Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

A MANUFACTURING-ORIENTED SIMULATION PACKAGE TO SUPPORT SYSTEMS PLANNING AND ITS OPERATION

Shigeki Umeda

Musashi University
26-1, Toyotam-kami 1-chome
Nerima-ku, Tokyo, 176
Japan

ABSTRACT

This paper describes a manufacturing-oriented simulation software to support manufacturing systems planning and its operational control. The system configuration, modeling feature, algorithm, and its verification through an application to a practical manufacturing line will be described. The kernel of this software is an original manufacturing-oriented simulator. The simulator provides an original event-search algorithm, which handles efficiently simultaneous state-events to realize high performance of simulation. The simulator also supports an unique easy-to-learn modeling language, which can directly represent production-order methods in models. An advanced simulation software system includes the original simulator, the event trace database, which stores the event records of simulation, graphics post-processor, and scheduling evaluator. Finally, the author proposes a framework to integrate simulation to total production control system in CIMS environment.

1 INTRODUCTION

Simulation is an essential tool to support decision making in manufacturing operations at both planning stage and operational stage. A simulation model facilitates "what if" type analysis of a proposed manufacturing system.

Generally, implementing a simulation using general simulation languages, such as GPSS (Henriksen and Crain (1983)), SIMAN (Pegden (1985)), SIMSCRIPT(CACI Inc. (1983)) and others (Clementson (1985)), requires a special skill of simulation modeling. Manufacturing system can be represented as a network which posses processes, transporters, and other resources.

These manufacturing objects must be translated to simulation objects such as servers, queues, and transactions in a simulation model. In addition to that, manufacturing simulation models must represent not only physical material flow, but also control information flow, such as production-order or part scheduling. Accordingly, substantial time and effort is often required to implement the simulation models by using these simulation languages.

Recently, many manufacturing enterprises challenge to introduce Computer Integrated Manufacturing System (CIMS) for the purpose of a short-term decision making environment (IBM Corp.(1989), Scheer (1988)). Simulation can be also a powerful support tool to implement a hierarchical computer aided production management (CAPM) system in CIMS environment. In this case, we face a difficult problem how to integrate simulation into the total production management system.

First, this paper proposes a novel manufacturing-oriented simulator. The advantages of the simulator are an easy-to-learn modeling language and its introduction of production-order methods into simulation models. These facilities enable production engineers to implement complicated simulation models in a short time. Additionally, the simulation models can represent factory logistics models: such as MRP(Orlicky (1975)), KANBAN(Kimura and Terada (1981),Kim (1985)), and their mixed types. The simulator also supports an original simulation algorithm, which handles efficiently simultaneous state-events to provide high performance of simulation.

Second, the author discuss a framework to integrate simulation to total production control system in CIMS environment through an application to a practical manufacturing system.

2 SIMULATION MODEL

2.1 Modeling Overview

The simulation model is composed of three elements: manufacturing equipments configuration, manufacturing logistics models, and simulation condition parameters. The configuration model represents the physical structure of a target manufacturing system as a network. The logistics model represents relationship between parts flow and manufacturing control information, such as production-order. The simulation condition parameters are the control parameters which drives simulation such as run time length, data sampling intervals, and others. These elements are represented by a descriptor-based modeling form.

The descriptor-based modeling is a novel modeling methodology for describing manufacturing simulation models (Umeda (1989,1990,1991)). Even if the actual production system is vast, only a few production system models (modeling descriptors) will be defined. Accordingly, the descriptors that make up a manufacturing system model, such as machines, parts, transportation tools, pallets, operators, and their accessories, are countable. The modeling descriptors proposed here are typical elements for describing manufacturing systems. They are represented by frames with several slots (Figure.1). This modeling method is easy to learn, and enable users to implement a simulation model in short time.

2.2 Configuration Model

* Process

A process is represented by a network node, called a CELL. A CELL has multiple machines, two buffers for input/output, and job information. The job information includes setup information, work information, and job priority rules.

Machines are classified into two types: manual and automatic. The machines located in same CELL belong to same type. The manual one requires operators when it works. Machine-down can be also defined. In this case, it is necessary to give the parameters of recovery time distribution, and the number of operators for recovery.

Input/output buffers are basically defined according to each part name. The buffer size can be defined for individual part, and can be shared among multiple parts. A buffer can also have a criterion, which is equivalent to a replenishment point for production-order. The criterion is active when the CELL works at PULL mode.

Multiple Jobs can be defined. A job definition consists of input/output parts, parameters of process time distribution, and the setup name which the job requires. Job sequence is an option parameter, which indicates a job execution order.

```
/*   CELL FRAME DEFINITION */
CELL:cellname;
DEFINE MACHINE;                    /* MACHINE FRAME   */
   MEMBERS:m1,m2;
   TYPE:automatic;
   INIT  SETUP:m1=set1,m2=set2;
   END;
DEFINE INPUT BUFFER;               /* BUFFER   FRAMES */
   SIZE:p1=2,p2=1;
   CRITERIA:p1=1,p2=1;
   END;
DEFINE  OUTPUT  BUFFER;
   MIXED  SIZE:p1+p2=2;
   CRITERIA:p1=1,p2=1;
   END;
DEFINE SETUP:set1;                 /* SETUP FRAMES    */
   TIME:minimum=2,average=2.5;
   OPERATORS:1;
   END;
DEFINE SETUP:set2;
   TIME:minimum=1,average=1.5;
   OPERATORS:1
   END;
DEFINE WORK:wk1;                   /*   WORK FRAMES   */
   SETUP:set1;
   TIME:minimum=4,average=4.5;
   INPUT  PART:p1=1,p2=1;
   OUTPUT  PART:p1=1;
   END;
DEFINE WORK:wk2;
   SETUP:set2;
   TIME:minimum=3,average=3.5;
   INPUT  PART:p1=1;
   OUTPUT  PART:p1=1;
   END;
DEFINE JOB:jb1;                    /*   JOB FRAMES    */
   WORK TYPE:wk1;
   REPEAT:5;
   END;
DEFINE JOB:jb2;
   WORK TYPE:wk2;
   REPEAT:5;
   END;
JOB  SEQUENCE:  jb1*5+jb2*3;
                        /* JOB SEQUENCE DEFINE */
END
```

Figure.1 Modeling descriptors

* I/O Process (Customer/Vendor model)

The input process is called SOURCE, which is an entrance node of the network, and output process is called SINK, which is an exit node of network. Each part enters at SOURCE, travels through the network, and finally leaves at SINK. These process owns one buffer which structure is the same as CELL.

SOURCE and SINK are classified into four types: "periodic-type", "distribution-type", "order-type", and "scheduled-type". The periodic-type SOURCE/SINK sends or draws parts periodically. The distribution-type one has a distribution function of the time interval for sending or drawing parts. The concept of order-type is

equivalent to fixed-size ordering method of the inventory model.

When the scheduled-type is assigned, a schedule list should be described in the model. The schedule list has information on the schedule time, part names, and volume of parts. The scheduled-type SOURCE/SINK sends or draws parts according as the given schedule list.

*WIP room

The WIP room, the storage place for WIP inventories in the factory, is called POOL. On a large scale, it can be a warehouse between factories, and on a small scale, a lumber room in a factory.

*Transportation

Transportation is indicated by a network arrow. A transportation owns multiple carts or a conveyer as its tool. A transportation owns transportation lot-sizes, transportation time, and the names of upstream and downstream CELL. Transportation is classified into five operational types: PUSH, PULL, ESCAPE, REQUEST, and WAKE. Here, the author describes the details of activities of each types, when it owns carts as its tool.

PUSH-type transportation always carries parts from its upstream process. The original position of cart is the upstream side of part flow. When the part volume in buffer has increased to its transportation lot-size, the cart moves to its downstream with them. After unloading parts at downstream, it returns to the upstream with empty. Parts are loaded at upstream by the sequence in which the upstream process placed parts in its output buffer.

PULL-type transportation owns several KANBANs, which are rotated between an upstream process and a downstream process. The KANBANs defined here are equivalent to the withdrawal KANBANs in the JIT production system (Monden (1983)). A cart carries parts from an upstream to a downstream, and also carries KANBANs in the opposite direction. The original position of cart is the downstream side of part flow. When the volume of part becomes lower than criterion of buffer, KANBANs is picked up by cart. As soon as a KANBAN is picked up at downstream, the cart moves to upstream with it. At the upstream process, it searches for the part listed on the KANBAN, and loads them onto itself. The cart then puts the KANBAN on the upstream KANBAN board, and gives a production order of the same part to the upstream process. As soon as the quantity of WIPs in upstream process reaches transportation lotsize, the cart withdraws the parts with KANBAN, and returns to downstream (Figure.2).

ESCAPE-type transportation works only when its upstream process is blocked by a full-output status. The original position of cart is the upstream side of part flow. As soon as full-output blocking occurs at upstream, a cart loads parts of lotsize and moves to downstream with them. The following activities are equivalent to the PUSH-type.

REQUEST-type transportation works only when its downstream process is blocked by a no-input status. The original position of cart is the downstream side of part flow. As soon as no-input status occurs at downstream, a cart generates a temporary KANBAN, and moves to upstream with it. The following activities are equivalent to the PULL-type. While, the REQUEST-type one disposes the temporary KANBAN, as soon as it returns to downstream.

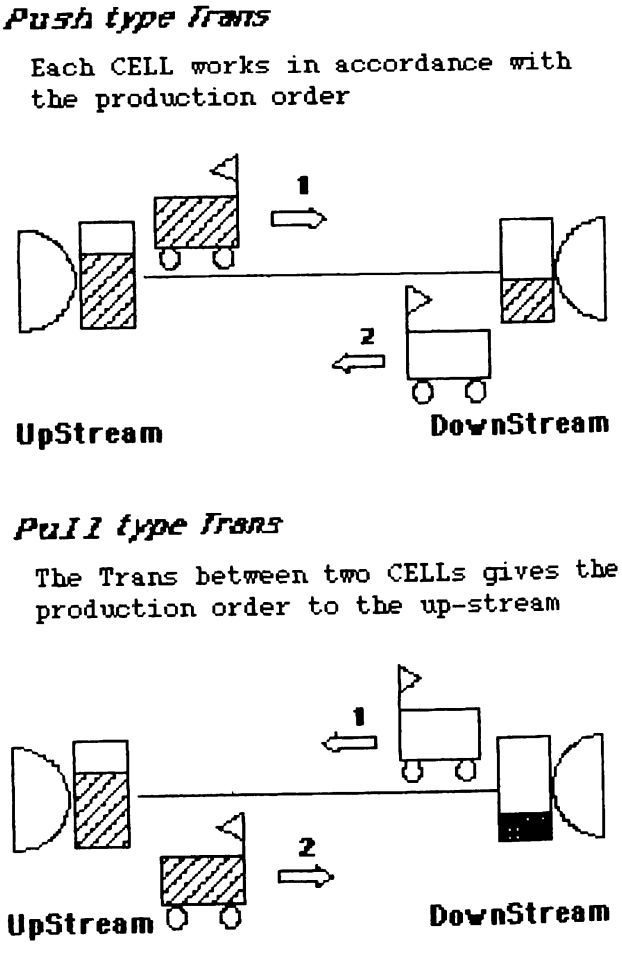

Figure.2 Transportation Model

WAKE-type transportation is used to represent a part-oriented manufacturing system such as a job shop floor. The original position of cart is the upstream side of part flow same as PUSH-type. Parts are loaded at upstream according to the sequence in which the upstream cell placed parts in its output buffer. A cart carries them to downstream. When the cart reaches downstream, it unloads parts and gives a job order to the upstream cell.

*Human resources

Human resources constitute a very important element in a manufacturing system. Line workers consist of machine operators, setup operators, and often maintenance engineers. Workers' groups and group work schedules can be defined.

2.3 Logistics Model

In general, manufacturing systems can be divided into two types, PUSH and PULL systems. A PUSH system may be characterized as a "schedule-driven" system, in which the required quantity of each individual part is determined by the use of conventional material requirement planning (MRP) techniques (Orlicky(1975)).

In PULL systems, on the other hand, the buffer between pre- and post-processors plays an important roles as an indicator that issue production orders (pull signals) to pre-processors. Thus, a PULL system can be characterized as a "buffer-driven" system. For general simulation language, a flexible modeling of PUSH/PULL logic is rather a difficult task for programmers and production engineers.

The main difficulty lies in the operation for synchronizing pre-processors with post-processors by means of information boards (KANBANs). The PULL system described here is a dual-card KANBAN system, which is equivalent to Toyota's famous JIT production system(Monden (1983)). One of the basic characteristics of the proposed approach is its ability to incorporate production-order methods by flexible usages. User defines PUSH, PULL, hybrid PUSH/PULL logic, and more complex logistics system by using modeling descriptors. The following phases describe these definitions, and the implementation methods of these logistics models by using modeling descriptors.

*PUSH system model.

The PUSH system is a schedule-driven system, as mentioned previously. In PUSH system, each process is controlled by the production directions (Master Production Schedule). In the proposed system, these directions are represented as a job sequence of CELL or a schedule list of SOURCE/SINK. Each process works independently of the others accordingly to its own job sequence or schedule list. And the type of transportation between processes is required to define PUSH-type. Consequently, the transportation works only as a mean of transporting modeling object (Figure.3).

*PULL system model.

There are several implementations of the PULL system(Monden(1983)). The PULL system described here is the dual-card KANBAN system, as mentioned previously. To simulate the dual-card KANBAN system, the following points should be represented in the simulation model:

(1) The timing with which the post-process invokes the pre-process

(2) The request order of post-process to pre-process

(3) The action of pre-process when it receives the request order from post-process.

The proposed system defines these items as follows. (1) is defined as the buffer criterion in post-process (CELL or SINK). (2) is represented by the KANBANs in the PULL-typed transportation. (3) is defined as the job definition in pre-process (CELL or SOURCE) (Figure.4).

*Hybrid PUSH/PULL system model.

This type is a complex set of logistics. A typical example is a system, which is overall controlled by schedule-driven operation, being locally controlled by KANBAN operation. There are often constraints on the implementation of a production system, such as space allocation, vendor management, machine characteristics, and management policies.

Alternatively, a manufacturer might plan to integrate the scheduled-driven policy used in his/her own factory into KANBAN-driven vender management. To implement such a hybrid PUSH/PULL system model, PUSH/PULL definition is independently described both at each CELL and at each TRANS (Figure.5).

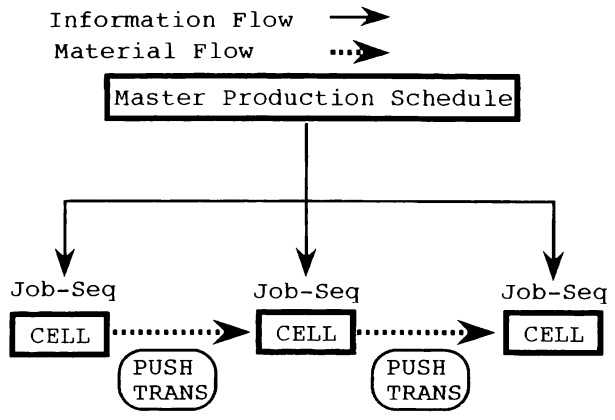

Figure.3 PUSH System Model

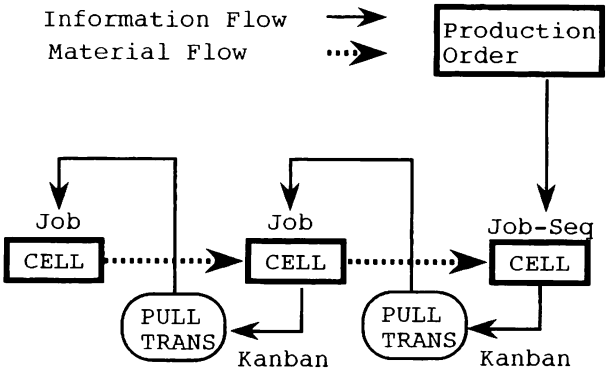

Figure.4 PULL System Model

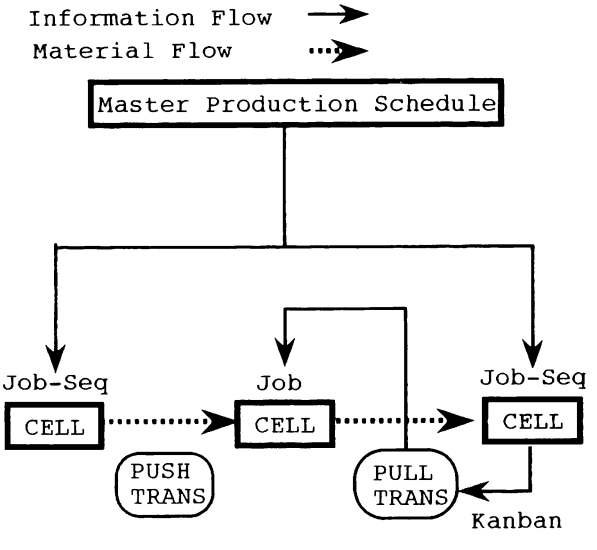

Figure.5 Hybrid PUSH/PULL System Model

2.4 Simulation Condition

SIMAN provides a simulation control block, which is described independently on system configuration models (Pegden(1985)). This method makes the model structure to be clear, and reduces workload to repeat simulations with the different control parameters. The control parameters are described in a condition frame. The details of description are as follows:

(1) Execution time of simulation
(2) Replication
(3) Trace specification for model debugging
(4) Event record specification
(5) Initial deletion specification

3.ALGORITHM

3.1 Simultaneous State-Event

One of the remarkable points of discrete event simulation is that simultaneous state-events frequently occur. The simultaneous state-events mean that multiple events occur at the same simulation time. When the simultaneous state-events occurs, simulation program must multiply scan its event list at the same simulation time. Accordingly these events often extremely disturb the simulation performance.

There are two types of simultaneous state-events. One is accidental simultaneous state-events, and the other is intentional simultaneous state-events. They are described by referring to Figure.6. The axis of abscissas of Figure.6 represents the simulation time (T), while the axis of ordinates represents each object. The circles with symbols A1 - G3 indicate events. A continuous line indicates the continuation of an event, while a dotted line indicates a chain relation i.e. an interrelated or linked relationship.

The accidental simultaneous state-events means a case where two or more events accidentally occur at the same time. In Figure.6, events A1 and F1 at the time T1 are examples of this.

On the other hand, the intentional simultaneous state-events are ones which are intentionally caused when an event occurs. Examples of the intentional simultaneous state-events in Figure.6. includes: B4; a work-in-process enters into the input buffer -> C4; a machine waiting for material starts to drive. C5; a machine executing a job completes the work -> C6; the work-in-process enters into the output buffer. C6; the work-in-process enters into the output buffer -> D6; the product is loaded on a

supply cart waiting for material. And F2; a machine in operation fails -> G2; an operator starts a repair work.

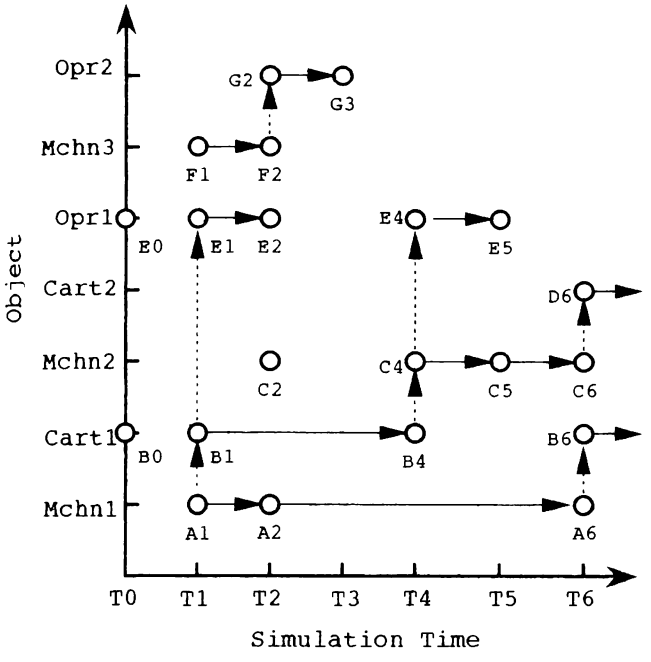

Figure.6 Simultaneous state-event

3.2 Simultaneous State-Event Rule Form

Once a system configuration is determined, it should be possible to make limitations on what sort of events occur, and the type of events that are possible when events are chained with other events. The simultaneous state-events rule collection describes this in a rule form. That is, the simultaneous state-events rule collection describes the patterns of intentional simultaneous state-events. Examples of rules includes:

Example.1

A machine object performs output to the buffer
-> Search for a supply cart waiting for input in the downstream of the cell to which the machine belongs.

Example.2

An operator is released as a machine object completes an arrangement change
-> Search for a machine that has failed and is waiting for repair.

or
-> Search for a machine that is waiting for an operator for changing an arrangement.

3.3 Event Process Algorithm

When a simulation begins, the simulator generates a list of object-frames. Each object-frame represents a object, which causes events: machine, cart, operators, and others. The simulator also owns a stack: the Event-stack, which stores the object-frames which cause simultaneous state-events.

The proposed algorithm is the object-frames based event search algorithm by using simultaneous state-event rules. The procedures is shown as follows.

1. Sets Current Time on Initial Time.
2. Thus, performs a search for object-frame pointers in order. Finds all events which occur at the Current Time. Updates the Current Time at that time, if necessary. They are the accidental simultaneous state-events. Pushes them into Event-stack.
3. Processes the accidental simultaneous state-events on the Event- stack in order. When an event is proceed, look up the intentional simultaneous state-events rule collection.If any intentional simultaneous state-events matches, add them on Event-Stack.
4. Repeats above operations until the Current Time is over Simulation time.

4 OUTPUT REPORT GENERATOR

The proposed simulation is based on Monte-Carlo method using random numbers generated by computer. Accordingly, the simulation output data require a statistical analysis for the system performance to be evaluated. The proposed system depends on the independent replication method, which is applicable to both terminated simulation and steady-state simulation (Law and Kelton (1982)). The former is suitable for predicting the behavior resulting from a specific production strategy, whereas the latter is suitable for understanding performance evaluation data under stable conditions.

The output report includes detailed level-of-performance indices. For instance, machine performance is divided nine indices: idling time (total time in which any machine in CELL does not work), no-input time (total time in which there are no input parts), full-output time

(total time in which the output buffer are full), working time, recovery time for machine-downs, setup time for job exchange, waiting time for operators, waiting time for recovery, and waiting time for setup operators. These details of the data enables users not only to understand the status of the system easily but also to discover any hidden bottlenecks in the system.

5 APPLICATION EXAMPLE

The author applied the proposed system to an actual production system. This system is composed of several CELLs containing more than twenty automatic machines, one SOURCE, and two SINKs. And this system has a complex material flow with loops. The control operation is based on typical hybrid PUSH-PULL logistics (Figure.7). Figure.8 shows a comparison of accumulated production volume between actual and simulation at the final process. The results of simulation confirmed that the proposed system is applicable to practical production systems.

A manufacturing engineer could implement a simulation model within a few days. And the simulation run of one day operations required a few minutes cpu time.

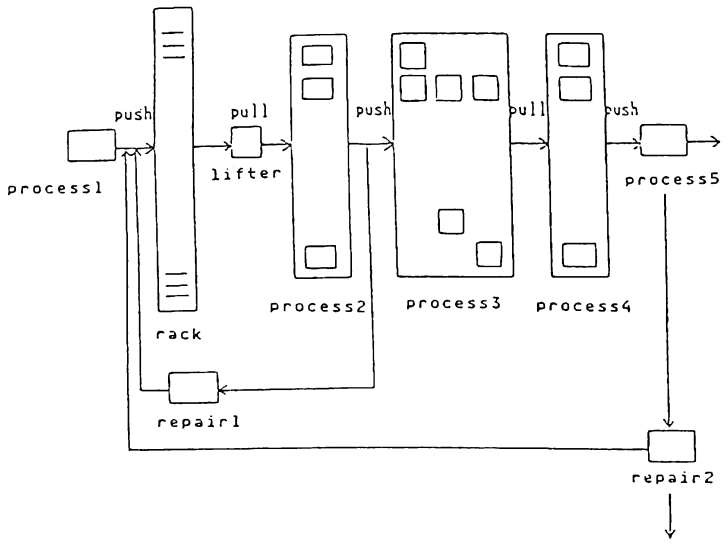

Figure. 7 Configuration of the Applied Manufacturing System

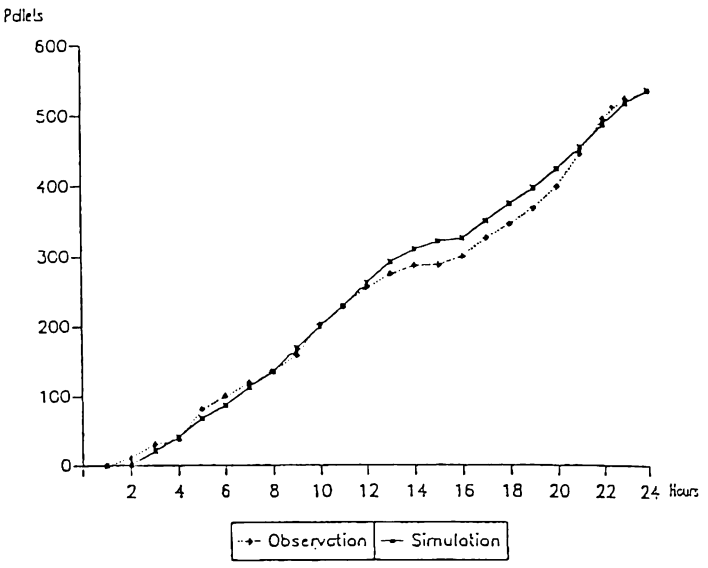

Figure.8 A comparison between actual and simulation at SINK

6 ADVANCED SIMULATION SYSTEM FOR PRODUCTION PLANNING AND CONTROL SYSTEM

Thus far, the author have suggested a manufacturing system simulator. In this section, the author proposes an advanced simulation system, which includes the above simulator as its a kernel module. It also includes a graphic post-processor, a model generator, and a schedule evaluator. A system configuration is shown in Figure.9.

The graphic post-processor supports graphic charts which represent the output analysis data of simulation. The model generator gets both the system configuration and the manufacturing schedules, and generates the simulation model automatically. The schedule evaluator provides the facility of manufacturing scheduling simulation.

There exist two types of manufacturing simulation to support planning and control operations in a factory: off-line simulation and on-line simulation. The former is an assessment at design and planning stages. Manufacturing engineers mainly discuss the system configuration, such as buffer size, machine capacity, vehicle speed and others.

This operation supports the analysis works such as systems planning or line improvement. These works require simulation independently on daily manufacturing operations.

On the other hand, the on-line simulation supports the daily plant control operations. The main input parameters of the on-line simulation are daily manufacturing schedules. MPS/MRP produces the daily manufacturing schedules at production planning phase. While, planning changes or line-troubles often occur in daily manufacturing practice. Accordingly, manufacturing field workers require so-called re-scheduling operations to synchronize their field works with the plans. Re-scheduling operation requires special skill of operators. Furthermore, it takes much time to select one among many possible schedules.

Repeats of simulation with different scheduling parameters will support such real-time re-scheduling operation. This operation requires a high performance of simulation, and dynamic changes of the simulation parameters in real-time scale. The model generator and the schedule evaluator will provides such works. The former directly rewrites the manufacturing schedules in the simulation model, such as the shipment orders lists and part-throw lists, and the later estimates whether the given schedule is executable or not.

"Virtual Plant System" is a concept which combines real-time monitoring system with on-line factory simulation by computer network (Figure.10). The real-time monitoring system observes all the data of main activities at each plant layer: such as production orders, venders actions, WIPs changes, machine-downs, and others. And, it periodically sends these data to the parameters generator. The parameters generator transfers them to simulation parameters, and send them to the on-line factory simulator, which has a hierarchical model of multiple lines or sections. Thus, the simulator can realize the same situation as the actual plant by the latest information.

While, a planning system such as MPS/MRP sends the plant control information to both the actual plant and the virtual plant. In this environment, simulation can realize the same material flow and information flow as the actual plant in computer. Such environment will enable manufacturers to predict the key events of plant, whenever a new direction is performed. This will be a great support for manufacturers to make decisions in plant operations.

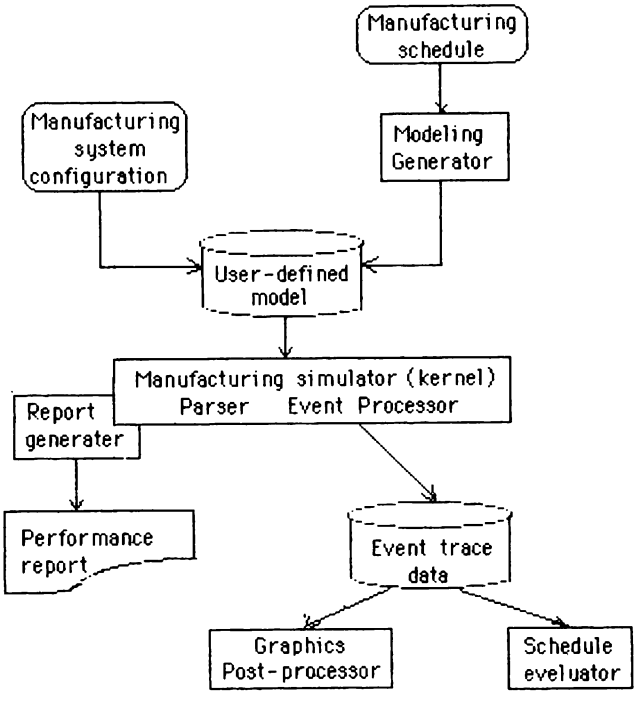

Figure.9 Configuration of the advanced simulation system

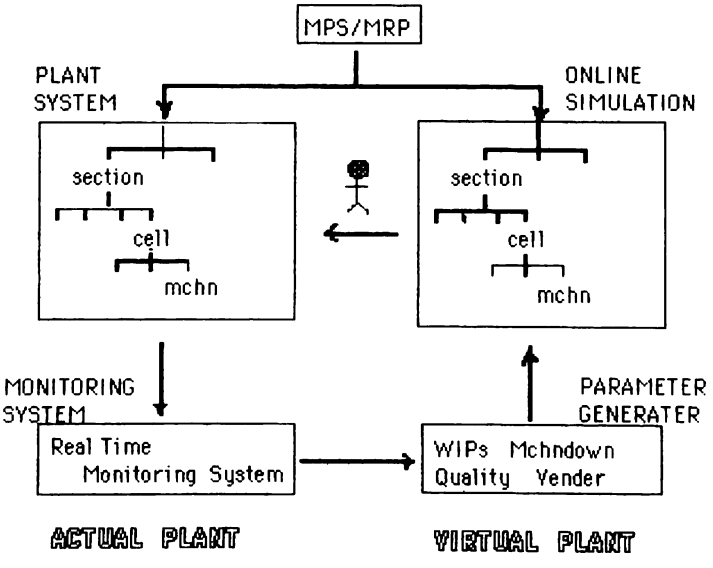

Figure.10 Concept of Virtual Plant System

CONCLUSIONS

The author has proposed a novel manufacturing-oriented simulator. This simulator is currently available as a software product of IBM Japan Ltd. The advantages of the simulator are an easy-to-learn modeling language and its introduction of production-order methods into simulation models. Further, the simulator owns an original event search algorithm using the simultaneous state-events rules. These facilities has shown its high performance on both implementation of a modeling and its execution through an application to a actual manufacturing system.

Further, the author has proposed an advanced simulation system, which includes a graphics post-processor and schedule evaluator. The current version is on primitive stage. This system, which is installed in a factory of an IBM user, is currently being modified to be available on workstation environment.

"Virtual Plant System" is a framework that integrates a planning/scheduling system (MPS/MRP), a factory monitoring system, and the advanced simulation system. This is an intelligent simulation-based decision support system in CIM environment.

To realize the above concept, both a fast simulation and a distributed simulation environment by using multiple workstations will be required. The author believes that the proposed simulation system will be useful to realize such environment.

REFERENCES

CACI Inc. 1983. SIMSCRIPT II.5 Reference Handbook, 2nd edition: Los Angeles

Clementson,A.,T. 1985. Extended Control and Simulation Language: Cle-Com Ltd. Birmingham

Henriksen, J.O., and Crain, R.C. 1983. GPSS/H User's Manual, 2nd edition: Wolverine Software Corporation, Virginia

IBM Corp. 1989. The CIM Enterprise, G320-9802

Kim,T. 1985. Just-in-time manufacturing system: a periodic pull system, International Journal of Production Research 23,3: 553-562.

Kimura,O. and Terada,H. 1981. Design and analysis of Pull system, a method of multi-stage production control, International Journal of Production Research 19,3: 241-253.

Law,A.H., and Kelton,W.D. 1982. Simulation Modeling and Analysis: McGraw-Hill

Monden,Y. 1983. Toyota Production System: Institute of Industrial Engineers, New York

Orlicky,J. 1975. Material Requirement Planning: McGraw-Hill Book Co. Inc., New York

Pegden, C.D. 1985. Introduction to SIMAN, version 3.0: Systems Modeling Corporation, State College, Pennsylvania

Pritsker, A.A.B. 1984. Introduction to Simulation and SLAM II, Halsted Press and Systems Publishing Corporation

Scheer,A.W. 1988. CIM - Computer Steered Industry: Springer-Verlag.

Umeda,S. et al. 1989. The Manufacturing-Oriented Simulation Package-MANMOSS, IBM Inter-divisional Technical Liaison.

Umeda,S. 1990. The Manufacturing-Oriented Simulation for Performance Evaluation, Proc. of Japan-U.S.A. Symposium on Flexible Automation.

Umeda,S. 1991. Manufacturing-Oriented Simulation for Production Management, Proc. of 6th International Conference on Operational Management Association.

AUTHOR BIOGRAPHIES

SHIGEKI UMEDA is an Associate Professor in the Department of Management Science at Musashi University, Tokyo. He received B.E. and M.E. degrees in industrial engineering from Waseda University in 1980 and 1982 respectively. He worked as a research staff of IBM Tokyo Research Laboratory from 1982 through 1990. His research interests are focused on Artificial Intelligence, Production Control, and Simulation.