Proceedings of the 1983
Winter Simulation Conference
S. Roberts, J. Banks, B. Schmeiser (eds.)

A MODULAR APPROACH TO THE SIMULATION OF MANUFACTURING PROCESSES

Sarah A. Shoaf
Westinghouse R&D Center
1310 Beulah Road
Pittsburgh, PA 15235

During a recent computer simulation study of a highly automated Electronics Assembly Plant, a new approach to modeling manufacturing systems was developed. This new approach involves a modular structure for modeling that will facilitate the design, description, coding and use of the models. In this paper, seven functional "modules" are defined, each containing functionally related subroutines that are interconnected according to the nature of the relevant system. Use of this modular approach provides a more flexible and less expensive method of simulation modeling.

1. INTRODUCTION

Simulation has proven to be a valuable tool for use in the design and development of manufacturing systems. As the systems that are modeled become increasingly more complex, standard network simulation models can become unwieldy and very expensive to run. In this paper, a new approach to the simulation of complex manufacturing processes is defined which uses discrete modeling techniques to create more manageable models. In this approach, the code is divided into seven "modules", each containing functionally related subroutines. We have found that models designed in this fashion are highly flexible since all coding for a particular action is located in one subroutine rather than spread throughout a large block of integrated code.

A recent simulation study of a highly automated, thirty-operation Electronics Assembly Plant (EAP) currently under construction provided the stimulus for developing this new approach. The plant planning team desired to use simulation to provide detailed information on such things as product flow rates, manpower requirements, transporter merge point queues and Automated Storage and Retrieval System volume and transaction rates. This paper will present a comparison of two separate models, both developed by the author, created to address these questions. Both models were developed using the FORTRAN-based Simulation Language for Alternative Modeling known as SLAM (Pritsker & Pegden, 1979).

This paper is divided into five sections. Following the introduction, section 2 contains a brief discussion of the Electronics Assembly Plant and the original network simulation effort. Section 3 details the modular structure for simulation that was developed. Section 4 describes the modular EAP model and Section 5 presents conclusions and a discussion of further applications.

2. ELECTRONICS ASSEMBLY PLANT

The Electronics Assembly Plant (EAP) currently being built by Westinghouse at College Station, Texas, will produce approximately five thousand Printed Wiring Assemblies (PWAs) per month. Each PWA consists of a printed circuit board with various components attached according to the desired function of the PWA. In 1981, during the early stages of the plant planning effort, a simulation model was developed to provide detailed information on such things as product flow rates and manpower requirements. The model was also used to determine volume and throughput requirements for an Automated Storage and Retrieval System (ASRS) that will provide the basis for the material handling system in the new plant.

The manufacturing activities in the EAP consist of a sequence of operations that a particular job undergoes, resulting in a completed PWA. The PWAs will travel through the plant in 'lots' of ten or so similar boards and will be carried in plastic boxes called totes. Some lots will require more than one tote, depending on the number and size of the components to be attached to those particular boards. In addition, lots may be split before certain rework operations so that boards not requiring rework may continue with their normal process flow.

The EAP manufacturing process flow contains approximately thirty different assembly operations. However, most lots will only require between fifteen and twenty of these operations. Process times at each operation are determined by board type.

The flow of lots between operations utilizes a highly automated material handling system. All lots must travel to the central ASRS area between operations. The movement of material between the ASRS and the shop floor is handled by a series of transporters and conveyors. Once a lot arrives at the ASRS, a central computer identifies the lot and determines its next operation. This is accomplished using a laser scanner and individual bar codes.

Once the next operation has been identified, the computer checks to see if the next operation is busy and to see if there are any higher priority jobs waiting in storage for the indicated operation. If "No" to the above, the lot will proceed directly to its next operation, making a storage 'bypass'. Otherwise, the lot must be placed into storage and will be retrieved when a worker is available.

The above description of the EAP indicates the size and complexity of the general problem. Both of the simulation models that were developed model the basic system as described above. Details of plant operation are different or more defined in the second model due to planning changes and evolution of the plant designs.

The original SLAM model of the EAP relied heavily on network modeling techniques to produce a network similar to the manufacturing process flow diagrams. This model made extensive use of conditional and probabilistic branching in order to handle such things as rework, routings and variable process times. These factors were considered in the SLAM network code as they occurred, rather than having all the decision logic located in a central subroutine. The network model used 112 SLAM user functions to perform the checking necessary for storage decisions and to assign process operation times. This complexity made the model extremely unwieldy. A simulation run of the model for nine months of simulated time required well over an hour of CPU time. The original model was difficult to debug and even more difficult to change as plans for the plant evolved. The code was also nearly impossible for a non-SLAM specialist to understand, much less to run or modify.

The original effort was able to provide the desired information on overall plant operating characteristics and results of this simulation were used as input in the decision as to what type of storage and retrieval system to purchase. However, when further, even more detailed simulation studies of the EAP were desired, it was decided to start over and approach the problem in a new way.

Approximately one year after the original simulation was completed, it was desired to again use simulation to investigate additional features of the EAP manufacturing system. In this new effort, information was desired on such things as queues at transporter merge points, individual storage carousel characteristics (four were purchased based on the results of the first simulation) and their corresponding Mobots. Lot splitting, multiple tote lots, two shifts per day and specified routings for board styles were also to be incorporated.

A new model of the EAP manufacturing process was to be developed that would be more efficient, easier to understand and modify and that could be developed more quickly. The SLAM simulation language would still be used, but a new approach to the problem was required. Instead of basing a network model directly on the process flow and using discrete subroutines to fill in the gaps, it was decided to break away from the manufacturing flow diagrams. The new model would be a set of many discrete subroutines, each unique in function, and a network consisting only of individual operational queues and ensuing activities. Although the proposed approach requires more programming knowledge than is needed to develop a network model, the benefits of the "modular" approach were found to far outweigh this disadvantage.

3. A MODULAR STRUCTURE FOR MANUFACTURING MODELS

The modular approach to simultaion, described in this report was conceived of and developed by the author in conjunction with G. C. Powell of the Westinghouse Manufacturing Systems Simulation Group. The idea was first formalized by G. C. Powell and as work on the second EAP simulation proceeded, the Modular Approach to Simulation for Manufacturing Systems evolved into the practicable methodology described below.

This new methodology involves a modular structure for modeling manufacturing systems that facilitates the design, description, coding and use of manufacturing models. In this approach, seven functional modules are defined: control, entity development, scheduling, routing, process/operation, decision logic, and summary information. Each module contains functionally related subroutines that are interconnected according to the nature of the relevant manufacturing system. The modular approach to simulation has as its basis the desire to keep the model coding for each action as independent as possible of all other coding. The seven functional modules are described below:

1. Control Module

Contains driver subroutines unique to the simulation language being used such as initialization routines or sets of subroutine call statements.

2. Entity Development Module

Assigns descriptive information to the objects that will flow through the model as they arrive at the manufacturing system. This information will be used to distinguish between similar entities or to provide information for later use such as lot size, part type, etc.

3. Scheduling Module

Schedules the arrival of entities or objects at the beginning of the manufacturing process flow as well as indicating the amounts of resources available in the model over time. Entities can be scheduled to arrive daily, weekly, monthly, etc., and resource availabilities may change from one shift to the next.

4. Routing Module

Contains information describing the flow of entities throughout the manufacturing system for each part type.

5. Process/Operation Module

Contains information on the nature of each operation (process times, resources required) and specifies the type of operation (rework, assembly, material handling, etc.).

6. Decision Logic Module

Contains the logic required for decision points in the process flow in order to accurately represent the real system. This logic would be used in such areas as making storage decisions, deciding when to send more work to the floor, allocating workers, selecting alternate routings or in including any other operating policies not considered elsewhere in the model.

7. Output Information Module

Provides user-collected statistics on data not automatically generated by the simulation language being used.

Each module may contain more than one subroutine addressing the relevant function. The subroutines in each module will generally be interconnected. The manner in which the modules themselves interact will depend on the nature of the manufacturing system being modeled. The interconnections for the modular simulation of the EAP are shown in Figure 1.

The modular structure that we propose allows for highly flexible models since any changes can generally be made in a single subroutine rather than having to be made throughout a large network or large integrated block of discrete coding. The subroutine call statements and common blocks in a particular subroutine

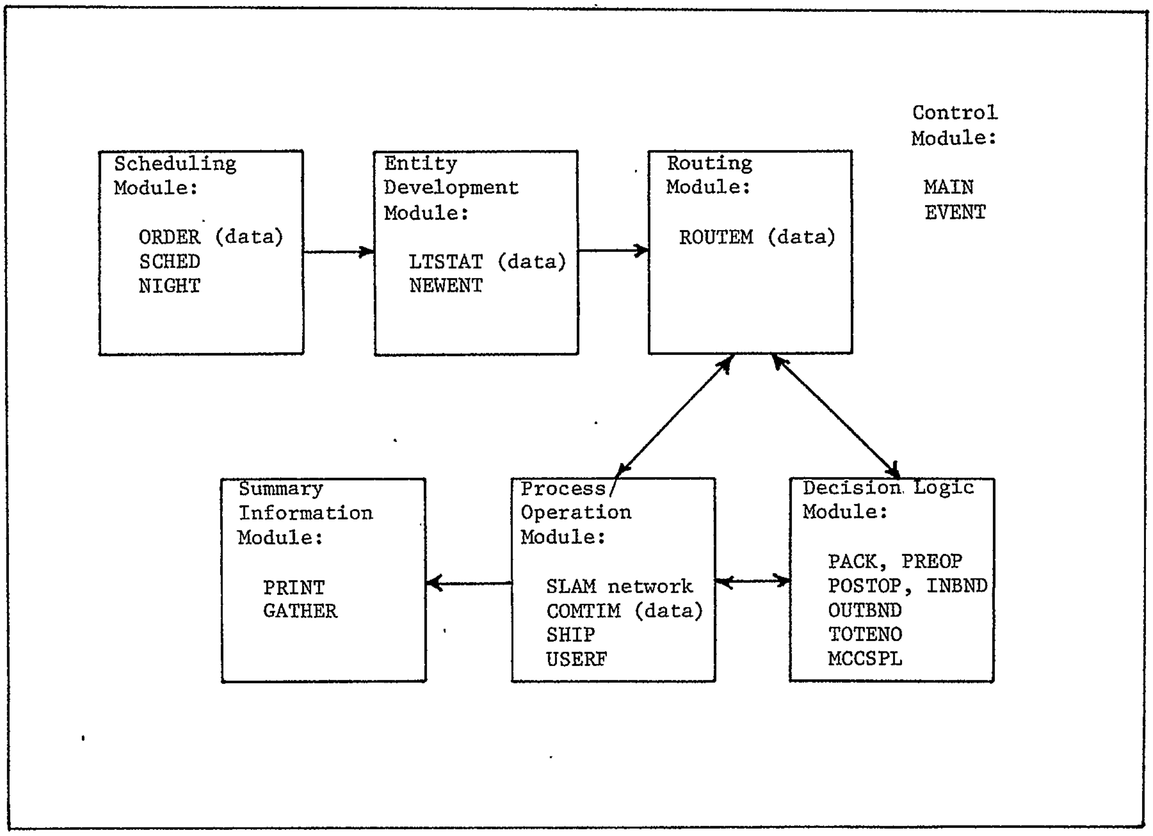

Figure 1: The Modules For The EAP Simulation

will indicate exactly which other subroutines must be checked when changes are made. The models are also relatively easy to debug since errors can usually be traced to a particular subroutine. The separation of the problem into specific modules aids in the process of explaining the model to non-simulation experts. Also, data can be concentrated in specific data subroutines so that the parameters for a given run may be easily identified or changed.

In simulating the EAP, it was found that this complex manufacturing system could be easily modeled using the proposed modular structure. This was accomplished with no overlap between modules. The second EAP model will be described below.

4. THE MODULAR EAP SIMULATION

The modular simulation of the EAP relies on discrete coding rather than a network structure to handle almost all aspects of the modeled system. Twenty FORTRAN subroutines and a small block of SLAM network code are required to model the complex system described in Section II. Each of these blocks of code can be identified with a specific functional module. Figure 1 indicates the subroutines involved in each of the seven proposed modules.

The control module contains two driver subroutines, MAIN and EVENT. MAIN drives the entire simulation while EVENT is used to call the thirteen event subroutines used to manipulate entities in the manufacturing system and to collect data.

The scheduling module consists of three subroutines. ORDER reads in data on the frequency and number of orders that are to be scheduled. SCHED generates the proper number of lots at the order intervals specified and feeds them into the entity development module. NIGHT handles the scheduling of shifts. All operations except the oven and the wave solder stations are shut down during the second shift. Hence, every eight hours this subroutine is called to alter resource availability.

The entity development module consists of two subroutines. LTSTAT contains various data on board styles, lot sizes and the number of totes per lot. NEWENT assigns this information to the entities as they enter the manufacturing system. This subroutine then places each entity at the receiving operation in the process/operation module.

A single subroutine, ROUTEM, comprises the routing module. This subroutine consists only of data statements and contains routings by operation number for each of the four board styles being considered at present. It also contains transport times between the storage area and the operations and maximum queue sizes at each operation.

The process/operation module consists of four subroutines. DATA contains the SLAM network data cards. These establish the queues and activities for each of the twenty-six operations currently in the model. Storage queues representing the four carousels and activities representing the merge points on the inbound transporter lines are also included. COMTIM, which consists only of data statements, contains information on set up and run times for each operation by board type. Lot-splitting percentages at rework operations for each board style are also recorded. SHIP schedules the removal of completed jobs from the carousels for shipping purposes, while USERF contains the user functions for computing process times and determining when lot-splitting is required.

The largest module, decision logic, contains seven subroutines. POSTOP and PREOP handle the transporter delay between each operation and the ASRS. PACK is called as jobs exit storage in order to allow the representation of multi-tote jobs as single transactions while they are on the shop floor. In storage, all totes are represented individually so as to simulate the proper level of competition for storage space and Mobot time. MCCSPL splits lots when required into sets of good and faulty boards. INBND handles the placement of jobs into the storage carousels, while OUTBND handles the extraction of jobs, in order of priority, as dictated by the work levels at the relevant operations. The final decision logic subroutine, TOTENO, deals with the complicated process of tote reduction and storage bypass decisions. Multi-tote lots consist of one tote containing partially assembled circuit boards and other totes full of the remaining components required for assembly. At evaluation operations, only the tote containing the boards is required. Hence, the component totes can be placed into or remain in storage. These component totes must then be located after the main tote completes the evaluation operation before further assembly operations can be performed.

The output information module contains two subroutines. GATHER and PRINT are used to collect and display information on the number of lots waiting in storage for each operation, the number of totes currently in use, and other relevant statistics.

Although this model of the EAP is very complex, the sectionalization of the problem into small pieces made it much simpler to develop and to debug. A main goal in developing the new model was to have all of the decision logic come into play at only one location in the process flow -- at the ASRS. Hence, all decision subroutines are called as jobs enter or exit the storage area. In this way, there are no problems with keeping track of which decisions were made where and when. This also greatly simplifies the incorporation of changes since there can be relatively few "ripple effects" as compared to the previous simulation.

In this simulation effort, all data is contained in individual, non-functional subroutines rather than being scattered throughout the network on activity branches. This allows changes to be made much more easily, in addition to providing easy reference tables on exactly which parameters were used in a given run. Also, the data subroutines can be recompiled after changes are made without having to recompile the entire model.

After completion, the new EAP model contained approximately 1300 lines of code, the same as the previous model. Most importantly, it required only 25 percent of the CPU time needed for a comparable length run of the previous model. This was in spite of the fact that the second simulation included many more material handling details such as multiple totes, lot-splitting percentages, tote reduction and specified carousel storage areas.

All in all, the modular EAP model is eminently more satisfactory than the previous model of the same manufacturing system. The second version of the model provided all of the results of the first version and more and is very easy to explain to the non-simulator for outside use. Data can be easily changed or added, and, if new operating strategies emerge, only the relevant subroutine need be changed or a new one implemented.

5. CONCLUSIONS

The simulation of a very complex electronics assembly plant provided the stimulus for developing a new way to approach the development of manufacturing models. An existing model was very costly to run and inflexible. As a result, a modular approach was created and an attempt made to develop a new model conforming to the structure outlined in this paper. By creating a model of a highly complex system based on this modular structure, it has been shown that the modular approach to the simulation of manufacturing systems is indeed feasible.

In addition to being feasible, use of the modular approach provided many benefits. The modular EAP model was much less costly to run, was developed much more quickly, is easier to explain to the non-simulation expert and is easier to modify. This approach should be feasible for any simulation language with discrete modeling capability.

The modular approach to simulation would be worthy of consideration for any simulation of a manufacturing system using the seven modules described in this paper. The application of the suggested modular structure to other simulation problems should also be explored.

REFERENCES

Pritsker AAB, Pegden CD (1979), Introduction to Simulation and SLAM, Systems Publishing Corp., West Lafayette, Indiana.

Shoaf SA, et al. (1981-1983), Various internal Westinghouse Research Reports.