Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

SIMULATION-BASED FINITE CAPACITY SCHEDULING: A CASE STUDY

Maureen T. Rosenwinkel

John Crane, Inc.
6400 W. Oakton
Morton Grove
Illinois 60053
U.S.A.

Paul Rogers

Division of Manufacturing Engineering
Department of Mechanical Engineering
The University of Calgary
Calgary, Alberta, T2N 1N4
CANADA

ABSTRACT

This paper describes the selection and implementation of a simulation-based finite capacity scheduling package for a medium scale manufacturing company, John Crane, Inc. of Chicago, Illinois. The paper first discusses the characteristics and constraints of the manufacturing environment at the company and describes a vision of a finite capacity scheduling system aimed at working in this environment. From the vision, system requirements are defined and used to develop evaluation criteria against which alternative commercial packages are then measured, leading to the selection of the most appropriate package. The final two sections of the paper discuss the system implementation process, highlighting the steps needed for success and the lessons learned, and map out the future development of the system within John Crane.

1 INTRODUCTION

John Crane, Inc. is the world's largest designer and manufacturer of engineered sealing systems. Since 1927, an emphasis on research and development for new product lines, as well as enhancing existing ones, has helped give John Crane a competitive advantage in satisfying the needs of its customers. In the 1980's, John Crane revolutionized centrifugal compressor sealing with the Type 28 dry-running gas seal. This seal design solves costly gas containment problems with unique spiral groove technology. John Crane's world headquarters, which is located in Morton Grove, Illinois is the primary manufacturer of Type 28 seals for North America. A 35,000 square foot facility has been dedicated for the manufacture, test and repair of the Type 28 dry gas seal.

This paper deals with the machining cell that manufactures the Type 28 seal components. The cell consists of an electronic discharge machine, four computer numerically controlled lathes, two computer numerically controlled mills, four manual lathes, and two manual mills.

Type 28 Seals are custom engineered for each application. Orders are typically low volume (1 to 5 seals). With anywhere from 10 to 20 manufactured components per seal, and approximately 10 open seal orders at any one time, there is a wide variety of components on the machine floor at all times.

Predicting completion times, backlogs, and producing schedules are some of the many duties of the shop supervisor. Several home-grown approaches, from spreadsheets to databases, have been developed to help with these tasks. However, none of these were able to reliably answer some of the more critical manufacturing questions such as:

- Which orders will be late?
- Which operations will cause an order to be late?
- Are there bottleneck workstations?
- Can overtime at bottlenecks help to meet demand?
- What should each workcenter's job sequence be to adhere to the global plan?
- What is the impact of expediting an existing order?
- Is there any excess capacity and if so, where and when does it occur?

2 THE DRIVING VISION

With orders increasing regularly and the launching of a high volume T-28 seal, John Crane's management felt that a formal scheduling system was necessary. It hoped that a scheduling system would not only answer these questions but would also lead to reductions in work in process, overtime costs, setup, planning and scheduling time. The system requirements were defined as follows:

1) Finite capacity.
   John Crane needed to be able to consider the capacity of the machine shop, the machines that are working and the number of hours each machine and operator is available as well as recognize jobs already in the system.
2) Standalone system.
   Management wanted a system that could operate independently of the company mainframe system. The company is moving toward a networked system with file servers which permits each area within the company to use systems that work best for it. John Crane already had accounting, order entry, purchasing and other existing systems so there was no reason to duplicate these systems in adding on finite scheduling functionality.
3) Ease of use.
   The system needed to be able to predict manufacturing completion dates and generate a schedule in an easy to understand format. An additional required feature was the ability to evaluate various scheduling scenarios. John Crane's management wanted to be able to see the affect of adding overtime, changes in release dates, etc.
4) Scheduling both machines and personnel.
   The desired system needed to be able to recognize that in order to run a job there must be an available machine and an available operator. At first this sounds obvious but with more machines than operators, the software needed to be able to select both the machine and operator.
5) Handle managerial preferences.
   To complicate the issue further, many operators are qualified to operate more than one type of machine, but the supervisor has preferences. For example Dave and John can run both manual and automatic lathes. Due to speed and quality considerations, the supervisor prefers to have Dave run an automatic lathe and John run a manual lathe. The software needed to be able to account for this.
6) Open and flexible system.
   The system needed to be able to interface with existing computer systems and a future shop floor data collection system with the minimum duplication of effort. The system also needed to be capable of expansion to schedule non-machining areas such as the Assembly and Test Departments.
7) Established software supplier.
   Management wanted to ensure they were investing in a company that had a stable history, with a solid customer base, that would be able to provide future support. A supplier with local support would be a bonus but not a necessity.

At this point the architecture of the required system was still relatively undefined. However, the input and output requirements were now better understood (these are described graphically in Figure 1).

3 PACKAGE SELECTION

Having defined a vision for the system, candidate packages and suppliers were identified through technical journals, referrals and a TEL-TECH vendor search. Thirty-six companies were initially identified as suppliers of scheduling software and a two phase selection process was used to make a final choice from amongst these alternatives. Phase I took the form of a brief package evaluation and resulted in a simple "pass/fail"

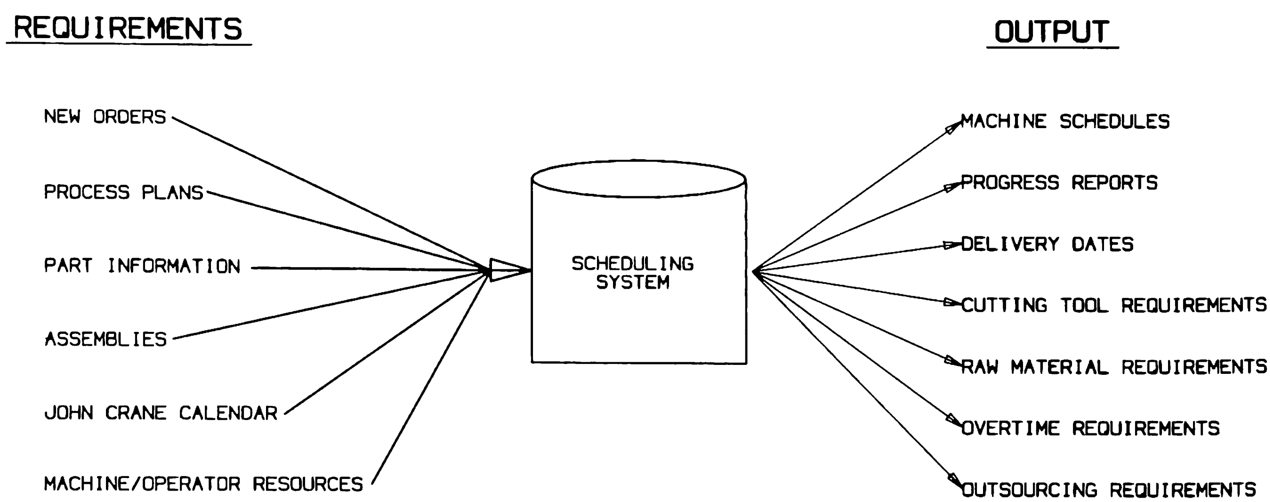

Figure 1: Scheduling System Input Requirements and Data Output

classification. Phase II evaluated in greater detail how well each of the "passing" software packages could meet a number of important criteria.

3.1 Phase I

In this phase the potential suppliers and packages were evaluated to determine if they were capable of meeting the criteria listed below:

- Finite capacity
- Standalone
- Predict completion dates
- Produce a schedule
- Recognize multiple constraints
- Evaluate various scenarios
- Interface with existing/future systems
- Schedule non-machining areas.

A supplier needed to meet all of these requirements in order to be considered in phase II. After phase I, the number of candidates was reduced to thirteen.

3.2 Phase II

In this phase a point system was used to evaluate each of the suppliers against four major criteria. The points were assigned, somewhat subjectively, based on personal and management opinion after conversations with the vendors, literature reviews, and viewing demonstrations of some of the products. This approach was felt to offer the best balance between the desire to make a wise choice and the need to avoid expending too much effort on the evaluation process.

The four criteria were equally weighted and scores for each package for each criterion were determined on a scale of 1 to 5 (5 being the best). Descriptions of the four criteria are as follows:

- Level of detail of the scheduling model.
  This took into consideration the amount of information the scheduling system could consider at one time. For example, a system on a DOS platform cannot digest as much information as a system operating under OS/2. Another dimension here related to whether schedules were generated by some form of optimization algorithm or through simulation? Optimization models aim to produce a schedule which best satisfies some objective function. This function will be for a set of variables within a given set of constraints. However if the assumptions of the system being modeled do not accurately fit the assumptions of the optimization technique, the results will not be valid for the desired system. Generally, optimization models require restrictive assumptions to be made concerning the given system, while simulation permits a more accurate model of a system to be developed (Starks 1991). Also considered in this evaluation category is the sophistication of the model. How intelligent is the model? Can it select operators based on supervisor preferences?
- Ease of use.
  The people who were intended to use the system were the shop supervisor and the manufacturing scheduler, neither of whom has had formal computer training. A friendly system that did not require a strong background or substantial previous experience in scheduling or computers was a necessity. The system needed to produce easy to interpret results and permit evaluation of various alternatives in a reasonable time frame (no more than 15 minutes per alternative.)
- Flexibility.
  The long term plan was to start scheduling the Machining department and then schedule the other departments: Assembly, Test, Design, and eventually Marketing. This would facilitate Marketing to more accurately predict completion dates for potential orders as well as determine the effect of new orders on the production schedule for existing orders (Rogers and Flanagan 1991). The selected system needed to be capable of expansion in this manner.
- Established supplier.
  Relevant factors here were the supplier's existing customer base and what typical customers had to say about the supplier and its product? A proven track record was an important requirement.

Once each supplier's product was assigned a point value for each category, the average score for each product was calculated and used to make a final decision. It is important to note that not all of the packages evaluated were simulation-based and that they were not all developed exclusively for scheduling. Some of them were general purpose simulation packages which have been expanded in one or more applications to be used as finite scheduling tools. Because of a lack of scheduling-specific functionality, these general purpose packages were not expected to score very well.

Cost was used as a consideration in the final selection stages. After the top three packages were identified, costs were obtained. Software purchase price, cost of training and cost of ongoing support were considered. The prices among the top vendors were found to be so similar as not to influence the final selection. Figure 2 shows a summary of the final evaluation of the potential scheduling systems.

	LEVEL OF DETAIL OF SCEDULING MODEL	EASE OF USE	FLEXIBILITY	ESTABLISHED SUPPLIER	AVERAGE
AT&T ISTEL WITNESS	1	3	3	4	2.75
FOCUSED APPROACHED, INC FOCUSED APPROACH	2	3	3	1	2.25
ICAM TECHNOLOGIES CORP. CAN-FAMS	2	3	2	3	2.50
INSIGHT LOGISTICS, LTD. INORDA	3	3	3	1	2.50
MANUFACTURING MANAGEMENT SYSTEMS CMS - CAPACITY MANAGEMENT SYSTEMS	3	3	3	3	3.00
MINDBRIDGE REALTRAC	2	3	2	2	2.25
PRITSKER FACTOR	4	4	4	5	4.25
RAVEN TECHNOLOGIES MASTER WORKS-FLOW MANAGEMENT	2	3	2	1	2.00
RWT AHP LEITSTAND	2	4	3	3	3.00
MANUGISTICS, INC SCHEDULER	2	3	2	1	2.00
SYSTMS MODELING SIMAN	1	1	3	5	2.50
WATERLOO MFG. TACTIC THE SCHEDULER'S ASSISTANT	3	3	4	1	2.75
WYMAN ASSOCIATES, INC. JOBTIME PLUS	3	4	3	4	3.50

Figure 2: Summary of Evaluation of Alternative Scheduling Packages

Phase II of the selection process identified FACTOR, a simulation-based scheduling system from Pritsker Corporation (Lilegdon 1992) as the best solution for John Crane's application.

4 SYSTEM IMPLEMENTATION

After FACTOR was selected, a time frame of three months was established to get the system up and running. The first step was the definition of a system architecture to provide a means for handling the information to drive FACTOR, which is shown in Figure 3. The required input information was classified into two categories. The first was information that would be changing regularly and the second was information that would remain relatively constant.

The class 1 data needs to be updated every time a simulation is run. It was determined that the best way to handle class 1 data was to maintain it in a database separate from FACTOR and then transfer the files to FACTOR prior to running a simulation. To make use of existing software, it was decided to develop a database using R:BASE, a relational database software from Microrim. The class 2 data does not change, therefore it was decided to have this information reside permanently within FACTOR.

Class 1

- Order data
- Part data
- Process plans
- Order status
- Machine status
- Operator status
- Bill of Material

Class 2

- Resource definition
- Shift schedules
- Company business calendar

The new scheduling database consists of twenty one tables of information. A menu driven application was developed to manipulate computer forms for entering and editing information in each table. Great care was taken to eliminate data duplication and to verify the quality of the data as it was being entered to reduce complications once the information is transferred to FACTOR. The database was developed with ease of use in mind and utilized existing paper forms as models for computerized forms. Figure 4 illustrates the database structure designed to support the FACTOR scheduling software.

Process plans are created in the R:Base database by selecting a master process plan to copy and modify.

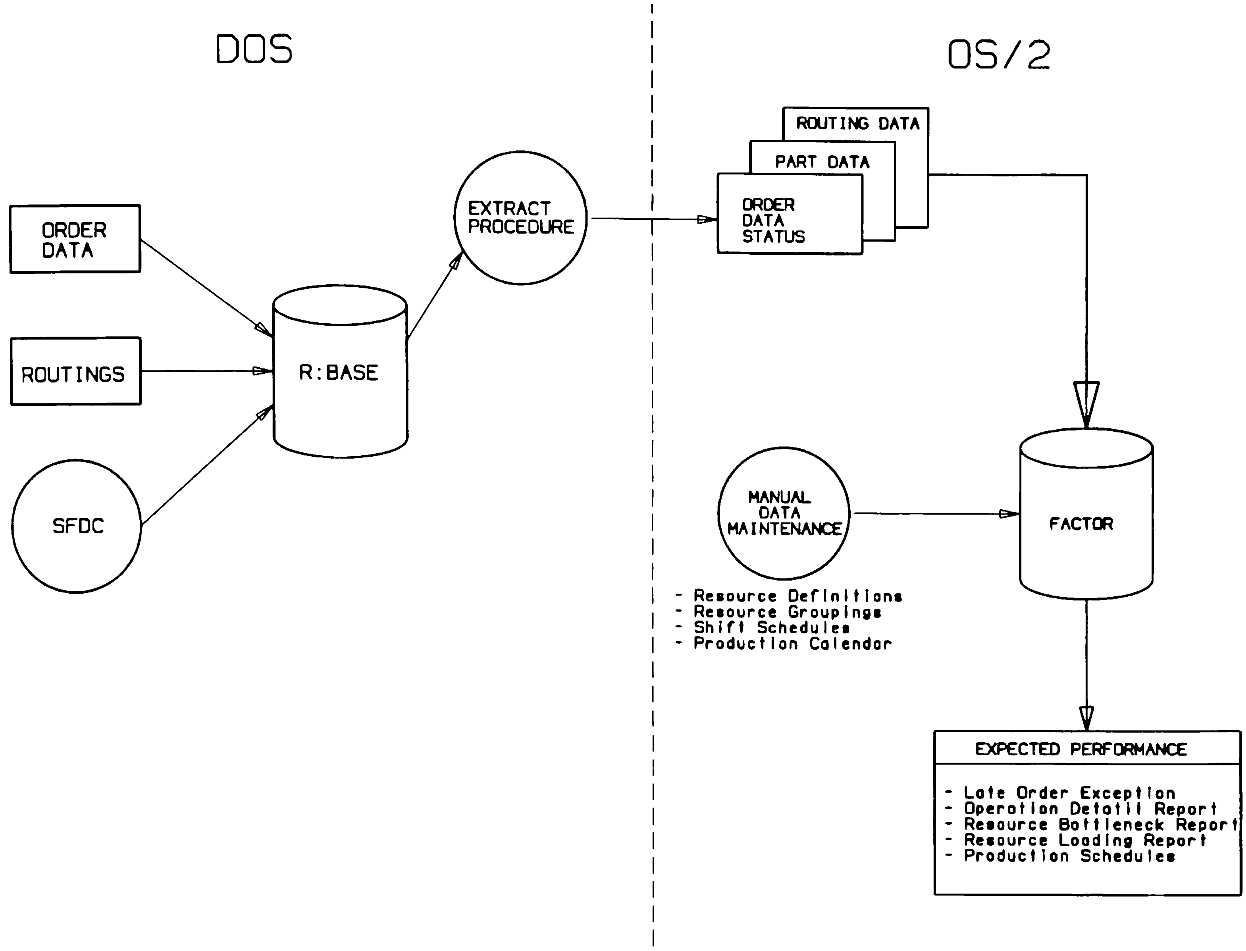

Figure 3: System Architecture to Support FACTOR

Modifying a process plan will include entering the information for a part including size, material, drawing revision, and the run time for each operation. During these modifications, the setup times from the master process plans are verified. Setup times were estimated from past experience. Run times were established though the numerical control programming system, with run times automatically calculated as programs were generated. The manufacturing engineers use this information when creating a part process plan. When new orders arrive they are entered into the database along with their release date and due date. An additional copy of the process plan is created for each factory work order. This will permit alternate process plans for the same part being made under different factory work orders.

Order status and machine status were the only pieces of information missing. Currently the manufacturing scheduler walks around the shop floor, identifies the status of parts and machines, records them in her log book, then updates an already existing R:Base database. It sounds crude but it's effective. It was decided to continue this practice with a slight modification. The status of each factory work order would be entered into the new scheduling database. This scenario will create a snapshot of the manufacturing facility. The facility will continue to make progress on components while the model will only predict this progress. For practical purposes, if the information is updated in the database within a few hours of when the simulation is run, the changes to the system will not be enough to affect the output of the model.

Once all the pieces were identified, (the database, FACTOR, and part status updating method) putting the puzzle together began. Two representatives from Pritsker Corporation came to John Crane for four days to train the system users and developers on how FACTOR works and to help develop a model of the John Crane system. The machining scheduler, shop supervisor, and system developer participated in the training. The training was made up of lectures, independent work sessions, and interactive discussions on the vari-

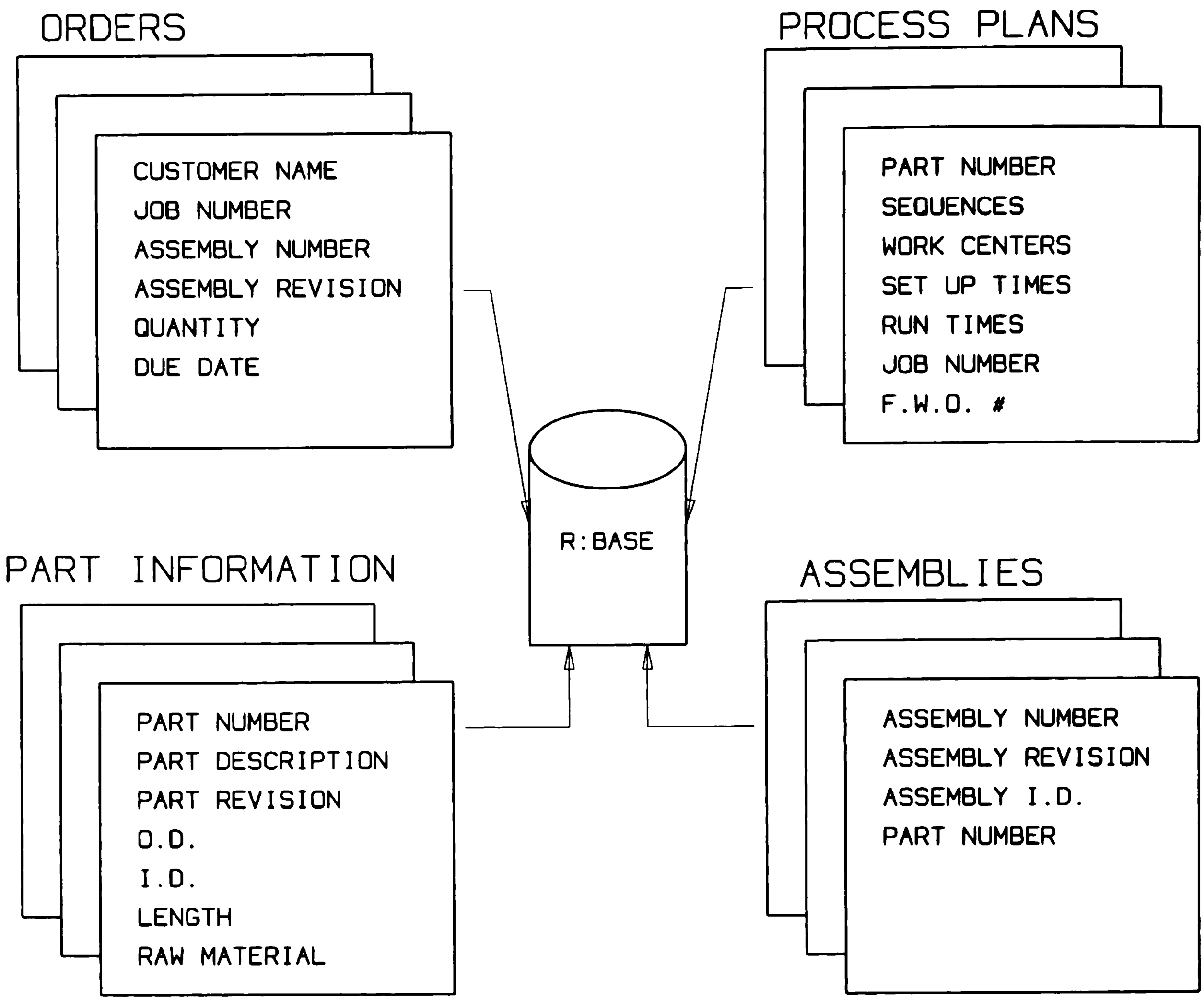

Figure 4: Scheduling Database Structure

ous ways to model the John Crane system. Pritsker's emphasis was on teaching the tools needed to make the software work.

Half a day of training was used to enter the class 2 information into FACTOR. An entire day was dedicated to identifying the data formats to drive FACTOR and where this data would be pulled from. The scheduling database was developed to be used for the scheduling software, so most of the information was readily available. However, the training course identified pieces of information that were missing from the new database such as job priorities and release dates. However, since the system was not yet in use, all that was required to add this information to the database was to modify the appropriate tables to hold the information and to add data entry spaces onto the corresponding computerized forms.

After the training program was complete, system development picked up speed. To facilitate data translation from one system to the other, an extraction procedure was developed. The standard R:Base report generator was used to develop three reports (Parts, Orders, and Process Plans) that would contain all the information necessary from the database. A subroutine was written in RBEDIT, the R:Base programming language, to execute these reports and to print each one of them to a file in the scheduling directory. R:Base is run under DOS and FACTOR is run under OS/2 so a dual boot function was used to permit transferring the files from DOS to OS/2. The files containing the reports could then be imported directly into FACTOR.

4.1 System Usage

The intended use of the system is described as follows. When a new schedule is needed the scheduler updates the part and resource statuses. Then, an extracting procedure is executed and the output is loaded into

FACTOR. The simulation is run and reports are generated in a matter of minutes. The FACTOR output report lists the dates each job is expected to complete. If all the jobs are not predicted to complete on time, different scenarios can be evaluated. FACTOR has easy to use interfaces that permit adjusting work schedules, order release dates, sequencing rules, and many other options. New schedules can be generated for each alternative and compared to select the best option.

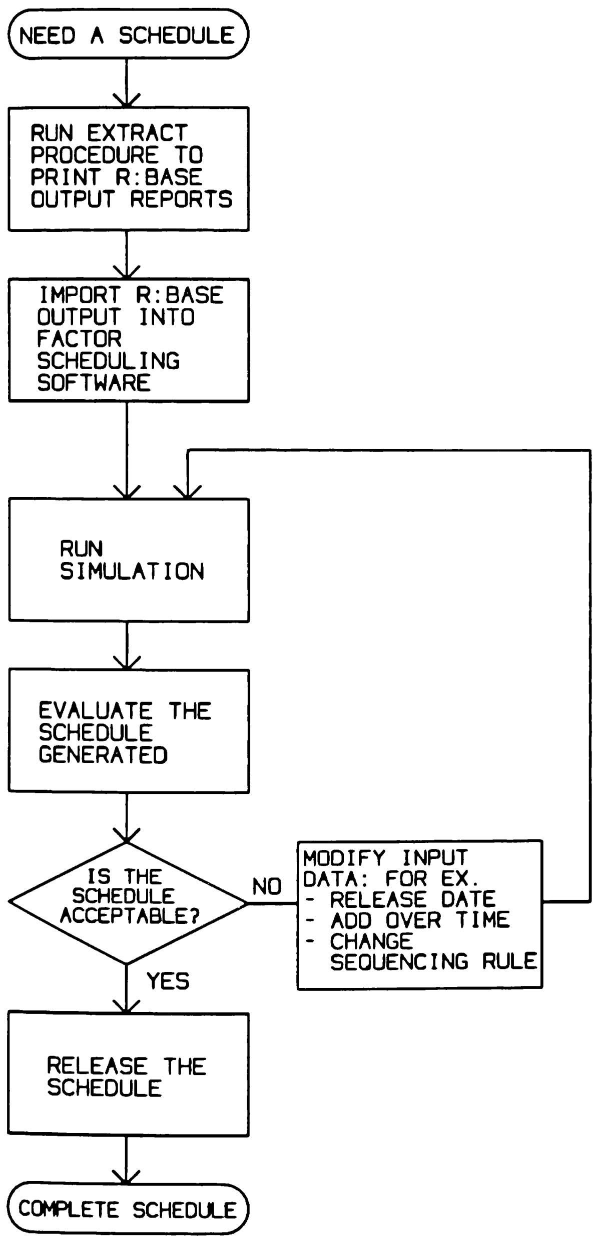

Figure 5: Flow Diagram of Scheduling System Execution

Initially it is intended to run the FACTOR simulation on a daily basis since in other environments where production times are low and schedule priorities are constantly changing, a rescheduling has been found to be necessary every day (Bauer 1991). However, once experience with the system is gained it is anticipated that FACTOR will be run only two or three times a week due to the nature of the Type 28 manufacturing group. Figure 5 shows a flow diagram of the system execution.

At the time of writing this paper, the database is completely defined and the system interfaces are fully developed. Sample schedules with real-life scenarios have been generated and the next step is to move to actual use of the system to schedule the shop floor.

4.2 Tips and Lessons Learned

This section lists the main lessons learned to date during the system selection and development process. This list is sure to lengthen as experience with using FACTOR to schedule the real system is gained.

- Involve the users from the start.
  This increases the users' ownership of the system and will help identify the "true needs" of the system. The users are the ones that can make or break the system so it is particularly important to make them be a part of its development.
- Avoid over-dependence on your supplier.
  Shoot for a canned solution with the minimum of customization since this will facilitate easier changes down the road. Many of the scheduling software companies today are offering special consulting or training with the software. Take advantage of their experience initially and use what you learn to develop your system so that as your system requires changes, you are experienced to handle them. Also, when the users and the developers learn together, developing the system becomes more of a team approach and thus reduces the users' fear of the system. Team training will also eliminate the vulnerability associated with there being only one in-house "expert".
- Maintain flexibility.
  Make sure that the software is not finicky about information formats and that interfaces to your existing and future systems can be established. A software package that will not accept extra spaces or is case sensitive may cause many hours of frustrating system debugging.
- Document as you go.
  Document your system procedures as well as your system architecture as they are developed. This

will not only become very helpful with future system modifications, but is required for quality system certification such as ISO 9000.

- Start simple.
  It is OK to make assumptions. Once you have a working schedule, refine it to account for the greater detail. Don't get so bogged down in the detail that it inhibits the progress.

5 FUTURE PLANS

As the system moves into regular use, a final detailed verification will be carried out by closely comparing initial schedules against actual performance (Law and McComas 1989). Part of this verification will include the comparison of estimated setup times and calculated run times used in the process plans with the actual times achieved on the shop floor.

It is also planned to have a Pritsker representative return to John Crane to look over our system to ensure all assumptions are valid and to identify potential areas for improved accuracy or simplicity.

To further enhance the system, a data collection system will be installed to automatically update part status. This will eliminate the need for the manufacturing scheduler to walk around and take inventory of part and resource status and key in the information. This will reduce the time to obtain a new schedule.

6 CONCLUSION

Integrated scheduling systems can be used to aid in decision making and make better use of limited company resources. In selecting a commercial package to become a component within an integrated system be sure to consider wider business integration issues so that only those packages which fit with your existing and future systems are considered.

Once an informed and careful choice of base software is made, ensuring that the needs of the people who will interact with the system are also satisfied, you should follow a similarly cautious and well-planned implementation plan. Start small, making assumptions to simplify your model and working with a small area of the entire shop floor. This will allow you to develop substantial in-house expertise which will support development of the system (removal of assumptions and broadening of scope) once problems are ironed out.

ACKNOWLEDGEMENTS

Paul Rogers wishes to thank the Natural Sciences and Engineering Research Council of Canada for its generous research support under grant OGP-012-1522.

REFERENCES

Bauer, M. 1991. Keeping commitments with CIM. *Manufacturing Systems* 9:18-23.

Law, A.M. and M.G. McComas. 1989. Pitfalls to avoid in the simulation of manufacturing systems. *Industrial Engineering* 21:28.

Lilegdon, W.R. 1992. Manufacturing decision making with FACTOR. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J.J. Swain, D. Goldsman, R.C. Crain, and J.R. Wilson, 361-365.

Rogers, P. and M.T. Flanagan. 1991. On-line simulation for the real-time scheduling of manufacturing systems. *Industrial Engineering* 23:37-40.

Starks, D. 1991. Optimization vs simulation. *Monthly Operations Review* 2:11. Pritsker Corporation.

AUTHOR BIOGRAPHIES

MAUREEN T. ROSENWINKEL is an Manufacturing Engineer with John Crane, Inc. in Morton Grove, Illinois. She is responsible for coordinating a range of projects within the company which are part of its movement towards full manufacturing systems integration. Project areas include development of an integrated scheduling system, implementation of an ISO 9001 certified quality system, and the expansion of communication networks on the shop floor. She is a member of IIE and ASQC and is a Certified Quality Engineer. She holds a B.S.I.E. degree from Bradley University and an M.S.I.E. degree from Purdue University.

PAUL ROGERS is an Associate Professor in the Department of Mechanical Engineering (Division of Manufacturing Engineering) at the University of Calgary. His research interests include discrete-event simulation, production planning and control systems, object-oriented modelling for intelligent manufacturing, and models for analysis of manufacturing systems. He is a member of IIE, SCS, and TIMS and serves on the Editorial Board of the International Journal of Computer Integrated Manufacturing. He holds Ph.D. and M.Eng. degrees from Cambridge University in England.