AD-A093 224 FLORIDA UNIV GAINESVILLE DEPT OF INDUSTRIAL AND SYS--ETC F/6 1/3
INTERACTIVE SCHEDULING OF A GENERALIZED FLOWSHOP.(U)
OCT 80 G W MCDONALD, T J HODGSON N00014-76-C-0096
UNCLASSIFIED RR-80-16 NL

1 OF 1
AD A
093224

END
DATE
FILMED
1-81
DTIC

INTERACTIVE SCHEDULING OF A GENERALIZED FLOWSHOP

Research Report No. 80-16

by

Gerald W. McDonald*
Thom J. Hodgson**

October, 1980

*BDM Corporation
Albequerque, N.M. 87105

**Department of Industrial and Systems Engineering
University of Florida
Gainesville, Florida 32611

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

This research was supported in part by the Office of Naval Research, under contract number N00014-76-C-0096, and by the Center for Econometrics and Decision Sciences, University of Florida.

THE FINDINGS OF THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE NAVY POSITION, UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

14

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
RR-80-16

2. GOVT ACCESSION NO.
AD-A093 224

3. RECIPIENT'S CATALOG NUMBER
9 Research rept.

4. TITLE (and Subtitle)
6 Interactive Scheduling of a Generalized Flowshop.

5. TYPE OF REPORT & PERIOD COVERED
Technical

6. PERFORMING ORG. REPORT NUMBER
80-16

7. AUTHOR(s)
10 Gerald W. McDonald
Thom J. Hodgson

8. CONTRACT OR GRANT NUMBER(s)
15 N00014-76-C-0096

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Industrial and Systems Engineering
University of Florida
Gainesville, Florida 32611

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
Arlington, VA 22217

12. REPORT DATE
11 October 1980

13. NUMBER OF PAGES
41

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
12 41

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

N/A

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

MIS, interactive, scheduling, aircraft maintenance, flowshop

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The history of the application of interactive computer techniques to the problem of production scheduling is a compendium of short lived success and failures. This article is a case history which describes the development of an interactive system being used to schedule the starting dates for the overhaul of military aircraft at a Naval Aircraft Rework Facility. The three parts of this serialized article each describe different aspects of the development of

404377

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

this system. Part I concerns itself with the historical causes of failure in this type of development effort, and the general steps taken to overcome these causes. In Part II, the development of a successful scheduling system for a Naval Air Rework Facility is discussed. Part III emphasizes the problems of quantifying the users' desired objective of the creation of "better" schedules.

TABLE OF CONTENTS

PAGE

PART I: Success Through Evolutionary Development

PART II: Development of Computer Programs and Files

PART III: Quantifying User Objectives to Create Better Schedules

Part I: Success Through Evolutionary Development

ABSTRACT

The history of the application of interactive computer techniques to the problem of production scheduling is a compendium of short lived success and failures. This article is a case history which describes the development of an interactive system being used to schedule the starting dates for the overhaul of military aircraft at a Naval Aircraft Rework Facility. The three parts of this serialized article will each describe different aspects of the development of this sytem. Part I concerns itself with the historical causes of failure in this type of development effort, and the general steps taken to overcome these causes. Subsequent parts will describe the computer software developed and the measurements of system effectiveness required to gain complete acceptance of the system by management.

INTRODUCTION

Development of a computer system for the interactive scheduling of starting dates for aircraft being inducted into overhaul at a naval depot level maintenance facility was commenced during May, 1978. Throughout the first few months of this effort a number of development areas were identified as being fraught with danger with regard to the ultimate success of the project. Because of these problem areas, it was decided that an *evolutionary approach* to system design and development would increase the probability of success. This approach was to rely heavily on interaction between the potential users of the system, at both user and management levels, and the developers of the system. Following the development of a framework for system design and implementation the actual work on the system itself was begun.

In a September, 1978 article [1], Victor Godin surveyed the history and the state of the art in interactive production scheduling. In general, his conclusions were that nearly, if not, all of the applications that had been developed up to that point in time had either: (1) failed prior to their being implemented, or (2) had been abandoned by the users shortly after becoming operational. The following is a condensed version of seven of the reasons Godin hypothesized for these failures:

(a) Excessive assumptions,
(b) Lack of system flexibility and sophistication,
(c) Lack of user personnel familarity with computer-based systems,
(d) Expense of graphic hardware and software,
(e) Unrecognized implications of bad schedules,
(f) Overriding of scheduling decisions by political pressures, and
(g) Commercial unattractiveness of the systems due to:
    (1) Custom design,
    (2) High user training costs, and
    (e) Difficult evaluation of cost savings.

The Godin article lent structure to the authors' concerns, and added impetus to the efforts to overcome the causes of failure before they could arise during system development.

PRODUCTION SYSTEM MODEL

The overhaul of aircraft involves a production system that can best be described as a *completely generalized flowshop.* The model schematic for such a system is shown in Figure 1. The aircraft overhaul tasks belong to the flowshop family because all of the different types of aircraft being overhauled proceed through the same set of operational phases in the same order, albeit some of the types spend zero time and require zero resources in certain of the phases. Individual aircraft are disassembled into major components and these components are worked on simultaneously in separate phases. This aspect requires that parallel and overlapping operational phases be introduced into the model. Figure 2 shows a typical flow sequence and the phase durations for an aircraft undergoing overhaul.

Phase durations and resource requirements within a given phase are considered to be deterministic. Should a given aircraft require a non-standard overhaul, either from a phase duration or a resource requirement viewpoint, a new set of deterministic standards is assigned to that aircraft and its intra and inter-phase schedules are adjusted accordingly. The requirement for a nonstandard overhaul may become known prior to the induction of that aircraft or not be discovered until it is undergoing one of the two Estimation and Evaluation (E & E) phases.

OBJECTIVE CRITERIA FOR SCHEDULE DEVELOPMENT

As in most production environments, the workload, in terms of quantity and types of aircraft to be overhauled, is imposed on the plant by an external

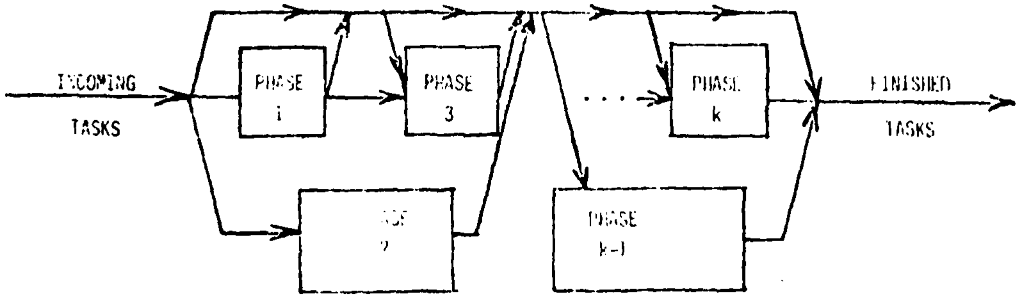
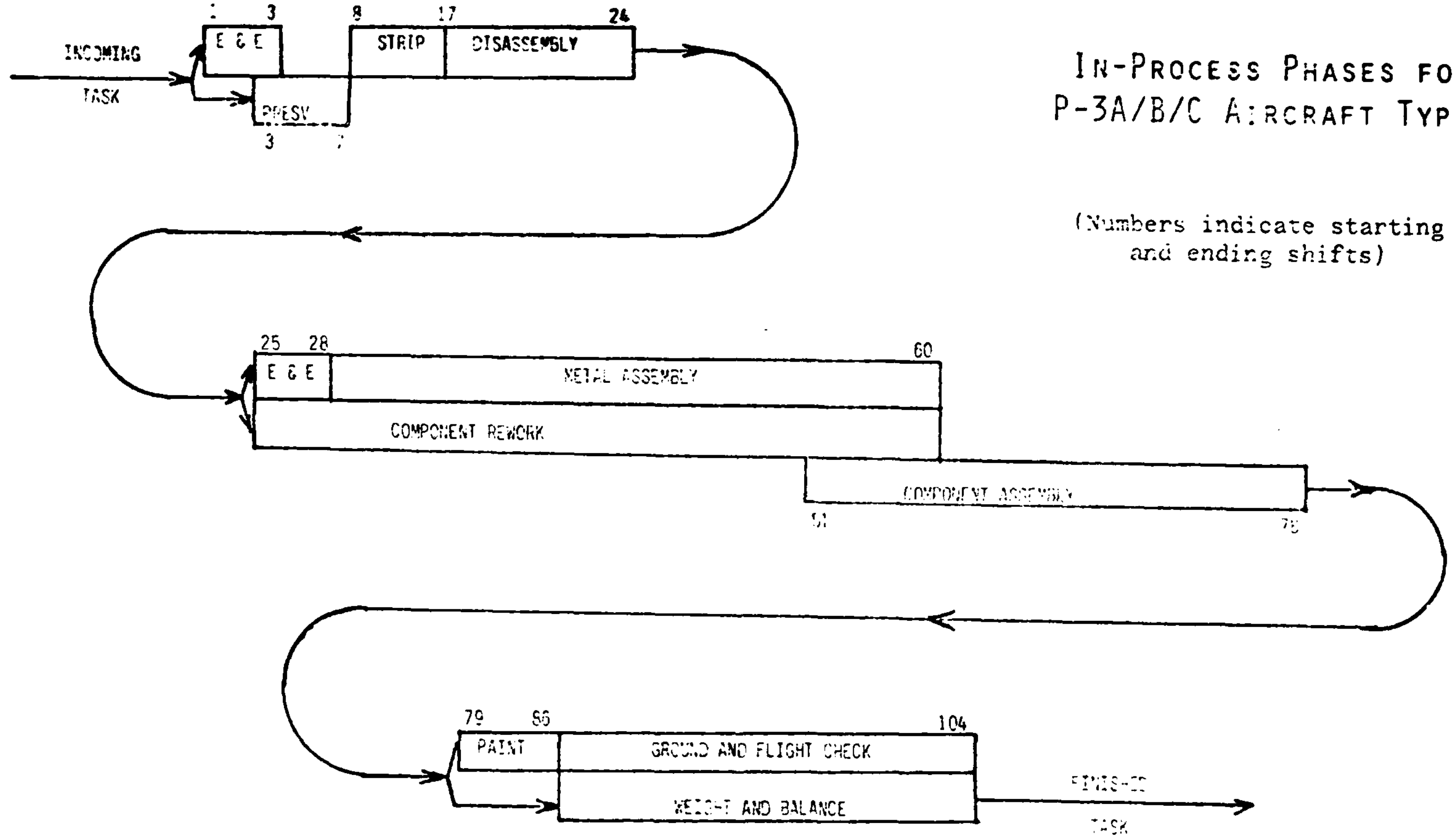


Figure 1 Schematic of a Completely Generalized Flowshop Model



In-Process Phases for P-3A/B/C Aircraft Types

(Numbers indicate starting and ending shifts)

Figure 2   Typical In-Process Phase Sequencing and Durations

source. Any schedule developed by a computer must accomodate the imposed workload in order to be feasible. In addition, certain fixed constraints on the production schedule are imposed by the limits of the facility itself. For example, the paint shop has a limited amount of space and each aircraft requires a certain amount of time in that phase, hence the schedule must accomodate these fixed requirements within the limited space-time continuum available. The desire therefore is to select the best schedule from the set of feasible schedules that satisfy these workload and constraint requirements.

Because of the evolutionary development character of this project, it was decided at the outset to have management determine the objective criteria by which both schedule creation methods and the schedules created by those methods would be judged. To eliminate unduly predjucing their decision, the existence of commonly applied criteria, such as minimum makespan, minimum maximum lateness, etc., were not described to the potential users. The criterion envisioned by management was that of *reducing the day-to-day swings in the manhour requirements for critical trade skills.* On one hand, readers familiar with problems of production scheduling will quickly recognize the universality of such a desirable objective, and on the other hand will understand the extremely difficult problems such a criterion presents. Technically such an objective is difficulty to satisfy because the resulting scheduling problem falls into the category of being NP-hard. The size of the problem is such that branch and bound techniques and other enumeration schemes are not practical, assuming that the specified criterion could be adequately quantified in the first place.

In response to this statement of objective criterion, the users were asked, "Given two different schedules for the same period and the same set of aircraft inductions, how would you determine which one was better?" The reply, "If both were feasible and acceptable [whatever acceptable means] then the better schedule would be the one which has less peaks and valleys for the daily manhour requirements for the critical skills." Their describing the criterion by rephrasing made it obvious that the users envisioned it as one that could be evaluated subjectively. At the same time, the authors recognized that eventually it had to be converted to one that could be objectively evaluated by the computer. The final resolution of this problem will be described in the third article of this series.

EVOLUTIONARY DEVELOPMENT OF AN INTERACTIVE SYSTEM

The subject of objective criterion was introduced above as a lead in to the concept of evolutionary development of an interactive scheduling system. For example, in spite of the fact that no clear objective criterion could be agreed upon at the beginning of development, work could begin on design and programming of the system with a view toward creation of the objective criteria in an evolutionary fashion as the capabilities of the ultimate system began to unfold. It also should be noted that this method of criterion development relates to overcoming of the first of the failure hypotheses listed above, namely the problem of excessive assumptions. In this instance the authors did not assume the form of the objective criterion, but, instead, allowed the users to state their own desires. Having done so, the urge to apply a criterion that would be easy to solve but hard to sell had been overcome, and the standard for evaluating future assumptions had been created --- *Let The User Decide Which Assumptions May Be Made* ---.

The next step was to ask the users, "How do you predict the daily requirements for the trade skills?". Their response was complicated but it boiled down to calculation of the total number of hours required over a three month period, to apportion those hours to individual trade skill by use of some fixed factor values, and then to divide these apportionments by the number of work days in the three month period. The answer did nothing in the way of providing enlightenment as to how the day-to-day swings in manhour requirements for the "critical trade skills" were being evaluated. Readers may recognize this problem as a good example of the fifth hypothesis for failure: Unrecognized implications of bad schdules. In this instance the bad schedules could be recognized only through the actual transfer of skilled personnel from production shop to production shop in reaction to over and under loads in the daily work requirements.

Evolution in this instance led to the creation of a Management Information System (MIS) as the first step in system development. The requisits for such a system included two capabilities: (1) the ability to predict daily manhour requirements for each trade skill that would accrue from a given induction schedule, and (2) the capability to answer "What if ...?" type of questions through interactive varying of the induction schedule.

The data and other information provided during the initial analysis of the production system in preparation for the development of the MIS contained a set of assumptions that were unknown to either the users or the authors at that time. These assumptions had been applied for so long that they were considered to be factual data. In general, these assumptions were that: (1) there were nineteen different trade skills employed within the production facility, and (2) only one type of trade skill was assigned to each of the different production shops. The first version of the MIS was created, debugged, and became operational before it came to light that, in fact, there were forty-nine different trade skills employed, and some of the shops employed as many as eight different trade skills. The nineteen skill, one skill per shop assumptions had apparently been made at some earlier point in time in order to make the problem of hand-calculation of trade skill requirements tractable. Over time, and through the replacement of personnel in the scheduling roles, the assumptions had become acceptable as facts. It should also be noted that the third hypothesis also was in effect: lack of user familiarity with computed based systems (and their capabilities).

Once management personnel began to recognize the capabilities of the computer through utilization of the initial version of the MIS, the decision to eliminate the trade skill, production shop assumptions was made. This meant 'going back to square one' to begin again with the development of an MIS for the prediction of daily trade skill requirements. Such a retrenching is in fact a reversal of the problem of excessive assumptions. In this instance, assumptions were eliminated through system development. In addition, a non-evolutionary approach to system development would have led to the creation of the schedule development segment of the system before the erroneous assumptions were discovered. Had a complete computer system been developed in such an instance it is highly unlikely that the required effort would have been expended to provide it with the necessary, more realistic capabilities; and another failure in system development could have been added to the list.

Adversity was turned into advantage in this case. During the creation of the initial version of the MIS and while it was operational, numerous attempts were made to design and write an interactive program to allow the user to incorporate data changes into the MIS to reflect changes in the production system. Such attempts were frustrated because the effort involved to make the program sufficiently flexible to allow for the full range of data changes that were encountered or envisioned as possible in the future was beyond the available programming resources. At 'square one' a complete redesign of the basic (raw) data files labeled AIRCRAFT and STANDARDS in Figure 3 was undertaken. The new structure of these files allowed the user to incorporate changes into the system at this point through the use of the file editing capabilities built into the computer's executive software. After such changes were input the user could simply recreate the structured data files utilized by the manhour prediction programs. Evolution had provided the knowledge and the means for overcoming the hypothesis on failure due to inflexibility.

Before the second version of the MIS was completed the users had gotten into the swing of creating ideas for additional capabilities to be incorporated. An example of such a feature is that of predicting the daily trade skill requirements for subsets of production shops, and in particular for shops grouped into divisions and branches.

The concept of an evolutionary approach to system development began to bear fruit, and the users were becoming personally involved in the matter of ultimate success of the system. Such involvement has at least two impetuses: (1) recognition that the system will improve the scheduling product and ease the scheduler's problems, and, more importantly, (2) the user has at least partial responsibility for the ultimate success or failure of the system.

Numerous components of the final interactive scheduling system resulted from ideas conceived either solely by the users or by the users working in consonance with the authors. The end result has been a system which has far more practical utility than would have any system which could have been created based entirely upon a set of specifications created before the beginning of system design and creation. Part II of this series deals with further exposition of the idea of evolutionary, and the actual development of computer programs and files necessary to perform the scheduling operations.

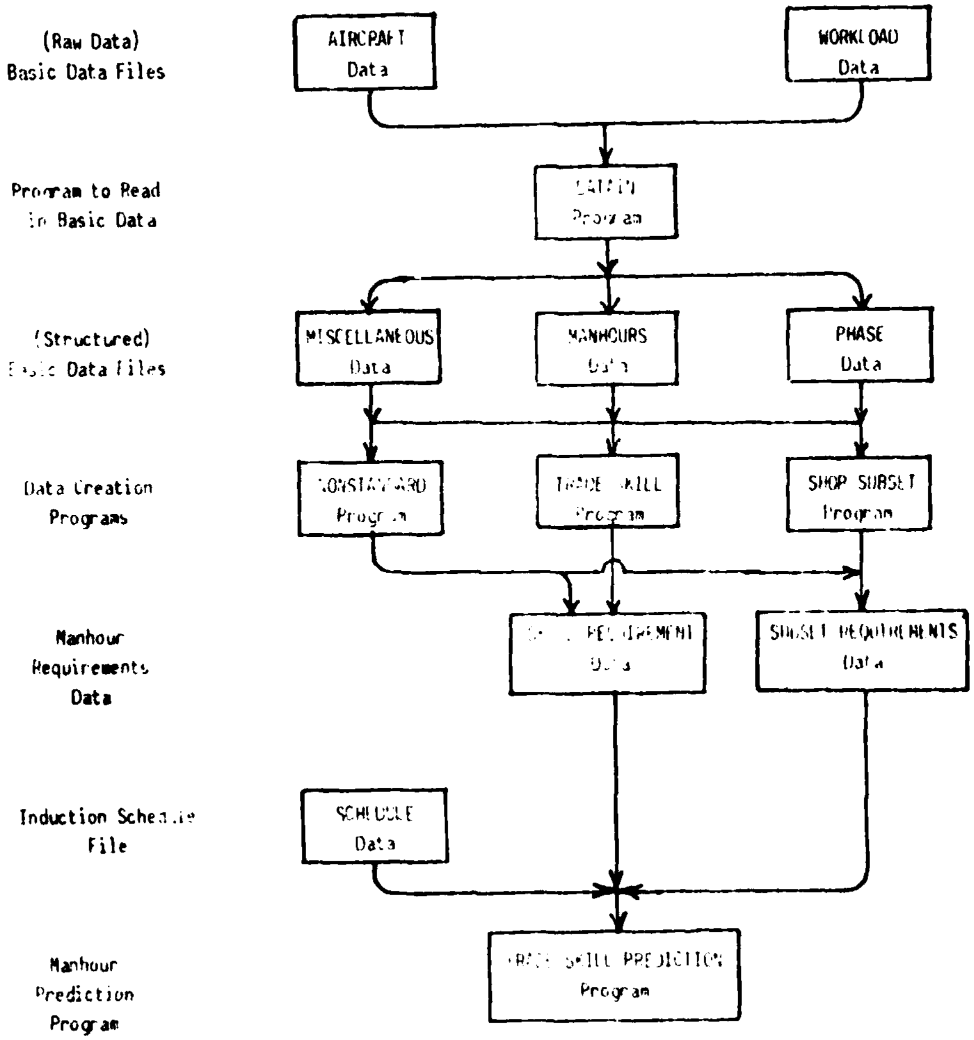


Figure 3 A Protion of the Revised
MANAGEMENT INFORMATION SYSTEM

REFERENCES

1. Godin, Victor, B., "Interactive Scheduling: Historical Survey and State of the Art," AIIE TRANSACTIONS, 10, 3, 331-337 (1978).

2. McDonald, Gerald W., "Evolutionary Development of an Interactive Scheduling System for a Completely Generalized Flowshop," Doctoral Dissertation, University of Florida, June, 1980.

3. McDonald, Gerald W. and Hodgson, Thom J., "User's Guide to the NARF-JAX Computer Aided Scheduling System (CASS)," Department of Industrial and Systems Engineering Technical Report 80-18, University of Florida, Gainesville, FL 32611, October, 1980.

Part II: Development of Computer Programs and Files

ABSTRACT

Part I of this series of articles introduced the concept of evolutionary development in the creation of an interactive scheduling system as a means of overcoming the problems that have beset others and caused the failure of many such attempts at the application of computers to production scheduling.
In this part, the development of a successful scheduling system for a Naval Aircraft Rework Facility is discussed in more specific terms. The emphasis continues to be on the evolutionary aspects of development which have led to its successful conclusion, however, a major segment of this article also discusses the problem of bringing the objectives of management for computer developed schedules into line with the actual capabilities of a computer system.

BACKGROUND

As a brief review of Part I, we recall the article by Godin wherein a set of hypotheses is set forth for the failure of almost all previous attempts at interactive scheduling of production shops [3]. In condensed form these hypotheses are:

(a) Excessive assumptions,
(b) Lack of system flexibility and sophistication,
(c) Lack of user personnel familiarity with computer-based systems,
(d) High expense of graphic hardware and software,
(e) Unrecognized implications of bad schedules,
(f) Political pressures overriding scheduling decisions, and
(g) Commercial unattractiveness of systems due to:
  (1) Custom design,
  (2) High user training costs, and
  (3) Difficulty in evaluating cost savings.

Discussion of the concept of evolutionary development of scheduling systems in Part I briefly touched on overcoming the problems of (a), (b), (c), and (e) above. The overcoming of these same problems, and those associated with the other three hypotheses, is continued below.

EVOLUTIONARY DEVELOPMENT DESCRIBED

In the development of any computer-based system, the first step is normally one of ascertaining the features and capabilities that are desired for the final product by the user. This step is commonly accomplished through a series of conferences during which ideas are exchanged between user and developer. Under a non-evolutionary method of system development the end result of such a series of conferences is a set of specifications for the final system. In that instance, the developer then proceeds to create the final product on his own. He then returns to the user with the product, effects its implementation on the user's machine, collects his fee, and leaves.

The main problem with such a development method is one of communications. In many instances the users have no real grasp of what a properly designed, complete system might be capable of doing, and the developer has no real understanding of the user's work routines which lead to problems requiring resolution. Often, the barriers of background and job-related terms in the conversations on each side will inhibit the development of a really comprehensive and meaningful set of system specifications which could be used as a basis for the design and implementation of a fully capable interactive system.

This is where the evolutionary development method comes into play. At the end of the initial conferences the developer's next task is one of creating a segment of the interactive scheduling system which will begin to fulfill the user's requirements. This segment is not intended to be a component of the final product. Instead, it is intended to be a prototype whose main role is to stimulate the interchange of ideas between user and developer in order to enhance future and final versions of the component itself and the other components making up the entire system. In addition, these exchanges provide the ideas used as the basis for the creation of additional segment prototypes which are used as building blocks to expand the capabilities and usefulness during system growth to its 'final' form.

In a properly functioning evolutionary atmosphere, the latest version of every segment prototype should be used as an avenue for a rapid, two-way feedback between user and developer for exploring the possible expansions of system capabilities. Only in this fashion can the final system be flexible and sophisticated enough to meet the needs and the demands of the user in the performance of his everyday roles. It is important to note that the (re)evaluation of any one of the available segments may lead to the creation of a need for changes in other segments, or to the need for an entirely new segment with new capabilities.

At this point, readers who are experienced as system developers are likely to think two disparate thoughts. First, "The concept looks nice", and second "It will never work in the real world." In many cases the first is based upon some problems which they have experienced in system developments in the past, and they now recognize that the evolutionary concept would have simplified their solution. The second is likely based upon the concern for a set of system specifications for use as a contractual basis. This problem in application is indeed difficult, but not insurmountable, and the improved mode for system development, with its attendant increase in the probability for success, has proven to be well worth the effort [5].

SYNOPSIS OF SYSTEM DEVELOPMENT

The first few steps in the actual development of a scheduling system for a Naval Aircraft Rework Facility were discussed briefly in Part I of this series. They involved the creation of a rudimentary Management Information System (MIS) whose primary role was one of allowing the scheduling personnel to predict the daily manhour requirements for each different trade skill, said requirements resulting from a given induction schedule. Utilization and evaluation of the initial MIS prototype lead to the discovery that assumptions as to the number of distribution of trade skills that had been accepted prior to automation had been considerably understated within the scheduling office for a prolonged period of time. This fact in turn lead to the creation of a second version of the MIS based upon the corrected trade skill factors. The new version included additional capabilities such as the prediction of trade skill requirements for separate production shops branches, division, and departments within the facility's organizational hierarchy.

The next major segment of the MIS involved a component which has the ability to predict manhour requirements for the individual production shops themselves, without distinguishing the trade skills assigned to those shops. This segment consists primarily of two additional computer programs (SHOP HOURS and SHOP PREDICTION) and one additional data file created by the SHOP HOURS program (SHOP REQUIREMENTS). These three elements are depicted near the left hand side of Figure 1, which in turn is an expansion of the MIS structure presented in Figure 3 of Part I. The requirements for each shop can be predicted by day, month, quarter, or any selected period from 1 to 66 work days long.

It is important to note that the need for a segment to predict the production shop manhours was never considered nor discussed during the initial system development conferences. In addition, the evidence on system utilization to date has shown that the production shop predictions are far more heavily utilized than are the trade skill predictions whose requirement provided the original impetus to begin development of the system.

After the shop prediction segment was up and running in prototype form, the developers of the system wanted to go ahead with the portion of the final system that would be used in the creation of induction schedules for future time periods. However, the users had other ideas. They proposed a new segment for the MIS, and insisted that a prototype for it be developed before beginning on the scheduling portion. A primary feature of this segment to be added to the MIS is that it provides the users with a capability to specify a future time frame of one quarter or one year duration, then to specify the number of manhours required for each production shop during the selected period (rather than to base the number of hours on a given induction schedule), and from the data specified to predict the number of manhours required for each

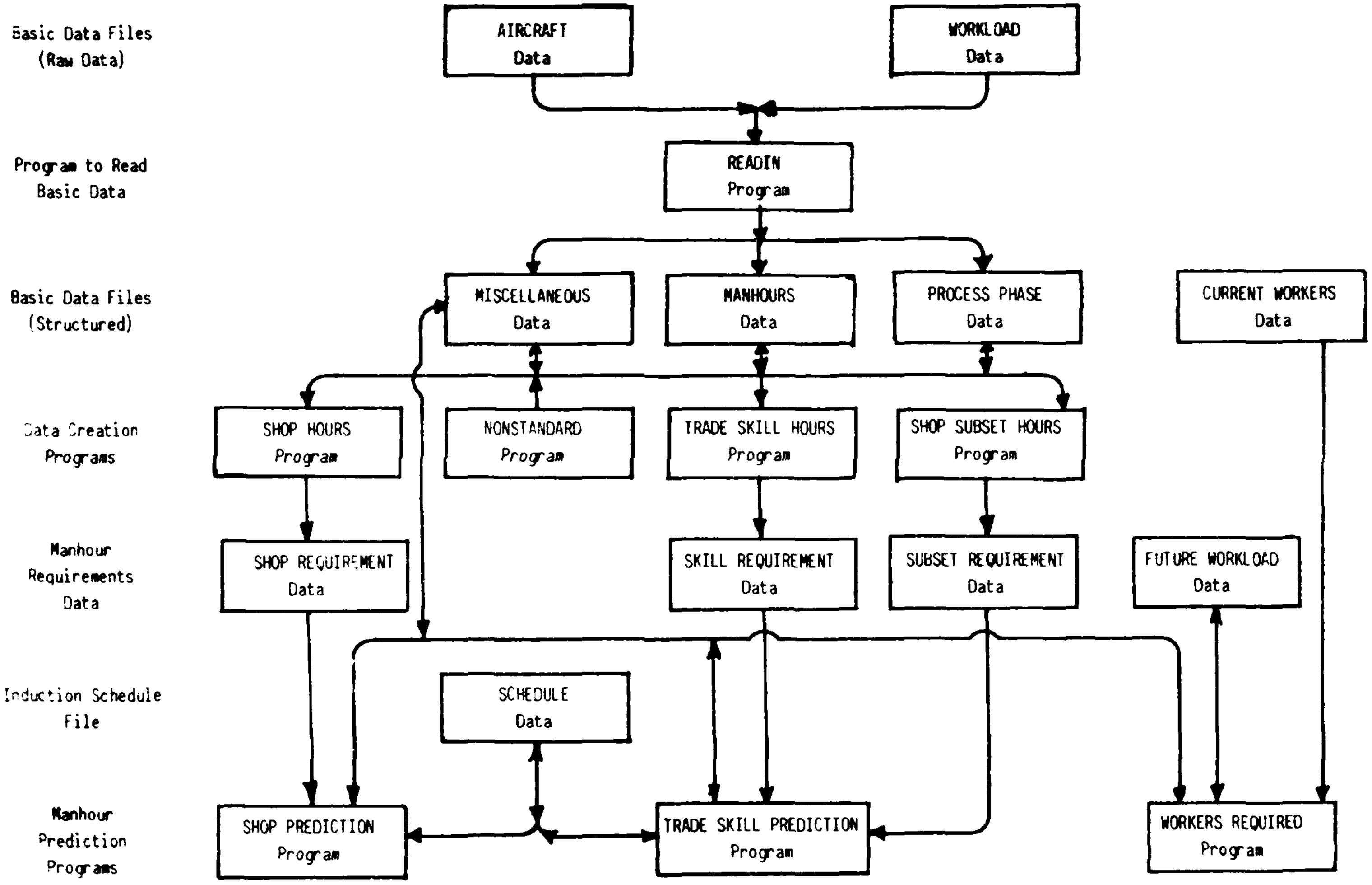


Figure 1 Basic Structure of the MANAGEMENT INFORMATION SYSTEM

trade skill within each shop over the selected period. to predict the average number of workers required for each trade within each shop during the period, to predict the average number of workers required for each trade skill across each branch and division as well as for the entire facility, and finally to attrit the current number of workers within each trade skill on the bases of historical attrition rates in order to predict the number of workers in each trade that are expected to be available during the time frame under consideration. Any shortfall in a given trade skill between predicted requirements and predicted availability represents the number of workers that will have to be hired for or cross-trained to that trade skill from ano ıer skill showing a predicted excess.

This latest MIS segment, whose concept was conceived entirely by the user personnel, is another example of the greatly increased capabilities for the final system which came about through evolutionary development. It represents, in some measure, an overcoming of the second and third hypotheses for failure; those dealing with lack of flexibility and sophistication and the lack of user familiarity with computer-based systems.

This segment is depicted near the right hand side of Figure 1 as the elements labeled CURRENT WORKERS Data, FUTURE WORKLOAD Data, and WORKERS REQUIRED Program.

SCHEDULE CREATION CAPABILITIES

One of the major problems to be solved prior to beginning the design and programming on the schedule creation portion of the system was hinted at in Part I of this series; that being the actual definition of the objective function for comparison measurement of different schedules drawn from the set of feasible schedules. The criterion envisioned by facility management was one of 'reducing the day-to-day swings in manhour requirements for the "critical" trade skills in order to reduce overtime costs'. The problems presented to a system developer by such a criterion reduce to two which are extremely difficult to solve: (1) how does one measure the "levelness" of a schedule to compare it against another schedule given a set of daily manhour predictions for each trade skill that accrue to each of the schedules, and (2) given that a method is decided upon for measuring this "levelness," what technique should be selected for the creation of the schedules to be compared. In addition, inherent in the measure of "levelness" is the determination of the relative criticality of trade skills and the number of skills which are to be considered as critical during the creation and comparison of schedules.

Solution of the question for measuring the "levelness" of a schedule does not necessarily have a single answer. For example, a measure which can be evaluated objectively, such a linear combination of standard deviations for each critical skill, may not be useful in convincing managers of the computer system's capability to create improved schedules; and it is the managers who will ultimately decide upon the success or failure of the system.

Another factor in measuring the levelness of schedules is the question of combining or weighting the manhour requirements for the critical trade skills in the development of the measure. For example, suppose that for a given schedule the standard deviation for trade skill A is five hours and for trade skill B is twenty-five hours. Can one say that the requirements for A are more level than for B. If one knows that they both have approximately the same mean or average number of hours, then the answer is yes. However, suppose the average daily requirement for A is ten hours and the average daily requirement for B is five hundred hours, then it would appear that the schedule is more level for skill B. Having considered this analogy, it becomes apparent that the standard deviation for trade skills by themselves do not necessarily provide a valid measure for the "levelness" ot a give schedule. More information on this subject will follow in Part III of this series. An even more extensive discussion of the problem, and its solution, is contained in [5].

As to the problem of schedule creation methods, the available literature in the subject provides little useful information as to its solution, particularly in the case of attempting to level the resource requirements per unit of time. The majority of flowshop research is based upon the definition of a flowshop which considers that only one task can be in a given phase at any one time; i.e. there is no passing of jobs during processing and the order of finish for jobs is the same as the order of start [1], [2], and [4]. In addition, flowshop research has concentrated on academic objectives, such as minimizing makespan or minimizing maximum lateness, rather than requirements such as levelling the day-to-day requirements for resources, or some other objective that would be more useful to industry [4]. For example, in a July, 1977 article, Dannenbring published "An Evaluation of Flowshop Sequencing Heruistics," wherein he discusses the concepts underlying eleven different flowshop scheduling techniques [2]. All eleven techniques were limited to the minimizing maximum makespan objective, and some techniques were effective only in the solution of problems involving only three or four tasks being scheduled on three or four machines. None of these techniques attacked the problem of a generalized flowshop, where tasks could pass during processing, nor the problem of having a continuum of input tasks over time.

An article by Gupta [4] divides the theoretical developments in flowshop scheduling, under the no-passing and minimum makespan assumptions, into three categories: (1) Combinatorial analysis, (2) Branch and Bound procedures, and

(3) Lexicographic search. None of these will provide a satisfactory approach to the solution of the scheduling problem at hand because the combinatorics associated with a large-scale problem such as this one are well beyond the capabilities of any of these approaches.

It appears, therefore, that anyone attempting to solve real-world flowshop scheduling problems must therefore turn to techniques which are heuristic in nature. One family of techniques which appears to show promise is discussed in an article by Page [7]. It is related to computer sorting methods involving individual and group exchanges of elements within a list, and a schedule can easily be considered a list. Part III of this series will discuss how one such heuristic was applied in the Naval Aircraft Rework Scheduling system.

REFERENCES

1. Baker, Kenneth R., INTRODUCTION TO SEQUENCING AND SCHEDULING, John Wiley and Sons, New York, NY, 1974.

2. Dannenbring, D. G., "An Evaluation of Flowshop Sequencing Heuristics," MANAGEMENT SCIENCE, 23, 11, 1174-1182 (1977).

3. Godin, Victor B., "Interactive Scheduling: Historical Survey and State of the Art," AIIE TRANSACTIONS, 10, 3, 331-337 (1978).

4. Gupta, J. N. D., "M-Stage Scheduling Problem, A Critical Appraisal," INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH, 9, 2, 267-281 (1971).

5. McDonald, Gerald W., "Evolutionary Development of an Interactive Scheduling System for a Generalized Flowshop," Doctoral Dissertation, University of Florida, June, 1980.

6. McDonald, Gerald W., and Hodgson, Thom J., "User's Guide to the NARF-JAX Computer Aided Scheduling System (CASS)," Department of Industrial and Systems Engineering Technical Report 80-18, University of Florida, Gainesville FL 32611, October, 1980.

7. Page, E. S., "On the Schedule of Jobs by Computers," THE COMPUTER JOURNAL, 5, 214-221, (1962).

Part III: Quantifying User Objectives to Create Better Schedules

ABSTRACT

Parts I and II of this series introduced the concept of evolutionary development to increase the probability of success for the final version of an interactive production scheduling system. The series is a case history which describes the development of an interactive computer system which is being used to schedule the starting dates for aircraft being inducted into overhaul at a Naval Aircraft Rework Facility. This article in the series will emphasize the problems of quantifying the users' desired objective for the creation of "better" schedules, and that of choosing a viable method for the creation of schedules when the objective involves the reduction of day-to-day swings in resource requirements.

CRITICAL RESOURCES

It may be recalled from the earlier parts that the users' stated objective in this instance was that of "reducing the day-to-day swings in the manhour requirements for the 'critical' trade skills." The quantification of such an objective function is difficult and involves more than one problem. The first problem that comes to mind is that of deciding which trade skills are critical, and an order for the criticality of those skills included in the critical subset. In searching for a method to solve this problem the first step is to determine if there is sufficient data for the choice of critical skills to allow them to be evaluated and specified quantitatively. The answer for this application to a Naval Aircraft Rework Facility was negative. An answer to this problem, therefore, had to involve the subjective decision making capabilities of the user personnel. In substance, the problem is one of selecting and ordering a set of critical skills from a list of fifty trades. The human mind would find it impossible to simply scan a long list of skills and select an order subset of those to be called the critical trades. Instead, one must present the selection problem as one of selecting from subsets of the original list, but when there is a long list to begin with the combinatorics make this approach very difficult due to the required number of repetative reviews required. The solution taken was one of presenting subsets of length one and having the user respond with his subjective estimate of criticality for that trade with an integer on a scale from 1 to 9, higher values for more critical skills. The result is a list of all skills semi-ordered on the basis of criticality. The user is then presented the upper portion of this list and given the opportunity to either (1) reorder that portion into the final critical subset list, or to (2) add other trade skills to the subset by returning to the subjective scaling from 1 to 9 for all trades.

The other decision with respect to the critical trade skills is one of deciding how many are to be considered as critical. Nothing would preclude leaving this decision up to the user. However, the greater the list length the more difficult and time consuming the problem creating schedules which will reduce the variation in daily requirements for those skills. Arbitrarily, therefore, the list length in this instance was set at five, approximately ten percent of the total number of trade skills.

One additional step was implemented in this application, that being the creation of a new data file which is a reordered subset of the SKILL REQUIREMENT Data file contained in the MIS portion of interactive scheduling system (refer to parts I and II of this series). The reordering being done on the basis of criticality.

Some insight into the basis for trade skill criticality might be of interest at this point. In this application there were three primary factors involved in determining the criticality of trade skills. They were: (1) impact of random calls upon certain trade skills to perform higher priority functions, such as the need for metalsmiths to perform emergency field repair operations, (2) the skills which involve the largest number of manhours and, therefore, have a larger impact on overtime costs during peak periods, and (3) the skills which are the most mobile to other industries, i.e., workers skilled in high technology areas such as computers and electronics.

The SKILL REQUIREMENT and CRITICAL SKILL REQUIREMENT Data files and the CRITICALITY Program shown in the upper right corner of Figure 1 are the elements in the interactive system which relate to the designation of trade skill criticality.

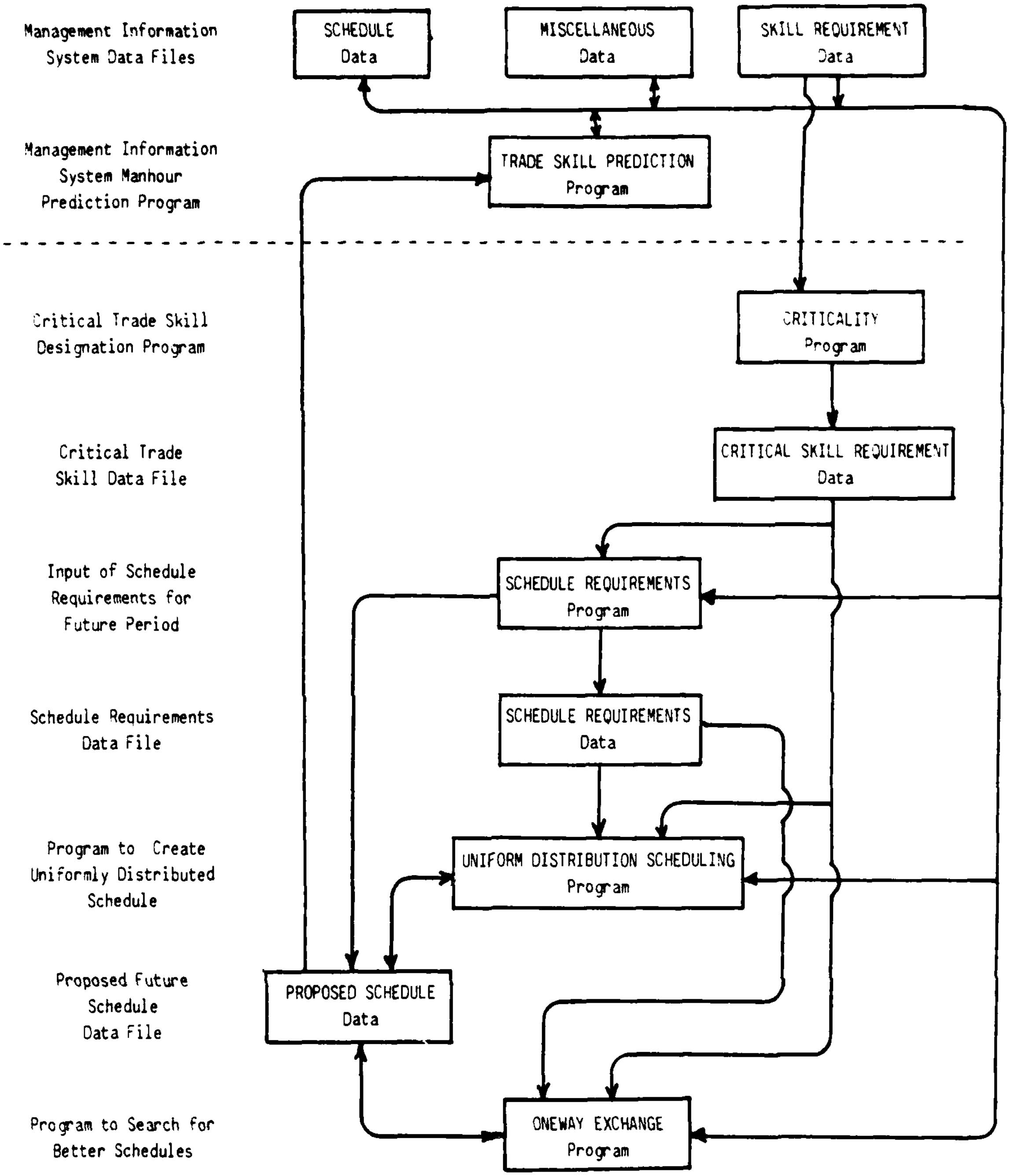

Management Information System Data Files
SCHEDULE Data
MISCELLANEOUS Data
SKILL REQUIREMENT Data
Management Information System Manhour Prediction Program
TRADE SKILL PREDICTION Program
Critical Trade Skill Designation Program
CRITICALITY Program
Critical Trade Skill Data File
CRITICAL SKILL REQUIREMENT Data
Input of Schedule Requirements for Future Period
SCHEDULE REQUIREMENTS Program
Schedule Requirements Data File
SCHEDULE REQUIREMENTS Data
Program to Create Uniformly Distributed Schedule
UNIFORM DISTRIBUTION SCHEDULING Program
Proposed Future Schedule Data File
PROPOSED SCHEDULE Data
Program to Search for Better Schedules
ONEWAY EXCHANGE Program

Figure 1 SCHEDULE CREATION Portion of the Interactive Production Scheduling System

OBJECTIVE FUNCTION

The stated objective of the user for the creation of improved schedules involves some measure of "levelness" for the critical trade skills. Problems associated with such a concept were discussed in Part II of this series. The solution in this application was as follows. Create a proposed induction schedule that satisfies all of the feasibility requirements of products required and system constraints. Using the data in the CRITICAL SKILL REQUIREMENT File, predict the daily manhour requirements for each of the critical trade skills. Using the results, calculate the standard deviaion and the mean for each critical skill. Divide the mean for each by the corresponding standard deviation to calculate the inverses for the coefficients of variation (herein referred to as the mean deviation and abbreviated meandev), and then use a weighted sum of these meandevs as a measure for comparing the proposed schedule against other feasible schedules which are subsequently developed.

The critical trade skill combinations and weight selections tested included both weighted and unweighted sums of:

(a) Maximum sum of minimum daily requirements for each critical skill,
(b) Maximum sum of the ratio for each skill calculated by dividing the minimum daily requirement by the mean daily requirement,
(c) Maximum sum of the minimum day divided by the maximum day for each skill, and
(d) Minimum sum of standard deviations for each skill.

Of the options provided by this list, the unweighted sum of mean deviations produced measurements which were slightly better than the results of the other options. The comparison of these schedule creation methods is discussed below. Suffice to say at this point that the final choice between weighted and unweighted sums of mean deviations was made in favor of a linear weighting of the skills on the basis of lower computer execution times during schedule creation. This choice was also in agreement with the results in comparing schedule creation methods on the basis of predicted overtime requirements discussed below.

Specifically, the objective function chosen to implement the users' desired objective was to choose from between two proposed schedules the one that minimized the sum of weighted mean deviations.

SCHEDULE CREATION METHODS

A review of survey literature available on the methods for scheduling flowshops led to the decision to implement some method of interchanging elements in proposed schedules as a means of searching for better schedules [1], [2], and [3]. As discussed in Part II, the common techniques applied in optimizing flowshop schedules were not practical in this application due to the size of the task.

The concept of improving schedules through a series of interchanging the elements requires that an initial proposed schedule be created. A number of heuristics were tested in this application to find a good method of creating an initial proposal. The best of these proved to be one which creates an initial schedule by uniformly distributing the required number of each product type over the period for which a schedule is being created. The types are added to the schedule in the order of most-to-least total requirement for critical trade skill man-hours and load limit feasibility constraints are incorporated to modify the uniform distribution as necessary to maintain feasibility. The end result of creating a schedule in this manner turned out to be, in reality, an automation of the methods utilized by the NARF, scheduling personnel in their creation of schedules by hand.

Given a proposed schedule as a starting point, the next step is to commence a series of interchanging the elements in the schedule in an attempt to improve the schedule. A one-way interchange for such a schedule is simply the removal of a product from its current starting point in the schedule and to move it to any other starting point in the schedule which retains feasibility of the schedule overall. The resulting schedule is then compared to the original and the better of the two is retained as the (new) proposed schedule. One can then stop at any point after the comparison of two schedules and retain the best one found to that point as the final schedule, or an exhaustive search can be conducted until no more one-way interchanges are left to be tested.

Such a one-way interchange method was programmed and implemented in this application. The possibility of conducting a series of two-way interchanges, where the starting date for two different product types are interchanged, was considered but discarded due to the computer execution times which resulted from conducting an exhaustive one-way interchange search. If a similar scheduling problem is automated on a computer system whose execution time is considerably faster than the PDP-11/70 used in this application, then one should evaluate the possibility of going to two-way or higher exchange methods in the search for better schedules.

At this point let us review the sequence for developing an entire schedule for some future period by referring to Figure 1.

The CRITICALITY program is executed to update the list of critical trade skills to reflect current manpower conditions. The user then executes the SCHEDULE REQUIREMENTS program to input all of the data necessary to begin the creation of a new schedule. This data includes items such as:

(a) Starting and ending dates for the period being scheduled,
(b) Number of each product type to be scheduled during the period, and
(c) The date for any of the products whose induction is already fixed for some reason beyond the control of scheduling personnel.

The SCHEDULE REQUIREMENTS program then takes the current schedule out of the SCHEDULE Data file, zeroes it out for the desired period, adds any products whose dates are fixed and copies the results to the PROPOSED SCHEDULE file. The UNIFORMS DISTRIBUTION SCHEDULING program is executed next to create an initial proposed schedule and that schedule is then placed in the PROPOSED SCHEDULE file. The next step is to run the ONEWAY EXCHANGE program to search for better schedules. The final schedule created by this search and selected as the best of two being considered is then stored in the PROPOSED SCHEDULE file. From that point the final proposed schedule can be analyzed through execution of the TRADE SKILL PREDICTION program contained in the MIS portion of the scheduling system. Final refinements to the proposed schedule may then be made by user personnel at this time, recognizing that such refinements may in fact improve or adversely affect the final results of critical trade skill requirements leveling. The final version of the PROPOSED SCHEDULE is then copied into the SCHEDULE Data file and at that time becomes the current schedule.

The problem of creating a schedule in this fashion sounds simple, and from the system user's point of view it is. From the system design and programming point of view, however, the problem is very difficult. The incorporation of feasibility constraints and the facet whereby certain elements in the proposed schedule have their starting times fixed greatly complicate the problem, but these concepts are mandatory for the system to attain the sophistication and flexibility to ensure any measure of success for the final product.

MEASURING SYSTEM EFFECTIVENESS

The final subject in this series is that of measuring the effectiveness of the system that was created. Two facets of effectiveness were evaluated;

leveling of critical trade skill requirements, and computer execution times for the various objective criteria tested.

The leveling of critical trade skills, in turn, was analyzed in two ways; statistical analysis for academic purposes, utilizing Student's t statistics, and comparative analysis for management's sake. The comparison of schedule creation methods involved evaluating the schedules created for eight one-quarter work periods, two years in all. A complete analysis is contained in reference [4], however, the main comparison of interest for academic purposes is the one between the schedules created by hand by the user personnel, who had been creating such schedules for years, and the schedules created by the computer utilizing the weighted mean deviation objective criteria. Table 1 contains the t-Scores which resulted from comparing the differences between the pairs of schedules for all eight of the quarters.

TABLE 1

t-SCORE COMPARISON OF INDUCTION SCHEDULES
Created by INTERACTIVE SCHEDULING SYSTEM Versus
Those Created by USER PERSONNEL Working Alone

MANPOWER RESOURCES	STANDARD DEVIATION	MAXIMUM DAY Minus MINIMUM DAY	MEAN DEVIATION
CRITICAL SKILLS only (n=39)	4.477	3.861	5.561
ALL TRADE SKILLS (n>39)	3.180	2.782	4.392

Note: For values of n ≥ 39, when t-Scores exceed 1.95 there is less than a 5% probability of making a type I error when stating that "The Interactive Scheduling System produces significantly better schedules when measured by the statistic indicated."

From a statistically sound point of view, the data in Table 1 supports the contention that the Interative Scheduling System does produce better schedules than those created by hand. This evidence, however, did not provide a comprehensively acceptable basis for management to agree with the claim of improved schedules. Therefore, another measure had to be devised for use a proof of the system's capabilities. That measure turned out to be one of comparing the predicted reduction in overtime manhours required for the computer-created versus the hand-created schedules. Table 2 contains the data developed for this purpose. Since both schedules contained identical types and quantities of products to be inducted, the average daily manhour requirements for the entire two year period were, for all practical purposes, identical for each trade skill. Using these averages for each trade skill as an estimate of the manpower available, one can estimate the overtime requirements for a given schedule by summing up the manhours required above the average for each day that the requirement is predicted to be above the mean. Summing up these excess requirements for all trade skills and for the critical trade skills resulted in the differences contained in Table 2.

TABLE 2

COMPARISON OF INDUCTION SCHEDULES
Based Upon Predicted OVERTIME HOURS Required

PREDICTED REQUIREMENTS	HAND-CREATED SCHEDULE	COMPUTER-CREATED SCHEDULE	COMPUTER SAVINGS
(Raw Data)			
ALL SKILLS	116,305	107,259	9,049
CRITICAL SKILLS	62,390	50,548	11,842
(Smoothed Data)			
ALL SKILLS	98,168	89,909	8,259
CRITICAL SKILLS	53,539	43,548	9,955

Management offered some slight objections to the use of "raw predictions" for the comparison of scheduling methods by use of predicted overtime. Their claim was that shop supervisors could smooth the day-to-day requirements to help level out the peaks and valleys in their own shop's workload. Agreement was reached that this movement of work load by supervisors was limited to about 'one day either way' because of the required interfaces between the work efforts of different shops. In view of this agreement, it was decided that the predicted manhour requirements would be smoothed prior to estimating the overtime requirements. The smoothing function employed was simply an application of a three-day running average for each trade skill. After this smoothing, the difference in manhour requirements for the interactive computer scheduling method versus the hand scheduling method were calculated. They in turn showed that the computer schedules would result in an approximate savings of more than eight thousand manhours over the two year period for which schedules were created. The overtime hours assessments were successful in convincing management that the system can, in fact, improve the production scheduling process for the entire facility.

CONCLUSIONS

The thesis for this series of artices, is that the problem of failure in the development of interactive production scheduling systems can be solved through the development of such systems in a more evolutionary manner than has been common in past system developments. The success of the system described in this series is considered to support the thesis contention. In addition, much more work needs to be done in the area of flowshop scheduling with respect to both the creation of more practical objective functions, and improved techniques for the creation of schedules.

REFERENCES

1. Baker, K. R., INTRODUCTION TO SEQUENCING AND SCHEDULING, John Wiley and Sons, New York, NY, 1974.

2. Dannenbring, D. G., "An Evaluation of Flowshop Sequencing Heuristics," MANAGEMENT SCIENCE, 23, 11, 1174-1182 (1977).

3. Gupta, J. N. D., "M-Stage Scheduling Problem, A Critical Appraisal," INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH, 9, 2, 267-281, (1971).

4. McDonald, G. W., "Evolutionary Development of an Interactive Scheduling System for a Generalized Flowshop," Doctoral Dissertation, University of Florida, June, 1980.

5. McDonald, G. W., and Hodgson, T. J., "User's Guide to the NARF-JAX Computer Aided Scheduling System (CASS)," Department of Industrial and Systems Engineering Technical Report 80-18, University of Florida, Gainesville, FL 32611, October, 1980.