INTERFACES
Vol. 11, No. 3, June 1981
Copyright © 1981, The Institute of Management Sciences
0092-2102/81/1103/0083$01.25

INTERACTIVE SCHEDULING OF A GENERALIZED FLOWSHOP. PART II: DEVELOPMENT OF COMPUTER PROGRAMS AND FILES*

Thom J. Hodgson

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

and

Gerald W. McDonald

BDM Corporation, Albuquerque, New Mexico 87105

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 [Godin, 1978]. 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
(g) Commercial unattractiveness of systems due to:
 (1) Custom design
 (2) High user training costs
 (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

PRODUCTION SCHEDULING — FLOWSHOP

*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 Science, University of Florida.

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 nonevolutionary 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.

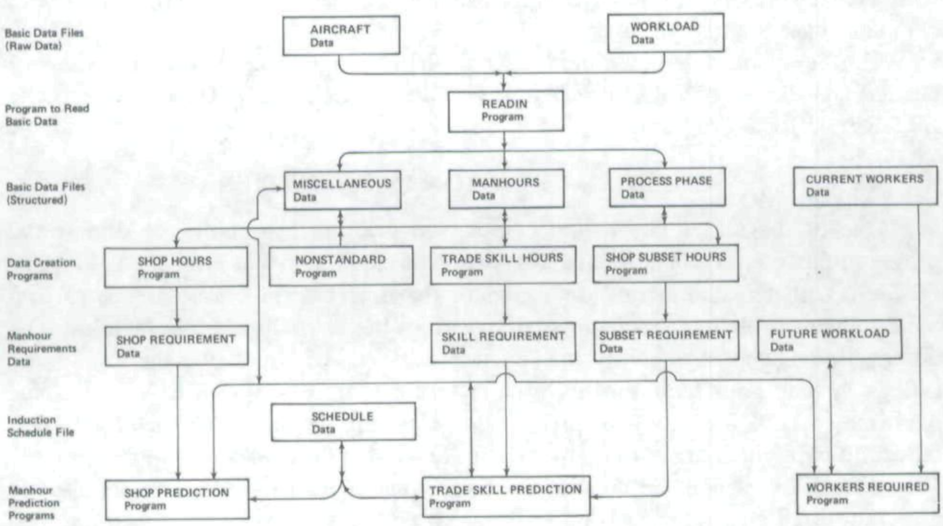
At this point, readers who are experienced as system developers are likely to think two disparate thoughts. First, "The concept looks nice," and second, "It will never work in the real world." In many cases, the first is based upon some problems which they have experienced in system developments in the past, and they now recognize that the evolutionary concept would have simplified their solution. The second is likely to be 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 [McDonald, 1980a].

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 man-hour requirements for each different trade skill, said requirements resulting from a given induction schedule. Utilization and evaluation of the initial MIS prototype led 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, led 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, divisions, and departments within the facility's organizational hierarchy.

The next major segment of the MIS involved a component which has the ability to predict man-hour 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 workdays long.

FIGURE 1. BASIC STRUCTURE OF THE MANAGEMENT INFORMATION SYSTEM.



It is important to note that the need for a segment to predict the production shop man-hours 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 man-hours required for each production shop during the selected period (rather than to base the number of hours on a given induction schedule). From the data specified it is also possible to predict (1) the number of man-hours required for each trade skill within each shop over the selected period, (2) the average number of workers required for each trade within each shop during the period, (3) the average number of workers required for each trade skill across each branch and division as well as for the entire facility, and finally (4) the attrition of current workers within each trade skill on the basis of historical attrition rates in order to predict the number of available workers in each trade 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 another 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 is, 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 man-hour 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 man-hour predictions for each trade skill that accrue to each of the schedules? (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 man-hour 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 5 hours and for trade skill B is 25 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 10 hours and the average daily requirement for B is 500 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" of a given 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 McDonald [1980b].

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 [Baker, 1974; Dannenbring, 1977; Gupta, 1971]. In addition, flowshop research has concentrated on academic objectives, such as minimizing makespan or minimizing maximum lateness, rather than requirements such as leveling the day-to-day requirements for resources, or some other objective that would be more useful to industry [Gupta, 1971]. For example, in a July, 1977 article, Dannenbring [1977] published "An Evaluation of Flowshop Sequencing Heuristics," wherein he discusses the concepts underlying 11 different flowshop scheduling techniques. All 11 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 [1971] 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 [1962]. 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

Baker, Kenneth R., 1974, *Introduction to Sequencing and Scheduling,* John Wiley and Sons, New York.

Dannenbring, D. G., 1977, "An Evaluation of Flowshop Sequencing Heuristics," *Management Science* Vol. 23, No. 11, pp. 1174–1182.

Godin, Victor B., 1978, "Interactive Scheduling: Historical Survey and State of the Art," *AIIE Transactions* Vol. 10, No. 3, pp. 331–337.

Gupta, J. N. D., 1971, "*M*-Stage Scheduling Problem, A Critical Appraisal," *International Journal of Production Reserach* Vol. 9, No. 2, pp. 267–281.

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

McDonald, Gerald W., 1980b, "User's Guide to the NARF-JAX Computer Aided Scheduling System (CASS)," Department of Industrial and Systems Engineering Technical Report 80-81, University of Florida, October.

Page, E. S., 1962, "On the Schedule of Jobs by Computers," *The Computer Journal* Vol. 5, pp. 214–221.

Copyright 1981, by INFORMS, all rights reserved. Copyright of Interfaces is the property of INFORMS: Institute for Operations Research and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.