type machine which transports five boards at a time. The buffer capacity before it must be at least five so that the batch machine can proceed. The buffer capacity following it must be at least five or else the vehicle will not be able to move due to pull system rules. To minimize WIP and cycle time, the buffer capacities before and after TR_SCREEN are both set to five.

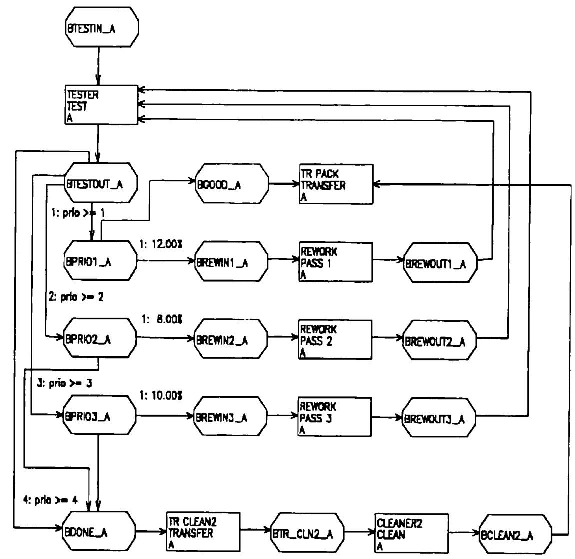


Figure 5. Routing for the TEST Area for Product A

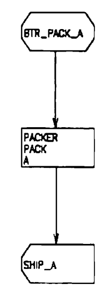


Figure 6. Routing for the SHIPPING Area for Product A

2.2 Modelling Control Policies

As discussed above, several products often share the same line in a card plant. In such an environment, machines must employ local control policies for choosing the next kit when they become idle. MAXX Version 4 offers several possible local control policies. They are:

1. FCFS (also known as FIFO) - The kit which arrived earliest is selected.
2. Priority - The kit with the highest priority is selected.
3. Round Robin - A kit is selected from the next queue in the sequence of queues serving the machine.
4. Smallest Output WIP - The kit with the most room in its output buffer is selected.

Regardless of the local control policy, a kit will not be selected if its output buffer is full. Figure 7 shows the control policy panel for the example card plant. Note, for example, that the CLEANER2 machine has no setups and is using the FCFS control policy to select its next kit.

Control Policy Panel

Edit

	Machine	Behavior	Tiebreaker	MinSetup	Units	ChangeOnIdle?
1	TR SCREEN	CALENDAR	FCFS			
2	SCREENER	CALENDAR	FCFS			
3	TR PLACE1	CALENDAR	FCFS			
4	PLACE1	CALENDAR	FCFS			
5	TR PLACE2	CALENDAR	FCFS			
6	PLACE2	CALENDAR	FCFS			
7	TR IR	CALENDAR	FCFS			
8	IR	CALENDAR	FCFS			
9	CLEANER1	CALENDAR	FCFS			
10	TR ROUTER	CALENDAR	FCFS			
11	ROUTER	CALENDAR	FCFS			
12	TR TEST	CALENDAR	FCFS			
13	TESTER	NOSETUPS	PRIORITY			
14	TR CLEAN2	NOSETUPS	FCFS			
15	CLEANER2	NOSETUPS	FCFS			

Figure 7. Control Policy Panel

2.3 Modelling Setups and Setup Changes

As we discussed in Section 1, many of the machines in a card plant require a significant amount of setup and tuning when converting from one product to another. For example, a component placement machine requires a new set of component feeders and a new numerical control program. The placement robot then has to be calibrated to the position of the new feeders and run on some test boards to ensure accurate placement.

Setups are modelled as follows. The user must specify the name of the machine to which the setup applies and assign a name to the setup, allowing multiple setups to be defined. The user must then select the machine's operations that are supported by the setup.

Next, the user must indicate how long it takes to change the machine from one setup to another. For each setup change, the user must specify the length of time to change the setup (in seconds, minutes, or hours). The user can optionally specify an operator skill that is needed to perform the setup change and the number of operators required to do it. For each machine, there is a predefined setup called NULL in which no operation can be performed. The null setup allows the user to specify how long it takes to set up a machine at the beginning of a simulation run. Alternatively, there is a way to initialize a machine to a particular setup at the beginning of a simulation run without incurring a setup change delay. Figure 8 shows the setup change panel for one of the machines in the example card plant.

Setup Change Panel

Edit

Machine SCREENER

	Previous	Next	Time	Units	Skill	Operators
	===TOP====	==========	======	=========	==========	====
1	NULL	A	1.00	HOURS	ASSEMBLE	1
2	A	B	1.00	HOURS	ASSEMBLE	1
3	B	A	1.00	HOURS	ASSEMBLE	1
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						

Figure 8. Setup Change Panel

The user must then specify which setups to use during the simulation run and when to change setups. In MAXX, setup changes can be specified in four different ways:

1. On a calendar basis

   Using a calendar, the user can specify the date and time of a setup change. A calendar panel for this method is shown in Figure 9.

2. On a volume basis

   The user can specify that a setup change occurs after a desired number of cards have been processed by the previous setup.

3. Maximum flow

   Setups are chosen so that the flow through the machine is maximized based on an analysis of the input and output WIP.

4. Dynamically according to the local control policy

   If the setup change time is small, the user may want to change setups dynamically to the appropriate setup for the next kit chosen by the machine's local control policy.

A machine which changes setups based on maximum flow or a local control policy may experience chattering in which setup changes dominate productive work. To prevent this, the user can specify a minimum setup duration during which the machine cannot change setups.

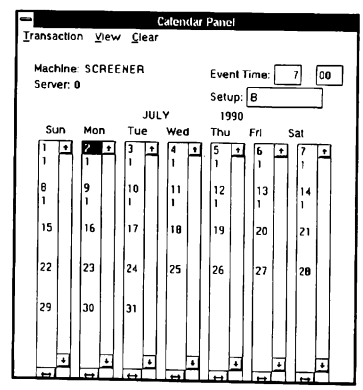


Figure 9. Calendar Setup Change Schedule Panel

2.4 Modelling Operator Constraints

In a manufacturing plant, there may be frequent shortages of operators with necessary skills. It is therefore important to model operator constraints carefully. In MAXX, the user first defines the operator skills that are used in the line. Then, the user defines a set of operator groups and assigns skills to each group. For the example card plant, there are several operator groups defined. The operator group ASSEMBLE possesses the ASSEMBLE skill for operating the machines in the ASSEMBLY area and the REPAIR skill for repairing these machines. The operator group MAINT possesses the TEST, REPAIR, and REWORK skills. For example, if the SCREENER machine breaks down and requires an operator with the REPAIR skill to fix it, then an operator from either the ASSEMBLE group or the MAINT group can be selected, since they both possess the REPAIR skill. If all of the operators with that skill are busy, then the machine will remain broken until an operator with the required skill becomes available.

Besides operational tasks, operators often have distractions, which in MAXX are called interrupts. Interrupts are used to represent things like going to lunch, attending a meeting, or being sick. Interrupts are modelled by specifying the length of the interrupt and either the periodic frequency or the calendar schedule for its occurrences.

2.5 Other Scheduling Parameters

In the discussion of setups, we mentioned that calendar and volume schedules are available for changing setups on a machine. We also mentioned the availability of calendar schedules for specifying operator interrupts. In MAXX, there are several other types of scheduling information including plant work hours, source schedules, shipment schedules, part deliveries and machine maintenance. Calendars are available for each of these categories. In addition, periodic frequencies can be assigned to most of these categories.

For example, a shipment schedule can be used to control the total amount of production. The buffers in a line impose a limit on the amount of WIP in the line. Shipments remove kits from the line, creating room for kits to flow and for new kits to be introduced. Shipments can be scheduled periodically or according to a calendar. In a periodic schedule, one might schedule shipments of 500 cards per day, with the time of the daily shipment varying according to a uniform distribution with a range of one hour. In a calendar schedule, one might schedule a shipment of 400 cards to occur at 5:00 PM on 2/17/90 and a shipment of 400 cards at 11:00 AM on 2/18/90. The calendar method provides the ability to model fluctuating demand schedules that cannot be modelled by periodic schedules with stochastic variation.

3. SIMULATION RESULTS

In this section, we present the results of several simulation experiments that we performed on the example card line. In MAXX, the results of a simulation run can be displayed either in graphical or tabular form. The outputs include both transient and aggregate measures of performance. The available simulation outputs include:

- shipments,
- activities at monitored buffers and machines,
- WIP per product,
- average WIP at machines and buffers,
- machine utilization,
- operator utilization,
- waiting time at machines and buffers, and
- cycle time.

3.1 Single Product Model

First, we simulated a model with only one product, A. There were no setup change requirements. The source was modelled as Always Available. We specified a shipment schedule that will ship all finished goods every 24 hours. The model was developed iteratively, adjusting the buffer capacities and the number of machines until the line was balanced in terms of machine utilization. Figure 10 shows the number of shipments of product A during a simulation of this model. Large irregularities in the shipment graph are caused by the failures that are modelled for some of the machines. Figure 11 shows the associated cycle time distribution and Figure 12 shows the machine utilization. The machine utilization graph shows the number of machines of each type on the horizontal axis.

The graphs indicate that the production throughput is about 3100 cards per day and that machine utilizations are about 70% or less. The machine utilizations are not higher due to the unavailability of operators during lunch and breaks, and also due to the use of minimum buffer capacities. For example, consider the routing of the ASSEMBLY area in Figure 4. The transport vehicle TR_PLACE1 takes 60 seconds to transport five boards. The process time for PLACE1 is 160 seconds per board and its output buffer (BPLACE1_A)

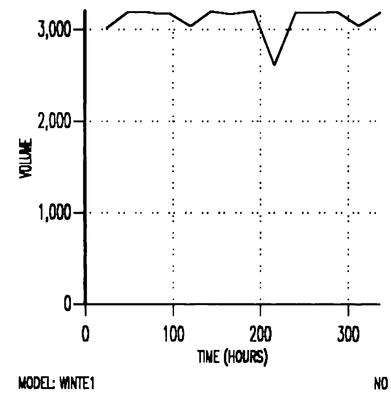


Figure 10. Shipment Quantities for the Single Product Model

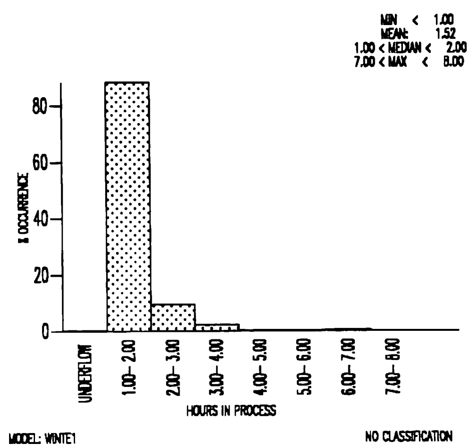


Figure 11. Cycle Times for the Single Product Model

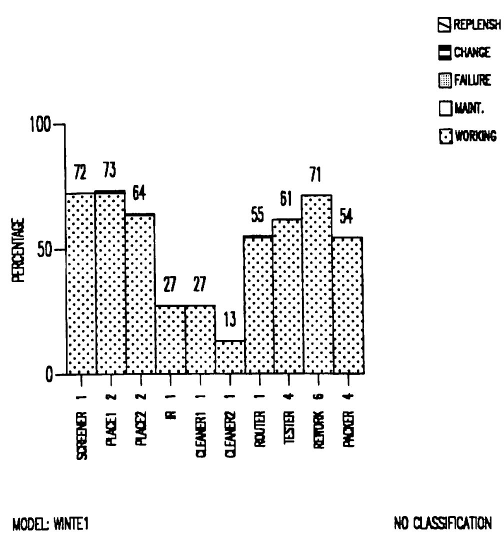


Figure 12. Machine Utilizations for the Single Product Model

can hold five boards. Whenever PLACE1 empties its input buffer, the transport vehicle can bring five more boards. But if the input buffer for TR_PLACE1 is not full to capacity with five boards, then the vehicle cannot move yet, and PLACE1 may become starved. This can happen if the process times for SCREENER and PLACE1 are unequal, or if SCREENER is starved. PLACE1 can also become blocked if its output buffer becomes full to capacity.

In general, a sequential flow line with minimum buffer capacities, a mix of machine types, unequal process times, and significant transport times relative to process times can have many possible interactions which lead to starvation or blocking. If higher utilization is desired, then the buffers can be enlarged. However, the WIP will increase if this is done.

3.2 Two Product Model with No Setups

In this experiment, we introduced another product, B. In the ASSEMBLY area, a copy of the routing shown in Figure 4 was made for the new product. Since the two products require different buffers and may require different processing times in ASSEMBLY, different routings are needed. At the end of the ASSEMBLY area, the two routings join together and enter the TEST area. In the TEST area, the two products share the routing shown in Figure 5.

No setups were specified in the model, allowing the machines in the ASSEMBLY area to switch freely between products based on their local control policies, which were all set to FCFS. The shipment quantities and cycle times for product A are shown in Figures 13 and 14 respectively. The corresponding results for product B are almost identical. There is a significant increase in the total number of shipments per day (3715) compared to the single product model. This increase is due to the ability of machines in this model to switch to another buffer when the active buffer becomes empty or blocked. There is also a small increase in the average cycle time (1.7 hours) compared to the single product model. This is caused by machines dividing their attention between the two products in this model.

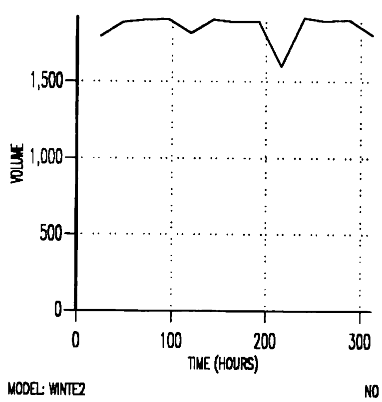


Figure 13. Shipment Quantities for Product A in the Two Product Model

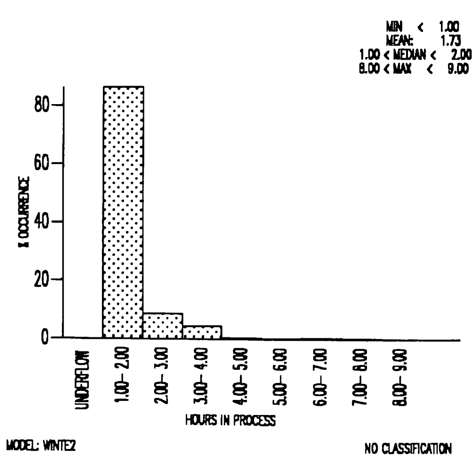


Figure 14. Cycle Times for Product A in the Two Product Model

3.3 Two Product Model with Setups

For this example, we chose the same model as above, except that setups are now required for the machines in the ASSEMBLY area. Each product requires a unique setup in each of these machines. It takes some of the machines one hour to change between setups. We scheduled setup changes every 24 hours on the machines in the ASSEMBLY area. No setups are required in the TEST area and hence the model for this area is identical to the one used in Section 3.2.

The shipment quantity graphs in Figures 15 and 16 show how production alternates between the two products. The average production per day (3060) is slightly less than in the other two models. The reduction in throughput is caused by the loss of production during setup changes.

Figure 17 shows that for most of the cards, cycle times are between 1 and 2 hours, but for a small portion of the cards, cycle times are between 26 and 27 hours, about one day later. The longer cycle times belong to cards that get trapped by setup changes. The trapped cards do not complete processing until the next day when the correct setups are available. To avoid trapping cards, we could stagger the setup changes down the line.

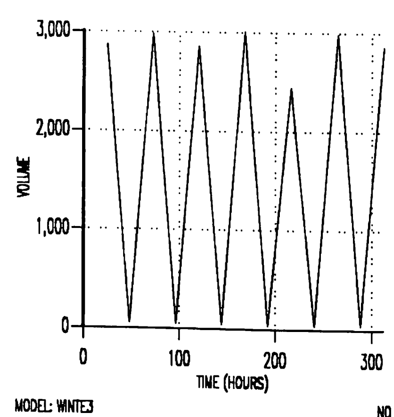


Figure 15. Shipments for Product A with Calendar Setup Schedules

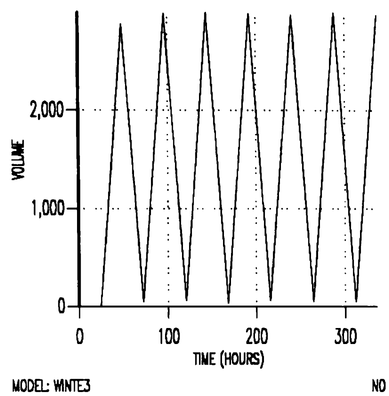


Figure 16. Shipments for Product B with Calendar Setup Schedules

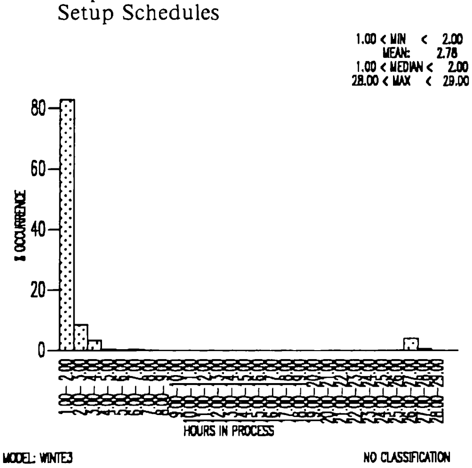


Figure 17. Cycle Times for Product A with Calendar Setup Schedules

The results of the simulation experiments for the three models are summarized in Table 1.

Table 1. Summary of Simulation Experiments

Model	# of Prods.	Setups	Ship-ments	Avg Cycle Time	Avg WIP
1	1	No	3109	1.5 Hours	49
2	2	No	3715	1.7 Hours	68
3	2	Yes	3060	2.7 Hours	87

4. SUMMARY

In summary, the MAXX simulator is capable of evaluating both long term line configuration issues and short term scheduling issues and is usable without any programming. In this paper, we showed how MAXX can be used to model and simulate an electronic card manufacturing plant.

Currently, MAXX is in use within IBM plants in beta test mode. It has proven useful in disk manufacturing and electronic card manufacturing and is presently being introduced into new domains.

ACKNOWLEDGMENT

The authors would like to acknowledge the contributions of Dorrie Kowalski and Jag Gill to the example model that is presented in this paper.

REFERENCES

An, C. H., S. Buckley, D. Palmer, J. Korein, J. Ahmadi, P. Ow, and U. Brunner (1990), "MAXX: A Manufacturing Oriented Simulator", In *Proceedings of the 9th IEEE Int. Electronics Manuf. Symp.*, IEEE, Washington, DC.

Chow, W.-M., E.A. MacNair, and C.H. Sauer (1985), "Analysis of Manufacturing Systems by the Research Queueing Package", *IBM Journal of Research and Development 29*, 4, 330-342.

MacNair, E.A. (1985), "An Introduction to the Research Queueing Package", In *Proceedings of the 1985 Winter Simulation Conference*, D.T. Gantz, G.C. Blais, and S.L. Solomon, Eds. IEEE, Piscataway, NJ, 257-262.