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

A METHODOLOGY FOR DEVELOPMENT OF SIMULATION BASED PRODUCTION SCHEDULE GENERATION SYSTEMS

Knud Erik Wichmann

SIMOS Inc.
Simulation Modelling Scandinavia
Gl. Hovedgade 12, 3
DK-2970 Hørsholm, Denmark

ABSTRACT

The paper describes structured methods, techniques and experiences which enables us to rapidly tailor a simulation based planning or scheduling system to the specific needs of a company. It does not aim to describe an "off the shelf" scheduling system product.

The methods include the function modelling technique IDEF0 and the information modelling technique IDEF1. However, the simulation specific viewpoint and purpose of the modelling activity has made it necessary to modify the IDEF1 syntax and to incorporate additional notation for objects, rules and comments. In addition, flowcharts to describe logic and decision trees may be created to model conditions and actions. The methods is based on advanced simulation modelling techniques and relational database techniques. For the implementation, the approach is to use a general purpose simulation language combined with database methods to dynamically search and select orders from a pool of orders. Each order is an object defined by it's attributes. Additional information necessary for the order selection is contained in one or more databases maintained and updated by the simulation. The databases are initialized and generated from external files, which are drawn out of existing MRP systems and/or other company databases. The order selection is based on order-attributes as well as system status information, which are detected from the simulation model.

1. INTRODUCTION

Every company has a specific and unique way of integrating resources, transport- and material handling systems, control logic, operational policies and products and product mix. A simulation based scheduling system has obvious advantages, as it will reflect the specific structure and operation of the company to any necessary level of detail. [Grant 1986; Novels and Wichmann 1989; Davis & Jones 1988; Wichmann 1990]

Different scheduling rules have been developed to assist in achieving various conflicting operational objectives. Practical experience, however, suggests that these rules do not perform consistently from company to company. Often the rules adopted are specific to a company and are the result of a compromise based on past experience and experimentation.

Detailed computer simulation provides the potential for more accurate experimentation which enables analysts to establish the trade-offs between conflicting objectives on a company-by-company basis. Typically, the computer model would be constructed to enable an analyst to choose different operational objectives and monitor the performance of the scheduling rules required to meet the objectives.

For a particular company, this would result in the establishment of a unique "objective v scheduling rule relationship" enabling the choice of scheduling rule to proceed in a more accurate and reliable manner.

The scheduling rules can be a combination of job-release strategies and dispatch-rules [Stecke, K.E.and Solberg, J. 1977]) based on attributes of the orders and parts to be manufactured such as earliest start and due dates, process plans, setup and operation times, production requirements, and other attributes specific to the system at hand, (figure 1).

Further more, the scheduling rule combination will often include company specific heuristic rules, so that the final selection of the scheduling rule combination and the final dispatch decisions are constrained by the dynamics of the system status: queues and bottle necks, machine utilization, alternate routings, material handling system characteristics etc.

2. CHOICE OF SIMULATION LANGUAGE

The key to simulation based scheduling is the availability of a simulation language that can model systems to a sufficient level of detail and interface to external databases.

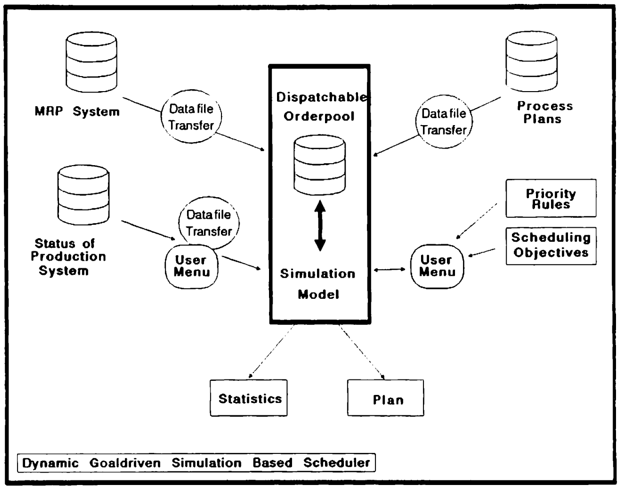

Figure 1. Dynamic Goal Driven Simulation Based Scheduler

The model must encompass all the needed specific components of the real system including scheduling rules and Control Logic. The specific nature of these components precludes the use of most task oriented simulation packages. A flexible simulation environment is required in order to adequately model complexities that include control logic and exception handling. Using the simulation language SIMAN/CINEMA the search conditions defining the scheduling rules and control logic can be expressions and logical conditions defined by the SIMAN blocks or user written subroutines. From our experience, a high quality animation is also required.

3. THE METHODOLOGY

Successful implementation relies on the methodology used by the modeller and his knowledge platform and expertise in the analysis of integrated systems. There are four major activities or phases in our approach (figure 2):

1. Problem Analysis

* Define the system objectives, analyze and understand the problem.
* Define the production goal of the factory, and the criteria on how to best solve the production and scheduling tasks.
* Develop conceptual models for the scheduling system, it's functionality, and the required data structures.
* Describe input and output and general system requirements.

2. Model development

* Specify requirements to model functionality
* Model implementation i.e. create SIMAN model, experiment, animation and user coded subroutines
* Pilot runs and model validation

3. Experimentation, integration analysis and prototype development

* Establish experimentation plan
* Establish schedule objective - scheduling rule relationship for the specific system

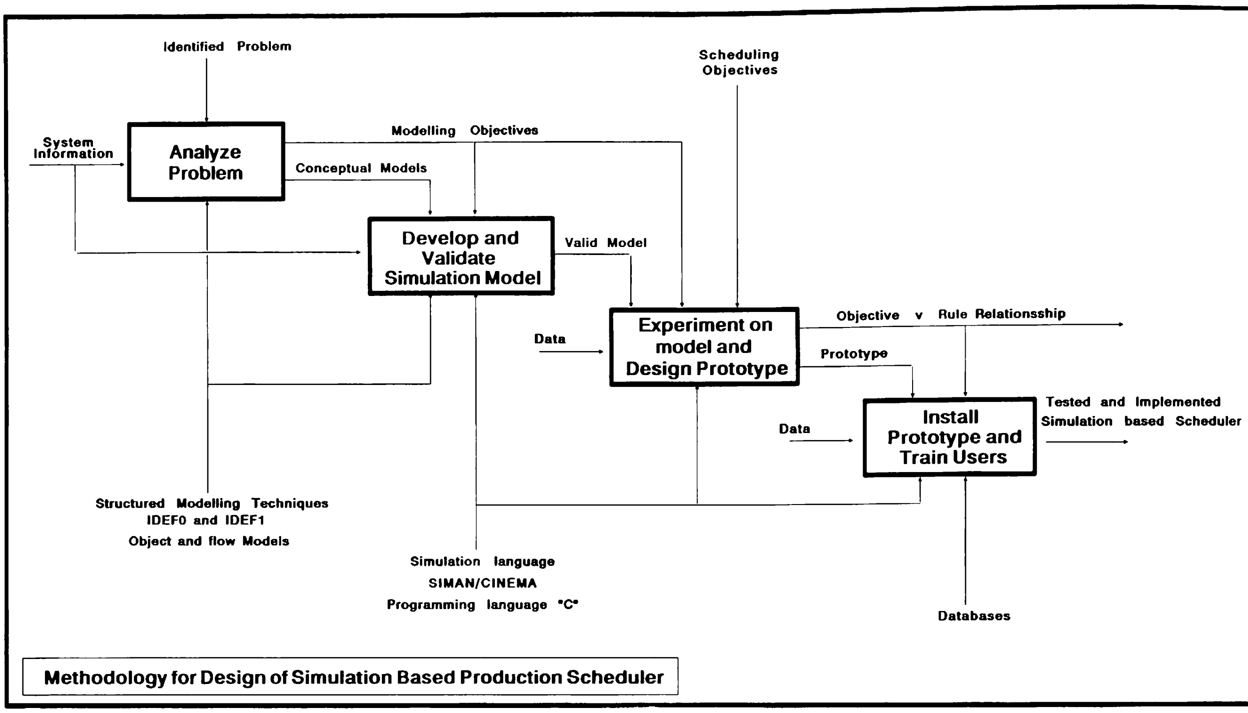

Figure 2. Methodology for Design of a Simulation Based Scheduler

* Establish and if possible formulate heuristic rules
* Implement the objective - rule relationship into the model and create a prototype with end-user menus.
* Test system prototype

4. Implementation, installation and training

* Tailor user interfaces and output reports
* Install system and train users

In the following we shall describe in more detail the techniques we use to rapidly establish good rapport between system developer and the company in order to efficieltly meet company specific objectives.

Ad. 1. Problem Analysis

A well defined objective and clear understanding of the problem is fundamental for the development. Therefore a technique to define and model the problem is of great importance.

The first step is to clarify the overall manufacturing goals and requirements which must be accomplished. This will normally be defined by management, but must be communicated to and understood by the modeller in its proper "industrial engineering context".

Next is needed a mechanism to provide for a consistent description of the structure and functionality of the system to be modelled and of the information to be used and processed by the system. This mechanism is provided by the function modelling technique IDEF0 or SADT and by the information modelling technique IDEF1 and by SIMOS' modifications of these techniques. It is beyond the scope of the paper to describe these technique in details, but we illustrate our use of them in the example. (For further details see [ICAM 1981]). It should be noted, that the original IDEF techniques are suggesting a large degree of formalism, which in some (CIM) cases may be required. However, we are primarily using the graphical part of the IDEF0 and IDEF1, as in most cases, keeping our objectives in mind, it is enough.

The modified IDEF1 syntax establishes a technique, which we call an "Object Model". Due to the simulation model viewpoint and purpose of the model, we have found, that a data model (Entity-Relationship model) in some cases cannot capture the information we need. We need to capture the rules which the objects are constrained or selected by, for example priority rules used by the scheduler. Additionally we draw attributes in their own boxes for improved communication between model builder and customer. In addition to these techniques, flowcharts describing logic and decision trees may be created to model conditions and actions.

The IDEF0 function models are used to :

- analyze the problem
- identify the functions taking place in the system
- identify the constraints, input, output and mechanisms for these functions
- establish a common understanding and platform for what is to be embedded in the model, i.e. the models functionality, control systems, etc.
- ease the communication between the manufacturing company and the system developer

After having identified the constraints for a function, we can detail the information needed and describe the required decision logic and control system. (other diagrams such as SIMAN block diagrams, decision trees, Petri nets, etc. may be used as well).

The function models shown in figure 2 and figure 3 are simple examples showing the principle. The arrows entering the box from above, indicate constraints or controlling information for the function. The arrows entering the box from the left are input to the function and arrows comming out from the right are output of the function. The arrows entering the box from below, are mechanisms for the function. A major advantage of the technique is that it is hierarchical. Each box (function) of the model can be "opened" or broken down on child diagrams into any level of detail. We have a simulation model viewpoint when drawing the function model. Doing this, we make explicit how the functions of the real system are modelled in the simulation.

The Data and Object models are used to identify the relevant data entities and their relationship to one another in order to create the various tables in the database containing order and parts information. The graphical data model facilitates the communication and knowledge acquisition, and

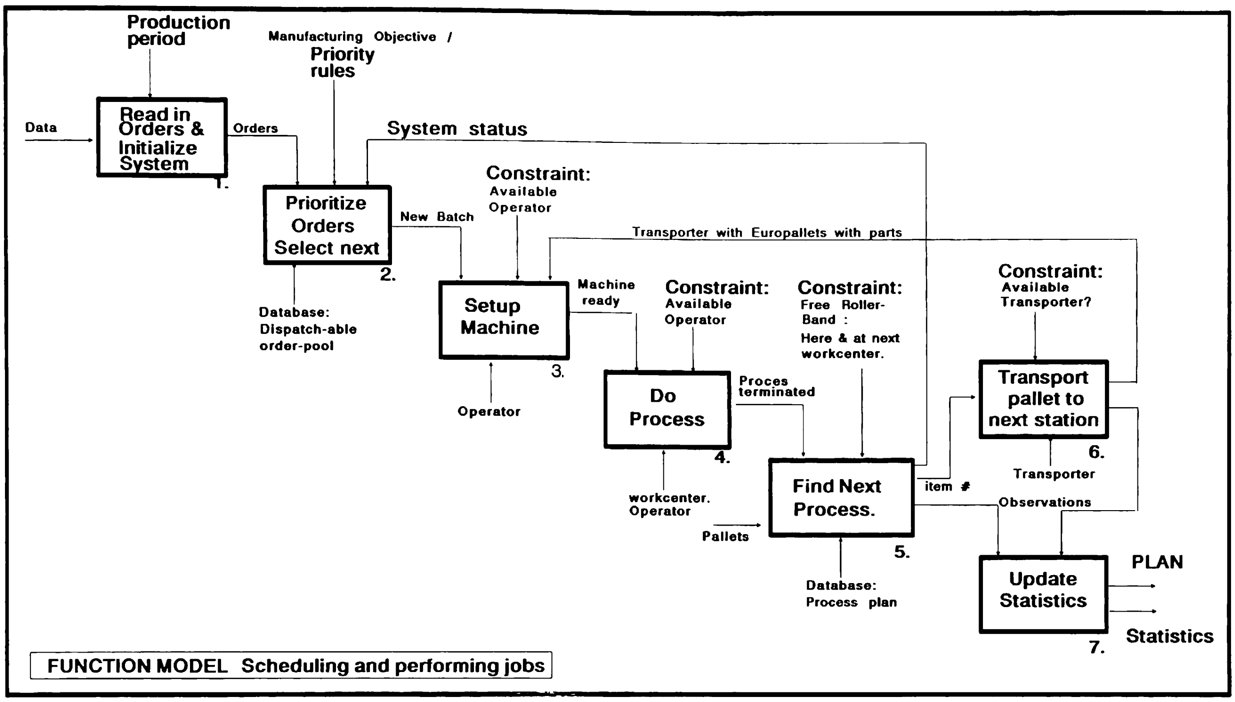

Figure 3. Function Model: Scheduling and Performing Jobs

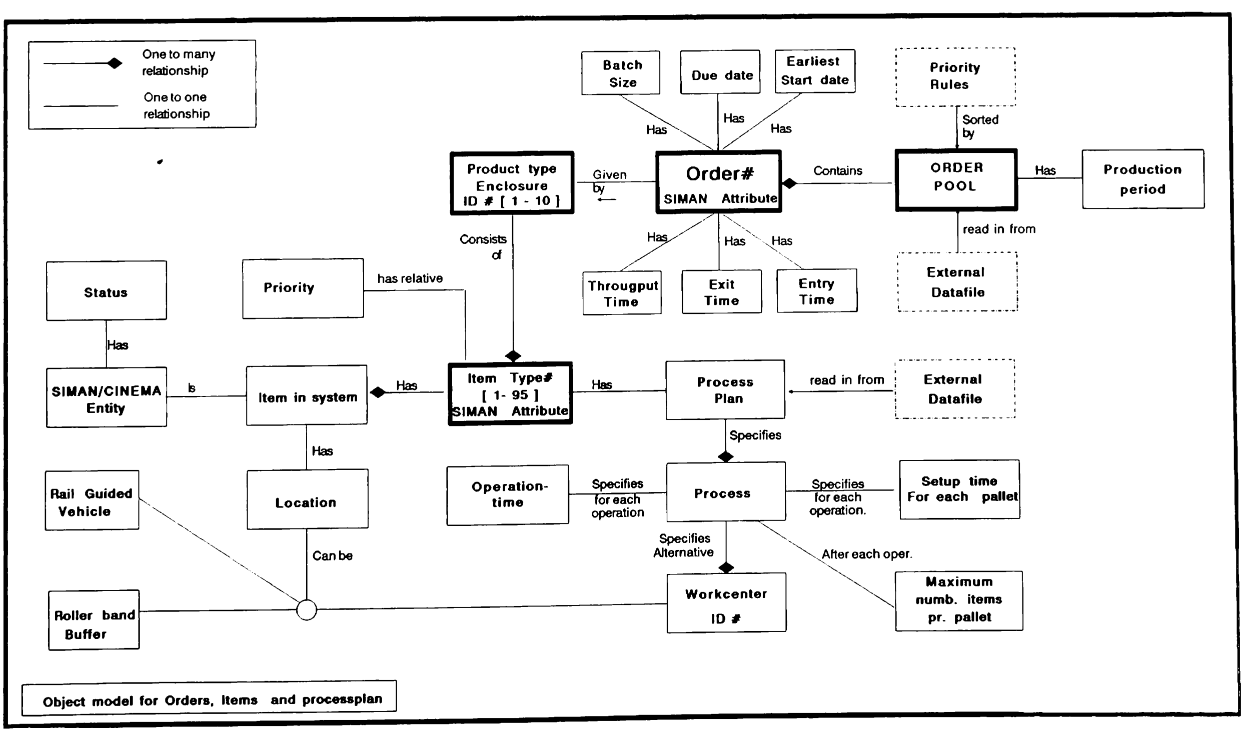

Figure 4. Object Model for Orders, Items, and Process Plan

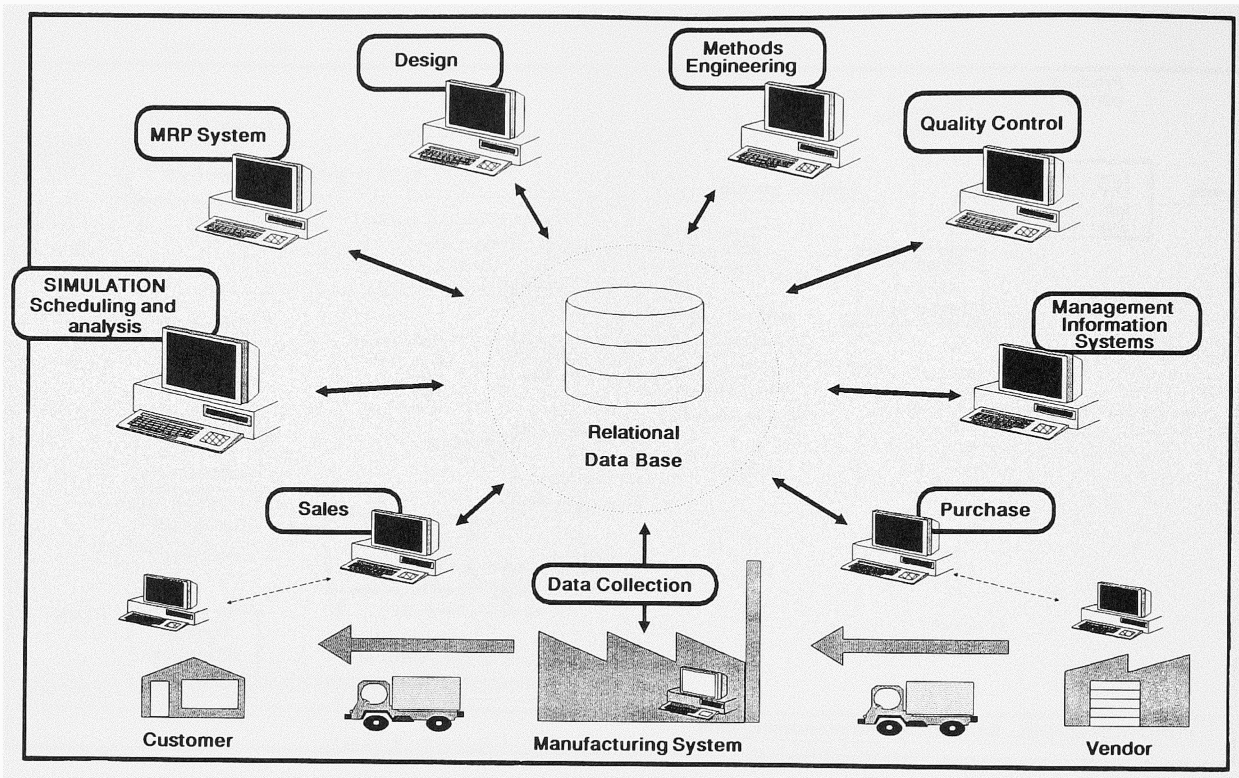

Figure 5. The Role of Simulation in CIM

establishes a common platform for discussions throughout the entire development process. The models thereby make explicit what information and data entities are to be included in the simulation model. The Object Model contains more explicit information about the implementation of data and objects in the simulation model. The Object Modelling syntax is currently being further developed. The scope of this effort is to create a tool, which is dedicated to the particular type of projects which are the subject for this paper. The Object Model shown in figure 4 is a simple version to illustrate the principle.

From the models, we also define uniquely the files and the data formats to be retrieved from external databases. Like-wise we define the format of the output files to be generated by the simulation model. We even use the models to indicate how the SIMAN representation of the information should be: we can indicate local and global variables, attributes, expressions etc. These techniques are excellent to define a structure in a way which establishes the best compromise between :

1. design criteria such as no redundancy in the data and
2. constraints, that may eventually be imposed by existing hardware or software, for example an existing MRP system providing order data to the simulation model, and
3. constraints that may eventually exist because of the simulation language or the way it is being used.

Ad. 2. Model development

These activities contain basically the same steps as in any other simulation model development. As the scope of the model is expanded, some additional knowledge and techniques are necessary. Assuming that the simulation language is capable of importing data from external files and databases, a definition of the data structures and attributes of the entities to be simulated is of special importance. Relational database techniques and theories are necessary to define the data structures in such a way, that the interaction with existing databases can be performed effectively and flexibly.

The simulation model should be designed in such a way, that it can read data in a neutral (f.ex. simple ASCII) format. When the data has been loaded into memory, the simulation model accesses the data. With the simulation language SIMAN it is possible to search and access data in a way similar to search in relational databases. This is the capability, which is necessary in order to select and prioritize jobs based on their attributes. The attributes can be any one defined within the context of a specific manufacturing system. We actually use SIMAN commands (blocks) to search, though SIMAN does not incorporate an actual query language like SQL. In some cases we link SIMAN with subroutines written in "C".

When defining and implementing the scheduling rules, it should be relatively easy to change the selection criteria to be used in the search and job selection. The model should be developed to facilitate this particular requirement.

Ad. 3. Experimentation and integration analysis

As different scheduling rules and search conditions are tested and evaluated against relevant scheduling objectives, they are included in the model. (provided they have a performance which fulfills the requirements). Doing this, we gradually build up a rulebase in the model, each rule corresponding to a defined schedule objective. Then, at a later stage, when the user needs to generate a schedule according to a selected objective, he can choose between the various blocks of the model which includes different code for different search criteria, by preselecting one from a menu before every experiment. This is a way to implement the objective - rule relationship into the model. In the process of doing this, we prototype end-user menus.

The rulebase can be expanded as the knowledge of the system increases and, particularly, new heuristics may be defined. If, at later stage, some changes are made in the manufacturing system structure, this should not impose any problems, as the model - which is a computer replica of the system - can be modified.

With an objective to use the scheduling system as part of a shop floor control system, information from shop-floor data collection systems must be added to initialize the model with actual status information. This data can be transferred to the model-database via keyboard entry or electronic data transfer in a neutral file format. Menus must then be designed for easy update of the model. With this level of ambition, we get a view of the future role of simulation in CIM, Computer Integrated Manufacturing, (figure 5).

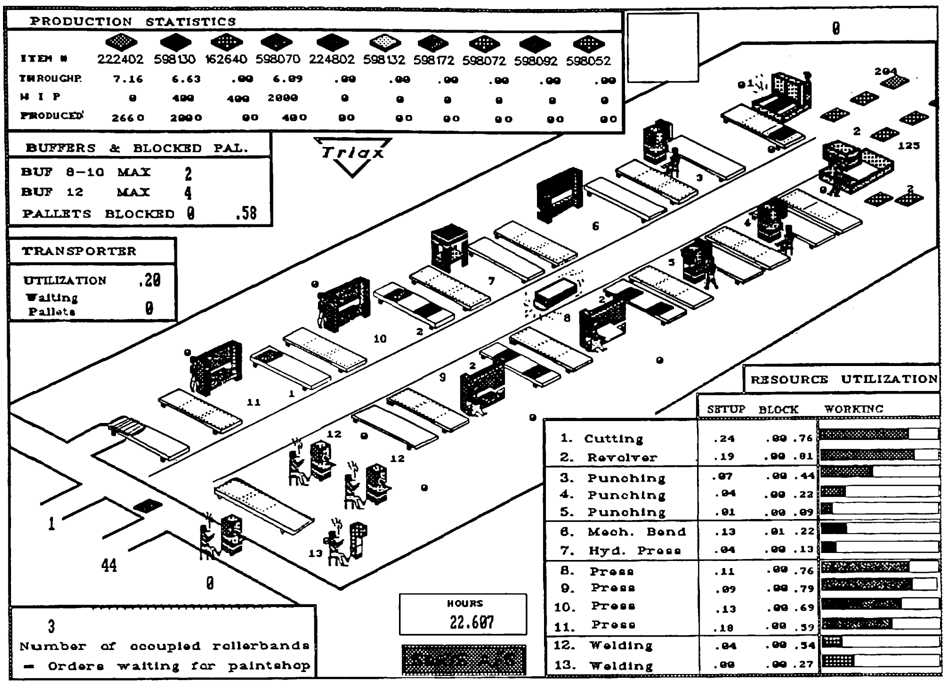

Figure 6. CINEMA Layout of Electrical Enclosure Shop

Ad. 4. Implementation, installation and training

Based on experiences and critique of the prototype, user interfaces and output reports are modified. Depending of the level of sophistication, e.g. the presence of computer network etc. the system can be connected and installed at the end-users desk. Of course the system can be used to educate shopfloor managers as well as being a tool for day to day scheduling.

4. Using the scheduler.

The user starts the program. From a menu he selects the datafiles containing part and order information. From another menu he initializes the system with current capacity available etc. From a third menu he selects the goal - and hence the scheduling rules to use in the simulation. (figure 1).

The simulation generates a file tailored to the specific requirements, for example containing the order#, the product type, the time at which the order had entered the machines, when the order was finished at the machines, the order throughput time, lateness etc. Observations on capacity utilization, WIP, etc. are gathered in order to have a means of evaluating the goodness of the plan.

The user may want to change the rules selected and change other parameters, try other product mixes and run the simulation again. By comparing the results of the runs, he finally commits the operations to a schedule.

If, after a while, some unforeseen circumstances at the shop-floor occurs like a machine breakdown, or if a rush order needs to be processed before anything else, the user can read in the new conditions into the system, and run the simulation again to create a new schedule and to predict the effects of the event, before decisions are taken.

4. THE ADVANTAGE OF A GENERAL "SET OF TOOLS" TO GENERATE SPECIFIC SOLUTIONS

Applying this approach, we can develop A simulation based scheduling system to meet the specific requirements of a company, and thus achieve substantial advantages compared to a standard product, which includes:

1. Ability to formulate the specific project goals and level of ambition for the scheduling system.

2. Ability to create the simulation model, the user interface and the output to the necessary and sufficient level of detail, as regards the integrated system, and the scheduling requirements.

3. Ability to link with existing computer equipment and databases, both upstream to get data from e.g. planning or MRP systems and downstream to data on shop-floor resources and timely information on production and part status

5. EXAMPLE 1: MANUFACTURING SYSTEMS

We have used this method to develop simulation models with a scheduling interface for a variety of completely different systems. An example is a facility to manufacture electrical enclosures. These are made by cutting, punching and bending sheet metal components and welding them together.

The new integrated manufacturing system (figure 6) was designed with a number of manual and semi-automatic punching, shearing, bending and welding equipment, placed along the trail of an RGV, which transports batches of parts through the cell according to the process plans of the different items. These are sheet metal components on euro-pallets with 10 different products consisting of 90 different items each with a unique route and process time and other characteristics.

The model proved the system's sensitivity to variations in product mix, giving possibility to develop scheduling rules to determine the order in which to dispatch the various batches into the system. The rules are based on attributes of the products/orders defined by the data structure. These were total processing time, setup time, processing time on bottleneck machines, customer priority, due-date and earliest starting time. All we need to do, is to establish the correspondence between the data elements in the data structure and the SIMAN attribute number assigned to contain the data. If we should need to modify the data structures, it would only require minor modifications in the model. Using any of the rules, the actual time of searching for and loading a selected batch are constrained by the current status of the resources within the cell (i.e. machines, buffer levels).

On average it takes one minute to simulate a 4 weeks production period, two shifts a day, of 40 different order#'s, averaging 300 products pr. order# with a mix of 10 different types of enclosures with a total of 90 different types of items, each with a unique route, process time and other characteristics.

The simulation generates a file (table 1), which tabulates the progress of each batch through the system containing the order#, the product type, the time of entry and exit from the cell, the due-date and the order throughput time. From this the user could see which orders would be delivered by the end of the planning period. Observations on capacity utilization, WIP, etc. are gathered in order to have a means of evaluating the goodness of the plan.

Table 1. Showing Part of an Output File

OUTPUT

Order#	Part#	In syst	Out syst	Duedate	Throughput tim
1,	1	89:11:20:00.00	89:11:20:22.49	89:11:20	22.49
2,	6	89:11:20:00.00	89:11:21:01.41	89:11:20	25.41
3,	2	89:11:20:23.00	89:11:21:11.40	89:11:21	12.40

Other examples includes

- A Cable manufacturing factory including machinery such as wire drawing, insulation and stranding machinery and an integrated storage and material handling system operating according to a JIT philosophy.

- An order producing paint shop including automated painting lines and 4 manual stations for various filling and grinding operations.

- Lines in a cell for manufacturing electric motors according to a cyclic planning scheme.

6. EXAMPLE 2: LOGISTIC SYSTEM

One further example of using simulation listed below, have proved to be successful in solving particular planning problems previously unresolvable using computers. As basically the same methods and techniques has been used, we shall briefly describe the systems:

An airport food delivery system loading catering directly to planes (more than 230 planes pr. day) during the turnround period of flights (time between flight's Arrival time 'ETA' and departure time 'STD').

A Highloader is a rather expensive, special truck with hydraulic lifting devices which is used to load catering into the doors (galleys) of passenger aircrafts. They drive from docks near the flight-kitchen with food (which is placed in little carts) to the flights. Since the turnround period often is very short, the Highloaders must be scheduled to meet the flight as soon as it has landed. Further more, they must if possible carry goods for up to three flights pr. trip to the airport. The main criterion for the schedule is, with minimal number of trucks service all flights so that no flight are being delayed due to lack of capacity in the catering business.

The planning of this system is - being controlled by the traffic plan and eventual delays and other disturbances - an exercise of extreme tight margins - literally a "just in time" problem. The planners who manage the highloaders uses rules and practices which of course are an important brick in the system.

As it could actually mean 10-20 % Highloaders more or less (using a "good" planner vs. a not so good), it was important to analyze the practices and rules of the "best planner". This was difficult, since his expertise were "implicit" in his mind being difficult to express and formalize. But it was possible to formalize these rules and implement them in SIMAN using search in a dynamically updated "order-pool" of flights to serve. The key attributes were parameters like ETA and STD and 'LLT' (meaning "Last loading time"). Further more: distances / airport position, driving time, HILO capacity as a maximum number of carts, type of flight to load, were important attributes, which were stored as a relational database.

Output from the airport model would tabulate the time the highloaders were scheduled to leave from which outbound dock, the total number of carts and the Flight # (up to three different) and door # to which to load the carts. Other output statistics are Highloader cycle time, average number of carts pr. trip, utilization of docks. Output files also shows, which flights were loaded how many minutes too late.

7. CONCLUSIONS

We have described a method to create a dynamic goal driven simulation based scheduler. The scheduler will take gross production requirements as an input and generate a detailed plan for sequencing and dispatching the orders. The method has many advantages compared to "off the shelf" products, still offering rapid tailoring and a competitive price. One problem with the approach may be the expertise required, i.e. an expertise which many companies may not have "in house". The major categories of knowledge required are the following, (to some extent in order of importance):

1. Industrial Engineering; theory and experience, particularly insight in potential problems and benefits of integrated systems.

2. Simulation modelling and statistics; Theory and project experience.

3. Advanced skills in using a flexible simulation language.

4. Structured development modelling techniques.

5. Database knowledge.

6. A computer language like "C" .

The scheduling system must contain specific knowledge regarding the system to be modelled. Thus the modeller needs a capability to communicate with people in a company at various levels, which posses this knowledge. Some times, even "knowledge acquisition" techniques borrowed from the "AI" profession are useful when attempting to transform implicit knowledge of some production system into heuristics in a form, which the computer can deal with.

REFERENCES

Davis, W.J. and A.T. Jones (1988), "A Real-time Production Scheduler for a Stochastic Manufacturing Environment," *International Journal of Computer Integrated Manufacturing*.

Grant, H. (1986), "Production Scheduling Using Simulation Technology," In *Proceedings of the Second International Conference on Simulation in Manufacturing*, 129-137.

ICAM (1981), "Integrated Computer Aided Manufacturing," IDEF0 and IDEF1 Manuals, USAF Materials Laboratory.

Novels, M.D. and K.E. Wichmann (1989), "Simulation as an Online Scheduling Decision Support Tool in Factories and Process Plant," In *Proceedings of European Simulation Congress*.

Stecke, K.E. and J. Solberg (1977), "The Optimal Planning of Computerized Manufacturing Systems," NFS Grant Number 15256, Report Number 10, Purdue University, West Lafayette, IN.

Wichmann, K.E. (1990), "Simulation Based Production Schedule Generation," *International Journal of Production Planning and Control*.