Order Number 8727484

# Setting due dates for computer based system development projects

Bock, Douglas Brian, Ph.D.
Indiana University, 1987



U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

**PLEASE NOTE:**

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __√__.

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print _____
3. Photographs with dark background _____
4. Illustrations are poor copy _____
5. Pages with black marks, not original copy __✓__
6. Print shows through as there is text on both sides of page _____
7. Indistinct, broken or small print on several pages _____
8. Print exceeds margin requirements _____
9. Tightly bound copy with print lost in spine _____
10. Computer printout pages with indistinct print _____
11. Page(s) __________ lacking when material received, and not available from school or author.
12. Page(s) __________ seem to be missing in numbering only as text follows.
13. Two pages numbered _____. Text follows.
14. Curling and wrinkled pages _____
15. Dissertation contains pages with print at a slant, filmed as received _______
16. Other ________________________________________

# SETTING DUE DATES FOR COMPUTER BASED SYSTEM DEVELOPMENT PROJECTS

Douglas Brian Bock

Submitted to the Faculty of the Graduate School
in Partial Fulfillment of the Requirements
for the Degree
Doctor of Philosophy
in the School of Business
Indiana University

July, 1987

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements of the Degree of Doctor of Philosophy.

James H. Patterson, Chairman

E. Wainwright Martin

Doctoral Committee

F. Robert Jacobs

July 22, 1987

A. Victor Cabot

## ACKNOWLEDGEMENTS

At the conclusion of a long and arduous task, one often pauses and reflects on the magnitude of the undertaking. During such reflection, the necessity to acknowledge the assistance of others is realized, for without their guidance and assistance, a successful ending would not be achieved.

I express my gratitude, first, for the efforts of my dissertation committee, Professors James H. Patterson (Chairman), E. Wainwright Martin, F. Robert Jacobs, and A. Victor Cabot. To Professor Jacobs, my thanks for providing the original research idea and for his continued support. Without his assistance, the research would have never been undertaken. To Professor Martin, my thanks for his sagacious comments and suggestions during the conduct of the research. Without his assistance, the hard questions might not have been asked or answered. To Professor Cabot, my thanks for his support and friendship. Without his support, I would have abandoned the project long ago. And finally, to Professor Patterson, my thanks for his dedication to excellence, his honest and oft needed guidance, and his sacrifice of the many hours required for the numerous reviews of my written work.

I also express my gratitude for the assistance of many other individuals. To Professor Robert P. Bostrom, my thanks for the kind words of praise when they were needed.

My thanks to Ms. Betty Watson for her assistance in wrestling with the IBM 4381 computer. And finally, my thanks to those individuals who read and commented on drafts of the proposal, to the corporate and government Management Information System departments and staff members who participated in the research, to those fellow doctoral students who provided support along the way, to the secretarial staff for their assistance in the preparation of the manuscript, and to the many others who assisted in some manner.

Last, I wish to thank my wife, Mary Ellen, for her support over the last five years. It is to her that I dedicate this work.

TABLE OF CONTENTS

LIST OF TABLES

## LIST OF FIGURES

## ABSTRACT

This research examines a heuristic, rule-based approach to the estimation of project due dates in Management Information System (MIS) departments. Due date estimation for computer based system development projects may be described as a multi-project, capacitated, multi-resource, dynamic, preemptive problem. Multiple projects compete for several classes of limited resources. The dynamic, continual arrival of new projects results in the preemption of resources due to project priority policies. In this dynamic environment, the estimation of accurate project due dates is a difficult and challenging problem.

The research is conducted in two phases. The objective of the first phase is the derivation of a model of computer based systems development for MIS departments using a life cycle development methodology. A series of field interviews are conducted with project managers and their supervisors from a convenience sample of eight MIS departments. The interviews are used to validate a general MIS department model. This model is used as the basis for the second phase of the research.

In Phase 2, a three-factor, full-factorial computer simulation experiment is conducted to test the relative effectiveness of combinations of due date setting, resource scheduling, and resource preemption heuristics. The due date and resource scheduling rules selected were previously examined in a general multi-project management environment. This research examines their effectiveness on the MIS multi-project problem. The MIS problem is representative of an environment which is markedly different from that used in previous research.

The results indicate that the effectiveness of selected due date and resource scheduling heuristics is very similar to that previously reported. The introduction of the resource preemption factor significantly affects the selection of appropriate resource scheduling heuristics by management. Recommendations are made for the selection of due date and resource scheduling heuristic combinations under the two preemption policies examined.

## 1.0 INTRODUCTION TO THE PROBLEM.

A 1980 survey of project managers and educators in the field of Management Information Systems (MIS) identified 13 major issues concerning the management of computer based system development projects [THAY80]. One issue, titled "Scheduling", was defined as the ability to accurately estimate the due date for a software development project (See Note 1). This is still a topic of interest to MIS practitioners and researchers. This research examines the due date estimation problem for MIS departments.

In MIS, due date estimation may be described as a multi-project, capacitated, multi-resource, dynamic, pre-emptive problem. An MIS department is normally engaged in the simultaneous development of several projects. A department has to allocate two main classes of limited resources, designer and technology, among competing projects. The problem is termed dynamic because the time horizon over which projects arrive for development is an infinite one. For this reason the department does not

----------------

Note 1. The term due date is used throughout this research to denote the estimated date by which a project will be completed and released to users for normal operation. This term is selected because it is found extensively in project management literature. Other literature may use the term completion date.

possess perfect knowledge as to future workloads. The department must also use the limited resources available to support requests for the maintenance of existing systems. These maintenance activities may preempt resources from projects under development with resulting delays.

Previous researchers have examined various aspects of the MIS due date estimation problem. For instance, one line of research has investigated methods of estimating the time required to write a computer program while other research has emphasized the development of measures of programmer productivity ([CHEN78], [CHRI81], [CHRY78], [CROS79], [GAYL71], [HENN69], [MYER78], [NORD70]).

Another line of research has evaluated techniques for estimating the duration time required for the development of individual computer based systems (see Note 2). Current methods utilized to estimate the duration time, and hence, arrive at an estimate for the project due date, vary markedly. Some departments rely on project management software termed "Project Control Systems" as decision aids in estimating project duration times. The majority of this software is based on the Program Evaluation and Review

----------

Note 2. The term duration time refers to an estimate of the time required to complete a project if all resources required are available and those resources are devoted exclusively to the project. The due date is *not* usually equal to the start date plus the duration time in the multiple project, constrained resource environment which is typical for computer based system development projects.

Technique (PERT), which is described in a number of books ([ARCH68], [MALC59], [MODE70]). Other departments establish estimates of project duration times by relying on techniques such as personal experience, analogy to similar projects, use of work factor analysis, use of the Standards Method, etc, as reviewed in Benbasat and Vessey [BENB80].

Although techniques are available for estimating activity and project duration times, the problem of accurately estimating the due date persists. In their recent text, The Management of Information Systems, Dickson and Wetherbe emphasize the need to overcome this problem. One result of poor estimating is that "few information system projects are completed on time or on schedule . . . consequently, MIS management's credibility suffers" [DICK85, pg 123]. MIS practitioner literature is replete with articles that reference projects which were either not completed on time or had poor performance relative to estimated due dates ([CANN77], [HARR83], [MARK85], [MERE85], [SHAN85]).

In general, the methods reported for estimating project durations fail to consider the capacitated nature of resources in the MIS department. This should not imply that it is impossible to achieve good activity or project duration estimates. The use of computer based project control systems and the compilation of databases containing information about previously developed systems provides the ability to closely estimate many types of project activities. "The use of standard procedures and standardized productivity tools has increased the MIS departments ability to deal with the complexity of systems development while eliminating much of the variation in programming and design" [SPRA86, pg 211]. However, the project duration estimate cannot simply be used to extrapolate a project due date. Such an approach would not consider the dynamic nature of the MIS due date estimation problem.

This research will examine a heuristic, rule-based approach to the estimation of project due dates. In general, good heuristics are "rules of thumb" that provide non-optimal, though "good" results when compared to other techniques and other heuristics. Heuristics are used when a problem exhibits characteristics that prevent its solution by optimization techniques as in the MIS due date estimation problem. Specifically, this research examines four heuristics for due date estimation. These heuristics utilize information concerning the capacitated nature of the MIS department resources as well as other information such as department resource scheduling policies, etc, in setting the due date.

At this point it is necessary to define the scope of this research relative to the system development methodology used by MIS departments. Computer-based projects may be developed utilizing various approaches. One current methodology is termed Prototyping. This is an iterative approach generally involving one user and one designer. The designer uses a fourth-generation language in the system development process. Due dates are not usually established for such projects, primarily due to the iterative nature of the process. For this reason, this research does not include the prototyping approach. A second methodology may be termed User-Development of systems. As the MIS department manager is not concerned with the due date for systems which are primarily user-developed, this class of projects is also excluded from this research. The third methodology is termed Life-Cycle Development Methodology (LCDM). The LCDM is the method used the majority of the time in MIS departments. This research concerns the selection of due dates for projects being developed under the LCDM.

MIS projects may be developed under varying environmental conditions. The performance of due date heuristics will depend on the variables and factors affecting the development process. Section 1.1 provides a discussion of the variables and factors in the computer based system development environment which may affect the project due date estimate. Finally, any research endeavor should examine questions of some significant importance.

Section 1.2 concludes this chapter with a brief discussion of the importance of the research and the contributions which it intends to make.

## 1.1 VARIABLES AFFECTING THE DUE DATE ESTIMATE.

The variables which affect the accuracy of the project due date estimate may be divided into the five classes as presented in Table 1.1. These classes have been derived from a review of MIS literature (principally [DICK85], [MARK85], [MART85], [SPRA86], [THAY80], and [WOOD85]) and from the conduct of interviews in eight MIS departments. (A detailed discussion of the results of these interviews is presented in Chapter 4.)

**1.1.1 DEPARTMENT ORGANIZATION.** The first class of variables affecting the accuracy of the project due date estimate is Department Organization. MIS departments may be organized with fixed design teams and permanently assigned team managers. Teams may be assigned the responsibility of supporting a particular functional area or set of systems for both new development and the maintenance of existing systems. Maintenance requests may also be assigned according to which team has slack resources available. Another approach is to establish separate development and maintenance sections. Design teams may be specially selected from the members of the development section [DICK85].

1) **Department Organization.**
   - Project Team Composure (Fixed Project Teams vs Specially Organized Project Teams).
   - Number of Designers Per Project Team.
   - Degree of Alignment of Fixed Teams to Functional/Systems Areas.
   - Assignment of Responsibility for Maintenance Support of Existing Systems.

2) **Resource Characteristics.**
   - Heterogeneity of Designer Resources (Number of Skill Levels).
   - Specialty Skills of Designer Resources.
   - Interchangeability of Designer Resources Among Skill Levels.
   - Availability of Technological Resources.

3) **Project Characteristics.**
   - Type (Maintenance vs New Development).
   - Type Mix (Percentage of Maintenance vs New Development).
   - Priority for Development.
   - Uncertainty of Project Requirements (Political Factors).
   - Number of Activities per Project.
   - Shape of Project Activity Network.

4) **Activity Characteristics.**
   - Type of Activities.
   - Number of Resources by Type Required.
   - Activity Duration (Stochastic).
   - Probability of Reworking the Activity.
   - Size of Rework Activity Loops.

5) **Managerial Decisions and Policies.**
   - Degree of Inter-Team Flexibility of Assignment to Project Activities for Other Teams.
   - Amount of Overtime to Incur.
   - Whether to Contract Design Work to Outside Agencies.
   - Preemptive or Non-Preemptive Assignment of Designers to Activities.
   - Internal vs External Selection of Project Due Date.

**Table 1.1 Independent Variables Affecting the Accuracy of the Project Due Date Estimates**

**1.1.2 RESOURCE CHARACTERISTICS.** The second class of variables is Resource Characteristics. Designer resources are heterogeneous with different individual skills and levels of experience. MIS departments typically classify designers into skill classes which reflect these differences. The skill classes are used to assign designers to activities. In some instances, specialty skills may be recognized. These include skills in database design, telecommunications programming, etc. There is considerable interchangeability of skill levels for assignment of designers to activities. For example, an activity that may best be completed by a senior programmer can be assigned to a junior programmer.

Technological resources may also require allocation to project activities. This class of resource is less limiting than the designer class of resources. Technology may be easily shared by designers. Project delays may result from the need to acquire new technology (hardware or software) or from temporary maintenance requirements such as short-term hardware/software failures, upgrading of systems, etc.

**1.1.3 PROJECT CHARACTERISTICS.** The third class of variables is Project Characteristics. There are two categories of projects, Development and Maintenance. Development projects include requests for new systems or for significant changes to existing systems. Development projects may vary in duration from two or three weeks to several years. Maintenance projects include requests to modify existing systems. The duration of maintenance projects varies from a few hours up to about two weeks. The workload mix of development versus maintenance projects varies from firm to firm. Firms have reported as little as 20% of their total workload to be maintenance while others have claimed maintenance occupies 80% of their time ([BART86], [MARK85], [MART85]).

Priorities for development projects may be established by user management, a corporate steering committee, or the MIS department [DICK85]. The priority listing is analogous to the projects joining a queue where they wait for resources to become available. Maintenance projects usually receive a high priority for resource allocation [MARK85]. This occurs when the maintenance request requires immediate attention, i.e. a non- functional system. It is important to recognize that this can result in maintenance projects preempting resources that are or could be assigned to development projects. Because maintenance requests are not predictable, their occurrence adds to the dynamic nature of the problem. As maintenance consumes available resources, the completion of development projects is delayed.

Both development and maintenance projects may be represented as a network of activities. Sample network diagrams are provided in Figures 1.1 and 1.2 using the Activity-On-Node representation. Although maintenance projects may consist of several activities, they are often represented as a single activity since it is typical to assign one designer the responsibility for an entire maintenance project [SPRA86]. Larger maintenance efforts may be represented by a network of several activities where some of the activities may be completed in parallel. In these instances, several developers may be assigned to activities.

MIS project networks may be differentiated from project networks in general by several characteristics. MIS development project networks tend to be linear and appear to be composed of smaller project networks interconnected by one or a few activities. The linearity and shape of these networks exists because of the LCDM methodologies employed in designing and developing new systems. Figure 1.3 presents an example LCDM methodology [DICK85]. Such methodologies identify critical milestones in the development process. These milestones require the completion of most of the activities within a development phase prior to the initiation of the next development phase [DICK85]. In general, the network will have more activities in the middle phases of the development process ([MART85], [NORD70], [PUTN78]).

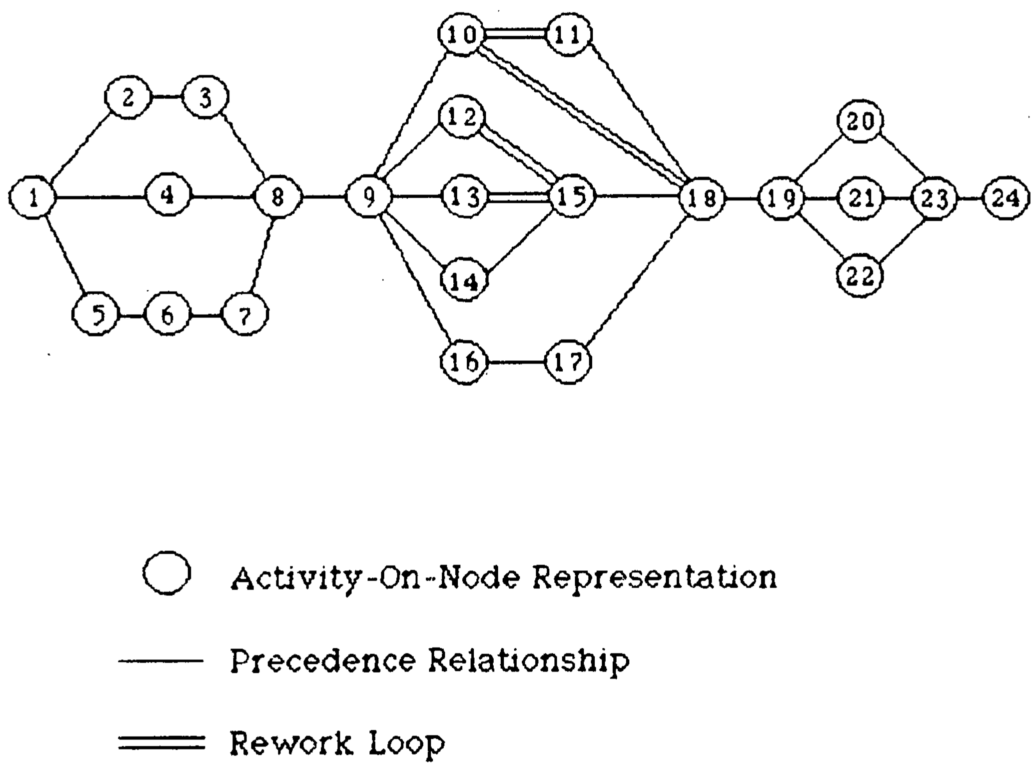
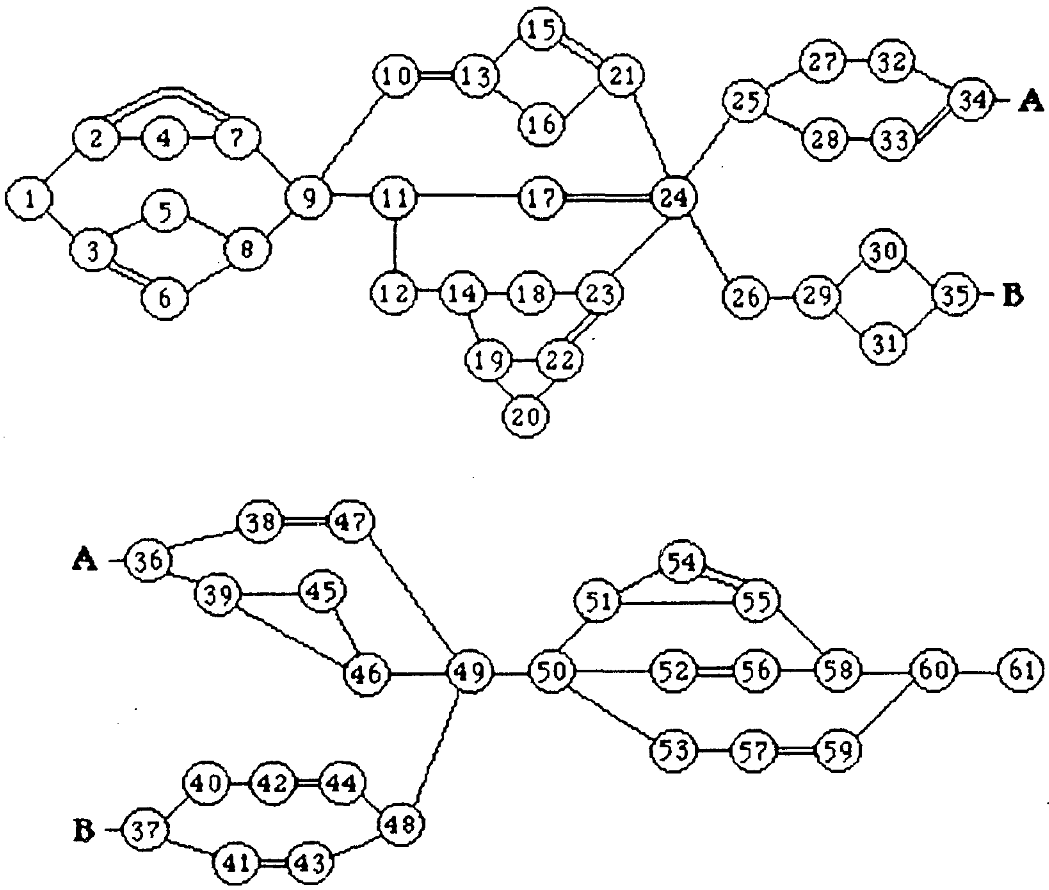
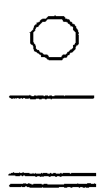



**Figure 1.1 Development Project Network #1**





Activity-On-Node Representation

Precedence Relationship

Rework Loop

**Figure 1.2 Development Project Network #2**

```
                              Analyze
 ----------->                Problems/
  |                        Opportunities
/ | \                            |
  |                              |
  |                            \ | /       Due Date Estimate
/ | \<-------->               Design  <--- Established During
  |                           System       This Phase For
  |                              |         Development
  |                              |         Projects
  |                            \ | /
/ | \<-------->             Develop and
  |                         Test System
  |                              |
  |                              |
  |                            \ | /
/ | \<-------->              Implement
  |                           System      <--- Due Date
  |                              |              for Project
  |                              |
  |                            \ | /
/ | \<-------->           Operate/Maintain
  |                           System
  |                              |
  |                              |
  |                            \ | /
  |                          Evaluate
  |<-----------             Post/Audit
                              System
```

**Figure 1.3. System Development Cycle [DICK85, pg 322]**

**1.1.4 ACTIVITY CHARACTERISTICS.** The fourth class of variables considered is Activity Characteristics. Individual activities may require varying quantities of resources. MIS activities typically require only one unit of a particular skill class of designer. It is also possible for an activity to require several designers of different skill levels and multiple units of designer resources within a skill level. An example of such an activity would be the presentation of a formal systems proposal to management.

Activity duration is stochastic. The duration of an activity is dependent on several factors including the skill of the designer and the degree to which the activity has been well defined.

Some activities may require rework. This occurs due to a poor design or a lack of understanding between the designer and the user as to the requirements of the project. Rework activities are represented as cyclic loops within project networks. These loops are usually small. Most managers agree that large rework loops representing the redesign of entire phases of projects do not exist in well managed departments ([BALL77], [MERE85]).

**1.1.5 MANAGERIAL DECISIONS AND POLICIES.** The fifth class of variables is Managerial Decisions and Policies. Designers may be assigned to activities from projects that belong to other teams. This is termed interteam flexibility of assignment. Management must also determine what policy to use relative to the preemption of resources by maintenance projects.

When the limited supply of designers adversely affects the ability of the MIS department to meet scheduled project due dates, designers may be required to work overtime. MIS departments tend to avoid this situation. Alternative solutions to designer shortages include contracting systems development projects to vendors or purchasing turnkey system packages [DICK85].

The project due date may be set internally by the MIS department or established by an external agency such as a corporate steering committee. For maintenance projects, due dates are not normally established since they tend to require completion as soon as possible.

## 1.2 THE IMPORTANCE AND CONTRIBUTION OF THIS RESEARCH.

This section has two purposes. First, it discusses the importance of research concerning the due date estimation problem. Second, it outlines the contributions arising from the conduct of the research.

**1.2.1 IMPORTANCE OF THE RESEARCH.** There are several reasons why the estimate of the due date is important from a practical standpoint. First, the due date represents a "goal" for management. Managers are often evaluated on their ability to reach this goal. In the introduction to this research, it was emphasized that a large group of project managers identified the ability to schedule and plan in the project area to be one of 13 critical areas for concern [THAY80].

Another reason for the importance of this research concerns the expectations of the potential users of a system being developed. If the users had not identified a need for the development or change in a system, the project request would not have been made. When a project request is accepted, it is often assigned a priority for development by a centralized steering committee. Once a project is initiated, users may view this as a signal that it now has priority for resources and for development. At this point users are often given an estimate of the due date and they tend to rely on it. Their perception of the success of a project is influenced by whether or not the project is completed in a timely manner [DICK85]. The following quote from a practitioner journal underscores this point.

> "Our Information Services organization is always behind. They never seem to get things done. I wish I knew what those people in DP are doing..." [MARK85]

A third reason for the importance of the due date estimate concerns the allocation of resources and the backlog which exists in an MIS department. The estimated due date, to a degree, represents the commitment of some quantity of a scarce resource, the designer, to a project. Conversely, this means that some other project which is in the firm's backlog is waiting for this resource to become available. An underestimated due date can have a serious rippling effect on the allocation of designers to all existing projects. Attempts to get back on schedule may result in requiring overtime work. Further, managers may assign additional designers to a project to provide priority for selected activities. It has been noted, however, that human resources cannot be arbitrarily substituted for time [TRIP80]. The additional help may even delay the project because of the necessity to bring the new project members "up to speed" on the status of project activities [BROO74]. Where designer resources are scarce, those added to a project are generally pulled from some other project. This may delay the other project to some extent.

A due date which overestimates the time required for development is also bad. The due date estimate may tend to become a self-fulfilling prophecy with work expanding to fill the time available.

**1.2.2 CONTRIBUTIONS OF THE RESEARCH.** In outlining reasons which underscore the importance of the due date estimate, the first of several potential contributions of the research has been introduced.

First, this research addresses an area which is recognized as being important in the practitioner community. Due date estimation is important to users because their ability to perform their jobs is often dependent on the computer-based systems available. MIS managers are concerned because the delivery of these systems is their responsibility.

A second contribution concerns the study of multiple project due date estimation in general. Research on the dynamic, capacitated, multiple project problem has not been reported to a great extent in the literature. The examination of due date estimation for MIS departments builds on the work of earlier researchers and extends the line of research to include the MIS environment.

A third contribution of this research is the development and empirical validation of a model of the MIS multiple project environment. Although the attributes of such a model might seem obvious, no current model description exists in the literature. The attributes must currently be confirmed by reading of numerous MIS periodicals and from discussions with practitioners and academics. Even recent textbooks published in the MIS field do not present an explicit and comprehensive description of such a model ([DICK85], [SPRA86]). The model should prove very useful to MIS managers. The comprehensive discussion of all relevant characteristics in the multiple project environment will provide a reference guide for managerial decision-making.

## 2.0 LITERATURE REVIEW.

Previous research relevant to the due date estimation problem may be divided into three categories, (1) Project Management literature, (2) Job Shop Scheduling literature, and (3) MIS literature. Section 2.1 introduces a taxonomy that is a useful framework in which to classify project due date literature. This framework may be used to classify both project management and job shop literature. Section 2.2 reviews research that has examined various scheduling heuristics used to allocate resources. It also reviews project management and job shop research concerned with due date estimation. Section 2.3 reviews MIS literature by focusing on various techniques that have been used to estimate duration times of MIS activities and projects.

### 2.1 A PROJECT MANAGEMENT RESEARCH TAXONOMY.

The study of project management is a vast domain. In order to restrict the review of literature to that relevant to this research, it is useful to employ a taxonomy based on certain characteristics of the due date research. Such a classification scheme is also helpful in determining where an area has been heavily studied and where there is a need for future work. Table 2.1 presents such a taxonomy. The research in Table 2.1 is divided into three areas: (1) Static Single Project Research, (2) Static Multiple Project Research, and (3) Dynamic Multiple Project Research.

| Researchers | Technique | Criterion | Max # of Resources Tested | Max # of Activities Tested | Max # of Projects Tested |
|---|---|---|---|---|---|
| **STATIC SINGLE PROJECT RESEARCH** | | | | | |
| Davis & Heidorn [DAVI71] | Bounded Enumeration | Completion Time | 3 | 30 | |
| Patterson [PATT84] | Survey | Completion Time | 3 | 50 | |
| Stinson, Davis & Khumawala [STIN78] | Branch & Bound | Completion Time | 6 | 43 | |
| Talbot & Patterson [TALB78] | Integer Programming | Completion Time | 3 | 27 | |
| Cooper [COOP76] | 26 Heuristics | Completion Time | 3 | 30 | |
| Davis & Patterson [DAVI75] | 8 Heuristics | Completion Time | 3 | 30 | |
| Holloway et al. [HOLL79] | Multi-Pass Heuristic | Completion Time | Multiple | NS | |
| Thesen [THES76] | 1 Heuristic w/Knapsack Algorithm | Completion Time | 10 | 200 | |
| Wiest [WIES67] | Combinations | Completion Time | 15 | 600 | |
| **STATIC MULTIPLE PROJECT RESEARCH** | | | | | |
| Pritsker, Walters, & Wolfe [PRIT69] | Zero-One Integer Programming | Total Throughput, Makespan, Lateness | 3 | 3 | 3 |
| Fendley [FEND68] | 8 Heuristics | Project Slippage, Resource Usage, Inventory | 3 | 20 | 5 |
| Kurtulus & Davis [KURT82] | 9 Heuristics | Total Throughput | NS | 63 | NS |
| Patterson [PATT73] | 7 Heuristics | Total Throughput, | 7 | NS | 34 |
| Wiest [WIES67] | 1 Heuristic | Not Specified | NS | NS | 107 |
| **DYNAMIC MULTIPLE PROJECT RESEARCH** | | | | | |
| Dumond [DUMO85] | 7 Sched & 4 Due Date Heuristics | Completion Time & 6 Others | 3 | 49 | 20 |

Legend: NS=Not Specified

**Table 2.1 Project Management Research [DUMO85]**

In the _static single project_ problem, all available resources are exclusively devoted to the one project. The term _static_ implies that all resource requirements are known in advance because only one project will require resources. This class of problems may be further described according to whether it is resource capacitated or uncapacitated. In the uncapacitated problem, the objective is usually to minimize the duration (makespan) of the project. Critical path analysis may be used to identify critical activities, which if delayed, will delay the completion of the project. Research on this problem has primarily focused on sequencing activities to give priority to the identified critical activities. If the problem if capacitated, the objective is to minimize the duration time of the project which may increase beyond the computed critical path time [DUMO85].

In the _static multiple project_ problem, several projects are concurrently under development. The activities of these projects will simulataneously compete for the resources available. Again, the term _static_ is used because it is known that this set of projects represents all projects which will compete for resources. As in the single project capacitated problem, it is necessary to prioritize the assignment of resources among the competing activities. In this situation, activities are often being completed in a parallel fashion from more than one project concurrently. Further, due to the large number of activities, the multiple project problem is combinatorially difficult to solve by the use of optimization procedures.

In the dynamic multiple project problem, several projects are again concurrently under development. However, the stream of projects competing for resources arrives to the MIS department over an infinite time horizon. The arrival of new projects for development is not fully predictable. New projects will create increased demands for the limited resources available and may tend to delay projects already under development. The term dynamic is used to describe this problem class.

For completeness, note that the dynamic single project area is not listed. The single project dynamic problem cannot exist since, by definition, the term dynamic requires a stream of several projects [DUMO85].

From the entries in these three areas, it is apparent that there has been extensive research on the static single project problem. Since the multiple project problem is an extension of the single project problem, much of the early work completed in the single project area is relevant to this research. The static multiple project problem has also been extensively researched. Research in this area is reviewed as it may provide direction in examining the dynamic problem. The third area, the dynamic multiple project problem, has received little attention. The only research reported is a simulation experiment conducted by

Dumond. This research will also be placed within this third area and will be the second reported study of the problem.

## 2.2 RESOURCE ALLOCATION AND DUE DATE LITERATURE.

**2.2.1 SOLUTION PROCEDURES.** A comprehensive review of project scheduling research was published in 1966 and 1973 by E.W.Davis ([DAVI66], [DAVI73]). Davis structured his review of constrained resource scheduling by focusing on the solution procedures employed in each area. Table 2.2 from Davis provides a classification scheme which summarizes the solution procedures available at that time. The Sampling Method was added from a later study by Cooper [COOP76].

| HEURISTIC PROCEDURES | OPTIMAL PROCEDURES |
|---|---|
| Parallel Allocation | Linear Programming |
| Serial Allocation | Enumerative/Other |
| Sampling Method | |

**Table 2.2 Classification Scheme for Solution Procedures [DAVI73, pg 301]**

Heuristic procedures involve the use of some "rule of thumb" which seeks to capture or take advantage of characteristics of the problem. A heuristic is used to determine the priority of activities which are competing for scarce resources. Heuristics are evaluated by their ability to produce a "good" solution. "Good" is often defined to be a solution which dominates a naive approach to the problem. An example of a naive approach would be the assignment of resources to activities on a purely random basis. Heuristic procedures can produce optimal or near optimal solutions in some cases. Their simplicity and ease of use makes them attractive. Two classes of heuristics have been examined. Serial heuristics assign priorities to competing activities before they are scheduled. Parallel heuristics assign priorities during the scheduling process. Parallel heuristics generally dominate serial ones [DAVI73].

Optimal procedures employ some exact analytical approach which seeks to arrive at the "best" answer to a problem. Studies which report the use of optimal procedures are much fewer in number than those which report the use of heuristics. Davis provides explanations from several authors which explain the lack of use of optimal procedures. One of these by J.E.Kelley [KELL59] states that:

> "Formulating the resource loading problem from a mathematical point of view is difficult because explicit criteria for obtaining the optimal use of resources is lacking...mathematical techniques do not exist for obtaining solutions in a reasonable time..."

Thus the major argument has been that the size of the problem as measured by the number of activities and resources per activity is often too large combinatorially to be solved. Other arguments raised include the difficulty involved with formulating the problem when an integer linear programming approach is utilized. Perhaps a more cogent reason for not relying on optimal procedures in a multiple project dynamic problem lies in the nature of the environment. The continuous arrival of new projects in a stochastic manner, the variation in actual activity durations versus estimated activity durations, and the existence of rework loops would cause the computed optimal solution to in fact be non-optimal. In this situation, a heuristic procedure may be favored because its ease of use is attractive to practitioners. For this reason, the discussion and detailed review of single project literature will be restricted to research reporting the use of heuristic procedures, although several studies using other methods are listed in Table 2.1.

**2.2.2 STATIC SINGLE PROJECT RESEARCH.** While the intent of this research is to focus on the multiple project problem, much of the research concerning the single project static problem provides a foundation for analyzing the multiple project problem.

In a 1975 study, Davis and Patterson evaluated the performance of eight heuristics on a set of 83 single projects for which optimal solutions were known. While literally hundreds of heuristics exist, they selected those which had been predominantly tested in the literature and employed by practitioners in single project and job shop environments. The objective was a comprehensive comparison of heuristics in a common test situation. The heuristics tested are briefly explained below.

The Minimum Activity Slack (MINSLK) rule gives priority to those activities which have the lowest slack time as computed using PERT techniques.

Another critical path related rule, the Minimum Late Finish Time (LFT), gives priority to those activities with the earliest values for late finish time as measured using PERT techniques.

The Resource Scheduling Method (RSM) is used to develop a priority index for activities. This index is calculated from a pairwise comparison of activity early finish and late start times. Activities are prioritized basically in the order of increasing late finish times.

The Greatest Resource Demand (GRD) method considers the resource demands of activities. Those with the greatest demand are scheduled first. Similar to this are the Greatest Resource Utilization (GRU) and Most Jobs Possible (MJP) rules. The GRU rule gives priority to the group of activities which results in the minimum amount of idle resources for a given scheduling interval. The MJP rule gives priority to the largest possible group of activities that can be scheduled in a time interval.

The Shortest Imminent Operations (SIO) rule is often called the shortest activity first. Activities with the shortest durations are given priority thereby completing the largest number of activities possible within a given time-span.

Other rules which are often used as baseline measures include the Random Activity Rule (RAN) which assigns resources randomly and the First Come, First Serve (FCFS) rule which assigns resources to the activities in the order in which they are started.

Davis and Patterson tested these heuristics, with the exception of the FCFS, using a performance measure of minimizing project duration. The baseline for performance evaluation was the optimal solution available for each of the different projects. The best performers were MINSLK, LFT, and RSM. MINSLK found the highest number of optimal solutions (29%). The heuristics which were based on resource requirements and the SIO heuristic did not perform as well as the RAN heuristic. Of the 83 problems, 60% of them were not solved optimally by any of the heuristics. An important point to note is that some of the better performers developed very poor solutions to some of the problems.

A 1976 study by Cooper tested 26 different heuristics using two different approaches [COOP76]. One approach was the parallel approach. The other was a sampling method. Only one of the heuristics examined considered the availability of resources. It performed the best.

The sampling method was used to test 14 of the heuristics. A large number of schedules were generated using the heuristics and the best schedule was deemed to be the solution. The sampling procedure improved over the performance of the parallel approach for the heuristics tested using both procedures. The best performing heuristic from the parallel approach, however, was not used in the sampling procedure because it was not amenable to the procedure.

None of the 14 heuristics performed significantly different from one another.

Holloway, et al [HOLL79] developed a complex multi-pass heuristic which addresses the multiple resource problem. It decomposes the problem into sets of single resource problems. While this procedure performed significantly better than the MINSLK heuristic and produced near-optimal solutions, it was restricted to projects where a given activity would only require a single resource at a time. Thus, it is of little interest for most computer based system development projects.

While other research exists which has addressed the single project static problem, the literature reviewed in this section is representative of that which has been reported. The points most relevant to this research include the recognition of the origins of many of the heuristic procedures which have been extended in their original form or in a modified form to the multiple project problem area.

**2.2.3 STATIC MULTIPLE PROJECT RESEARCH.** A study by Pritsker et al is representative of the literature which proposes approaching the multiple project problem with mathematical programming procedures [PRIT69]. The solution technique employed was zero-one integer linear programming. The scope of the solution procedure is fairly broad, but it lacks some of the attributes desirable in order to fully model a computer based system development environment.

These shortcomings are described in the paragraphs which follow.

The procedure facilitates the application of a limited resource constraint with substitution among resources. It provides for precedence relationships among activities and allows formulations that prescribe the concurrent or non-concurrent performance of activities. Multiple projects with varying numbers of activities can be solved. Multiple resources may be scheduled.

Pritsker's research built on the earlier work of Bowman, Wagner, and Manne ([BOWM59], [WAGN59], [MANN60]). Neither the formulation by Manne nor Wagner provided for multiple resources. Bowman's approach accommodated multiple resources, but the resulting formulation would be a larger set of variables and constraints than would arise using the method presented by Pritsker et al.

A limited comparison was made between the LP procedure and the two heuristics MINSLK and FCFS for a three project, eight job, three resource problem. The LP procedure dominated the heuristics. The authors utilized three performance measures. These were: (1) minimize total throughput time for all projects, (2) minimize the time for all projects to be completed, and (3) minimize total lateness for all projects.

A major shortcoming of LP procedures in general are that they require all projects to be known in advance. This limits their applicability to a static environment, although it is possible to treat the dynamic problem as a series of static problems.

Fendley employed discrete event simulation as a methodology in examining the multiple project, constrained resource problem [FEND68]. Within the context of this methodology, scheduling heuristic procedures developed for the single project problem are often employed. Fendley defined the scope of his research such that resources are fixed, project activity times are stochastically determined, and due dates are set internally. Three types of resources are allocated among activities.

In the first stage of his experiment, eight projects were formulated with up to 20 activities each. Activity times were generated using a Beta distribution. PERT estimates of the mean and variance were developed for the activities. These estimates were used to measure the baseline performance of the various scheduling heuristics examined. Eight scheduling rules were initially tested including the SIO, MINSLK, FCFS, and several other rules. Initially combinations of two projects were utilized in the experiment. The MINSLK rule performed best for the performance measure of project slippage. None of the rules dominated for all of the performance measures selected.

In the second stage of his experiment, Fendley chose the MINSLK rule to be used to schedule resources. He developed a regression model to predict project due dates. This model was used to predict due dates for the eight projects in a multiple project static environment. As might be expected with regression models, the predictions were good when projects required resources within the ranges of the data used to develop the regression parameters. When requirements fell outside of these ranges, the due dates predicted were not accurate.

In another study of the multiple project problem, Patterson tested five scheduling heuristics which worked well on single projects [PATT73]. The evaluation criteria included total project delay, weighted project delay, and resource utilization. The SIO rule performed best as evaluated against total project delay while MINSLK performed best as evaluated against the weighted project delay.

A recent study by Kurtulus and Davis evaluates nine scheduling heuristics [KURT82]. Six of the rules were developed by Kurtulus specifically for the multiple project problem. The other three rules were MINSLK, FCFS, and SIO. One of the best performing rules was the Shortest Activity from Shortest Project (SASP) rule which gives priority to the activities which require the least time and which come from the smallest projects. This rule seems to get at the idea of reducing the congestion of the problem space and appears similar to the shortest processing time heuristic used in research on scheduling of the manufacturing job shop.

A second rule, Maximum Total Work Content (MAXTWK), performed as well as the SASP heuristic. This rule recognizes the amount of resources which activities can require and hence uses more information than some of the other rules. It should be noted that the MINSLK rule was the next best performer.

**2.2.4 DYNAMIC MULTIPLE PROJECT RESEARCH.** The only research reported in this area is a simulation experiment conducted by Dumond [DUMO85]. As this research builds upon that of Dumond, his study will be examined in more detail than that of previous researchers.

Dumond focuses on the development of heuristics for use in estimating due dates for projects. A two-factor, full-factorial experimental design is used to evaluate seven resource allocation heuristics and project four due date estimating heuristics. The performance measures included project mean completion time, project mean lateness, project standard deviation of lateness, and the total tardiness for all projects, as well as others.

No significant difference was found among the scheduling heuristics, although it was interesting that the FCFS rule tended to perform reasonably well. Some of the rules which were completion date oriented tended to slightly dominate those which were not, although performance varied depending on the performance measure utilized.

The project due date estimating heuristics were found to be significantly different. The best performer was a rule which finitely schedules a new project with projects currently under development to arrive at an estimate of the due date.

The environment modeled includes projects which are composed of from six to 49 activities. The mean number of activities is 24. Three resource types are allocated. An activity can consume from one to 39 units of a resource as well as more than one resource type. Resources are assigned in a non-preemptive fashion. Activity duration times are deterministic. Projects arrived to the simulator at random with an interarrival rate set at eight days generated from a uniform distribution. Projects are fairly homogeneous. A sensitivity analysis phase was conducted where various percentages of the projects arriving had externally set completion dates. Thus the model examined by Dumond is multi-project, multi-resource, capacitated, and dynamic.

This model is very similar to the computer based system development environment. In fact, Dumond states that his results are generalizable to the due date estimation problem for software development projects. However, his model of the problem structure does not accurately portray many of the variables which affect the accuracy of the due date estimate for MIS development projects. His project networks do not exhibit the linear shape that is representative of MIS development project networks. The networks do not have rework loops to represent redesigning and recoding requirements that regularly occur in MIS projects. The activity times are modeled as deterministic, in Dumond's research, whereas stochastic activity times would increase the variability and duration of the projects during the simulation of the development process. The resource requirements (for some individual activities) are higher than would be experienced by an MIS department. Dumond does not model two different types of projects, i.e. maintenance and development, but treats all projects similarly. This treatment also does not allow for the preemption of resources to meet maintenance project requirements. Thus, the degree to which his results are generalizable to MIS project management is questionable and is evaluated in this research.

This completes the review of multiple project, management scheduling literature. It should be noted that much work has been completed in these areas which is not reported in the literature. Davis makes reference to this situation in his review article [DAVI73]. These studies have been conducted by individuals or firms developing project management software and most of their results remain proprietary.

**2.2.5 OTHER DUE DATE RESEARCH.** While some of the research reviewed has addressed due date estimation, other literature is available which has also reported the use of heuristic rules in making due date estimates. The majority of this literature comes from the job shop area due to a lack of reporting in project management literature. The similarities between the job shop and project management areas are emphasized by Davis [DAVI73] and reviewed by Dumond [DUMO85]. There has been much overlap in the study of these two areas and many of the heuristics developed in the job shop area must be considered as candidates for applicability to project management due date estimation. Although project management and job shop research is similar, the review of job shop research is separated here to emphasize the fact that it comes from a different body of literature.

A study by Weeks and Fryer examined four due date estimating heuristics for dual-constrained job shop production systems [WEEK76]. One rule very simply added a constant time period to the time when the job was initiated in order to arrive at the due date estimate. The other three rules used multiples of the estimated duration time for the jobs to assign due date estimates. Another study by these authors demonstrates how to select these multiples by use of regression analysis [WEEK77]. They noted that "tight rules performed better than loose" rules for the performance measure of mean flow time and variance of lateness. Tight means that the due date estimate provides little slack in the project as measured using critical path analysis. Conversely, loose rules provide considerable slack.

A study by Conway [CONW74] from the job shop literature as reported in [DUMO85] examined four methods of determining estimated due dates. A naive baseline heuristic is used to set the due date randomly. Two of the more sophisticated heuristics add slack time to the due date estimate by multiplying the estimate obtained by a constant factor. One of these uses a multiple of the total estimated job duration. The other uses a multiple of the number of operations to be performed on a job. These are tested in conjunction with six scheduling heuristics. The scheduling heuristics which considered information about the established due date estimate perform best in conjunction with the more sophisticated due date estimating heuristics.

Ragatz and Mabert provided a comparison of due date assignment heuristics and scheduling in a common test environment [RAGA84]. Eight due date assignment heuristics were selected as being representative of the literature. Three scheduling heuristics are used. A two-factor, full-factorial research design methodology is employed. The performance criteria are Standard Deviation of Lateness, Mean Absolute Missed Completion Dates, and Mean Tardiness.

Of the due date rules examined, the Jobs in Queue (JIQ), Work in Queue (WIQ), and Response Mapping Rule (RMR) clearly dominate across all performance criteria. The JIQ rule captures an idea of shop congestion while the WIQ rule provides some idea of the amount of work which is waiting in queue to be processed. The RMR is a more sophisticated rule which uses response surface mapping procedures to "identify important independent variables (X's) and estimate various functional rule equations".

This completes the review of due date estimating literature. Some observations to be made include, (1) the due date setting heuristics tend to perform better when they utilize more information concerning the problem, (2) due date oriented scheduling heuristics tend to outperform non-due date oriented scheduling heuristics, and (3) tight due date heuristics, i.e. those which provide little project slack, tend to improve performance over loose due date heuristics in terms of minimizing the mean project completion time.

## 2.3 PROJECT SYSTEM AND ACTIVITY DURATION ESTIMATING TECHNIQUES.

This section reviews MIS literature which describes techniques that have been reported as useful in estimating the expected duration time for projects or for activities which comprise projects. Table 2.3 provides a summary of the research in this area. The duration time for a project or activity is the time required to complete the work assuming that all resources required are available. Section 2.3.1 reviews methods of estimating the required duration time for entire project development systems. Section 2.3.2 reviews methods of estimating duration time for individual project activities.

**2.3.1 PROJECT SYSTEM ESTIMATING TECHNIQUES.** Table 2.4 identifies four methods which may be used to estimate the duration time for computer based system development projects. The first three are reviewed by Benbasat and Vessey [BENB80]. The fourth method, Project Management Decision Support Systems (DSS), overlaps somewhat with the Work Factors Method.

| Researchers | Estimation Technique | Applicable to Estimation of: |
| --- | --- | --- |
| Norden [NORD70] | Manpower Life Cycle Method | Project Duration |
| Gayle [GAYL71] | Parametric Equations (Multiple Regression) | Programming Costs |
| Weiss [WEIS73] | Work Factors Algorithm | Activity/Project Duration |
| Burton [BURT75] | Work Factors Algorithm | Activity/Project Duration |
| Donelson [DONE76] | Standards (Module Estimation Technique)<br>Proportion of Life Cycle | Activity Duration<br>Project Duration |
| Johnson [JOHN77] | Standards (Lines of Code Method) | Programmer Productivity<br>Activity Duration |
| Chrysler [CHRY78a]<br>[CHRY78b] | Parametric Equations (Multi Regression)<br>Parametric Equations | Programmer Productivity<br>Programmer Performance |
| Putnam [PUTN78] | Manpower Life Cycle Method | Project Duration |
| Toellner [TOEL78] | SPECTRUM Project Control System | Activity/Project Duration |
| Crossman [CROS79] | Standards (Function Counting Technique) | Programmer Productivity<br>Activity Duration |
| Benbasat & Vessey [BENB80] | Personal Experience and/or Analogy (Discussed in this Survey Article) | Activity/Project Duration |
| Christensen [CHRI81] | Standards (Operand/Operator Counting Method) | Program Complexity |

**Table 2.3 MIS Activity & Project Duration Estimation Literature**

---

1. Proportion of Life Cycle Method
2. Work Factors Method
3. Manpower Life Cycle Method
4. Project Management DSSs

**Table 2.4 Systems Development Estimating Methods**

The Proportion of Life Cycle Method uses extrapolation to estimate the total project duration. The extrapolation is based on an estimate of duration time required for the programming phase of a project. Donelson uses an approach similar to this to calculate project costs [DONE76]. A shortcoming of this method is that project managers are usually required to provide an estimate of the project duration at the time of the completion of the detailed investigation into the project's feasibility. This duration estimate is then used in conjunction with other available information to establish an estimated project completion date. At this point in the project's life, a good understanding of the exact programming to be required may not be clear. Such an understanding may not be available until the logical design phase is well under way. This renders the estimation method infeasible in many instances since the extrapolation is based on the programming required in the project.

Weiss published a 1973 article suggesting the use of Work Factor tables in estimating. These tables provide estimates of activity durations based on subjective evaluations as to whether the activities are simple, simple/complex, complex, or very complex [WEIS73]. Evaluations also consider a weighting factor for programmer/analyst experience. The project duration time is the sum of the estimates for all activities in the estimation methodology.

The Manpower Life Cycle Method is a more sophisticated method. This method is a heuristic approach which makes use of analytical procedures to achieve an estimate of the project due date. Norden introduced the idea that software development projects are composed of cycles which correspond to the phases of life cycle development methodologies [NORD70]. These individual cycles may be analytically described by a Rayleigh equation of the form given below.

$$Y = 2Kat * \exp(-at^2)$$

where Y = Manpower utilized in a time period.
K = Total cumulative manpower used on a project in units of man-years.
a = a shape parameter.
t = time in years.

Further, the individual curves resulting may be combined to represent the total system development effort. When combined, the result is also a Rayleigh equation. The curve described by this equation is provided as Figure 2.1.

Putnam examined this concept and found that the time of maximum manpower utilization, which is represented by the modal point of the curve, corresponds to the time when a system becomes operational [PUTN78]. This holds true for very large systems when the curve represents the system's useful lifetime. Putnam uses data collected from large projects in conjunction with regression techniques to estimate the parameters "K" and "a" as well as the modal point of the equation. A limitation of this method is that it has only been claimed to work for "very large" projects.

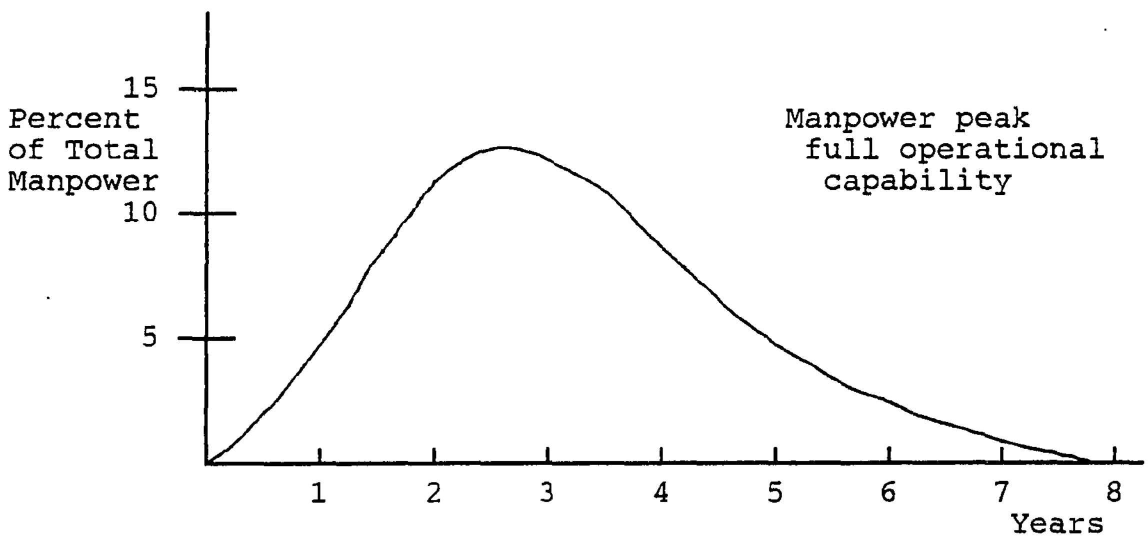



**Figure 2.1 Rayleigh Curve [NORD70]**

Very large projects are defined as those that take two man-years to reach an operational state.

A fourth method to obtain project duration estimates is through the use of a Project Management Decision Support Systems (DSSs). Various practitioner reviews have examined available mainframe and microcomputer packages by comparing features in terms of number of projects, number of activities per project, number of resources per activity, etc, which the packages can accommodate ([FILL86], [HARR83], [PETE79], [ZEMK84]). Many of these DSSs employ PERT and CPM methodologies. These packages usually weight factors such as programmer experience, activity difficulty, etc. As such, they combine elements of the Work Factors Method with elements of PERT. The resulting due date estimate is the sum of all activities which make up the phases of the project. They are usually based on a life cycle design methodology.

**2.3.2 INDIVIDUAL ACTIVITY DURATION ESTIMATING TECHNIQUES.** The literature concerning techniques for estimating individual activity duration principally reports on estimating the time required to write computer programs. This literature is reviewed because it is important to be able to estimate duration time for individual activities if the project due date is SET by the use of "bottom-up" estimating techniques.

Table 2.5 lists five methods which may be used to estimate individual activity duration times [BENB80]. These are listed in order of increasing sophistication.

---

1. Personal Experience
2. Analogy
3. Work Factors
4. Standards
5. Parametric Equations

**Table 2.5 Activity Duration Estimating Techniques**

---

The Personal Experience method is the most subjective of those listed. It is often used to estimate the duration time required for smaller projects which may be viewed as consisting of a single activity.

Analogy requires the retention of data from past projects developed by the firm. Simple analogy draws on experiential memories from individual mental databanks. Simple analogy is synonymous with the Personal Experience category given above. A more rigorous application of this technique requires the user to identify projects with similar functional elements, input/output tasks, etc, in order to formulate an estimate of the duration time required for a computer program or set of activities.

The Work Factors method at the individual computer program or activity level is similar to that reported at the systems estimating level. It is usually based on some estimating equation which is specified by several parameters. These parameters, or factors as they may be called, are extracted from sets of tables developed by the firm on the basis of past project experience. Some of these factors might include input complexity, output complexity, programmer experience, etc. A sample equation is reported by Weiss [WEIS73]. Similar studies include [BURT75] and [TOEL78]. These techniques have been criticized for their generality, subjectivity, and requirement to estimate factors which may not be known prior to completion of the activity [HENN69].

The Standards Method represents a class of approaches which seek to estimate activity duration times based on either (1) the number of lines of code which will be required for a computer program [JOHN77], (2) a count of operands and operators in a computer program which provide a measure of complexity [CHRI81], (3) the number of modules and module types required for a computer program [DONE76], (4) the number of functions in a computer program [CROS79], or some similar method. These methods require the use of an extensive database. Some tend to provide estimates when they are needed the least, that is once the activity is completed. Some of these techniques may work satisfactorily for well defined activities, given an adequate database.

The use of Parametric Methods to estimate activity durations represents a sophisticated approach to estimation ([CHRY78a], [CHRY78b], [GAYL71]). As with all regression models, estimates for activity durations which are similar to those found in the database will probably be satisfactory. Activities with durations which lie outside of the ranges specified within the database will probably be poorly estimated. Of course, parametric equations cannot be developed without a database of past projects.

This completes the review of literature concerned with estimating duration times for individual programs or activities. It should be noted that many of the techniques reported are lacking in sophistication and may not be useful because they are post-hoc techniques, i.e. they provide estimates assuming a great deal is known about the activity. Such information may only be available when the activity has been completed. This should not to imply that it is impossible to achieve good estimates for activities that make up projects. As was noted in Chapter 1, the employment of project management decision support aids and the development of databases containing information about previously developed systems has made it possible to closely estimate the duration for many types of activities.

## 3.0 RESEARCH METHODOLOGY, EXPERIMENTAL TREATMENTS, AND EXPERIMENTAL DESIGN.

This chapter provides a detailed discussion of the research methodology, experimental treatments, and experimental design. Section 3.1 discusses the research methodology selected and the development of a MIS department model used in the due date setting problem. Section 3.2 provides a detailed explanation of the due date, scheduling, and resource preemption heuristics which comprise the experimental treatments. Section 3.3 discusses the experimental design, data analysis procedures, sample size selection, and the expected results of the experimental treatments.

### 3.1 RESEARCH APPROACH TO THE DUE DATE SELECTION PROBLEM.

The selection of an appropriate research methodology is primarily dependent on the nature and number of experimental treatments to be examined, i.e. the methods used for setting the project due dates, scheduling resource usage, and preempting resources. The research will be more productive when the performance of several heuristics from each treatment factor are evaluated.

The conduct of a field experiment in a representative MIS department is one potential approach to addressing the problem examined in this research. This approach was not selected for several reasons. First, it is not viable because a firm was not found that would allow such experimentation. Second, this approach would consume an inordinate amount of time and resources. Third, the extensive time required to conduct the experiment would limit the number of heuristics which could be examined and reduce the benefit received from the experiment.

The alternative approach (selected for this research) is to evaluate due date heuristic performance using a simulation methodology. Simulation is selected because it is viable in terms of the time and resources required for completion. It also allows a large combination of experimental treatments to be examined in a full factorial experiment.

This approach necessitates the development of a computer model of an MIS department. To accomplish the task of model development and experimentation, a two phase research methodology is employed. This two phase methodology is explained below.

**3.1.1 PHASE 1.** Phase 1 is divided into two sub-phases; Phases 1A and 1B. In Phase 1A, a review of MIS literature was conducted to identify the factors and variables affecting MIS systems development. This resulted in a preliminary model of this environment.

In Phase 1B, a series of interviews were scheduled with MIS professionals in large government and corporate MIS departments. The purpose of the interviews was to confirm and/or modify the preliminary systems development model with an objective of producing a model with a significant degree of generalizability. A questionnaire was forwarded to interviewees in advance of the interviews to facilitate their preparation. An unstructured interview approach was used during the actual interview sessions. The MIS departments selected represented a convenience sample of eight firms. The individuals interviewed from these firms were project team managers and the supervisors of these team managers. The interview methodology was favored over a survey approach because the dialog between the interviewer and the participants provided data which is richer in detail. The results of Phase 1B are presented in detail in Chapter 4. Based on the analysis of the data obtained in Phase 1, a set of parameters for an MIS department model were selected. The model is summarized in Table 3.1 and described below.

Department Organization and Resource Characteristics. The MIS department model selected incorporates an organization with three fixed design teams. Each team has a total of nine designers divided into three classes or skill levels with three designers in each of the three skill levels. Skill level 1 designers are the least skilled and skill level 3 designers are the most skilled. Special skills such as database or data communications expertise is not modeled. Technological resources are not modeled as they are not generally constraining in MIS project management.

**1) Department Organization.**
* Three Fixed Design Teams.
* Nine Designers/Team.

**2) Resource Characteristics.**
* Three Skill Levels for Designers with Three Designers/Skill Level.
* No Special Skills Recognized.
* Designers Assigned to One Activity at a Time.
* Limited Interchangeability of Skill Levels for Assignment to Activities Allowed.
* Technological Resources Not Modeled.

**3) Project Characteristics.**
* Two Project Types: Maintenance and Development.
* Workload Mix: 50% Workload for Each Project Type. (Use 3 Networks to Represent Maintenance Projects and 2 Networks to Represent Development Projects.)

**4) Activity Characteristics.**
* Multiple Activities per Project.
* Duration - Random Variate Generated from a Gamma Distribution.
* Rework Probability - Random Variate Generated from a Uniform Distribution.
* Rework Duration - Random Variate Generated from a Gamma Distribution.
* Probability of Incurring Rework by Activity Type.
  - Design Activity - 10% Rework.
  - Test/Code Activity - 20% Rework.
  - Implement Activity - 1% Rework.
* Rework Loop Size - Limited.

**5) Managerial Decisions and Policies.**
* No Inter-Team Flexibility of Assignment for Designers.
* Overtime Not Modeled.
* Contract Work Not Modeled.
* Maintenance Projects Preempt Designers from Development Projects.
* Due Dates Set Internally 100% of the Time.
* Due Dates Not Set for Maintenance Projects.

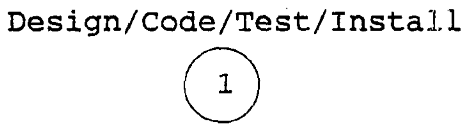
**Table 3.1 MIS Department Model**

A limited interchangeability of designers across skill levels is allowed as follows. If an activity requires a designer of skill level 2, for example, then a designer in either skill level 2 or 3 may be assigned to the activity because skill level 3 is the highest and most skilled of the three classes. The designers assigned are from the lowest appropriate skill levels available at the time the activity becomes eligible for resource allocation. All of these parameters are justified by the results of the Phase 1 interviews.

Project and Activity Characteristics. The stream of projects arriving to the simulator includes a mixture of maintenance and development projects. These two categories of projects were differentiated in Section 1.1.3. The maintenance projects are represented by three maintenance networks (see Figure 3.1). The duration time of activities from these projects is generated stochastically. These project networks are representative of maintenance and repair activities which could take from as short as a few hours to as long as two weeks for completion. These three networks are repetitively used to represent all maintenance activities the MIS department is required to perform.

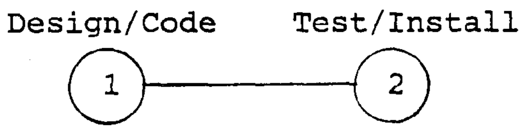
The development projects are represented by two development networks (see Figure 1.1 and 1.2 in Chapter 1). Figure 1.1 is the shortest development network and represents projects with an average duration of approximately two months. Figure 1.2 is a longer development network and

<u>**Maintenance Project Network #1**</u> Resources Required By Level 1/2/3: 0/1/0 Duration Estimate: 1 Day.




(This project is treated as a single activity.)

<u>**Maintenance Project Network #2**</u> Resources Required By Level 1/2/3 per activity: 0/0/1 Duration Estimate: Activity 1: 2 days; Activity 2: 1 day.




(This project is treated as two activities.)

<u>**Maintenance Project Network #3**</u> Activity Resources
Required By Level Duration Number: 1/2/3 & By
Activity: Estimate:

| | | |
|---|---|---|
| 1 | 1/0/0 | 2 days |
| 2 | 0/1/0 | 3 days |
| 3 | 0/0/1 | 4 days |
| 4 | 0/1/0 | 1 day |
| 5 | 1/0/1 | 1 day |

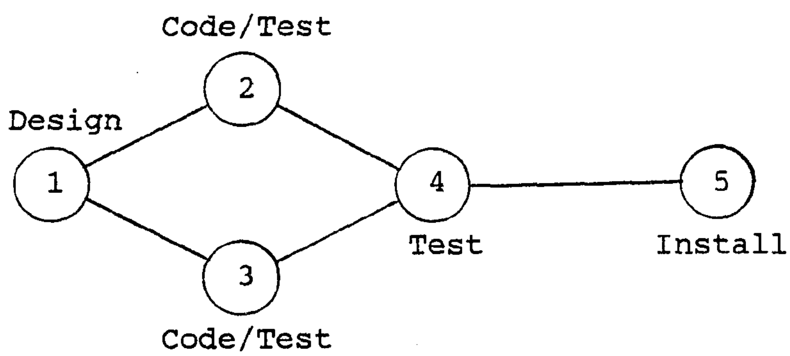



**Figure 3.1 Maintenance Project Networks**

represents projects with an average duration of approximately six months. The activities of these two development project networks also have stochastically generated duration times. These two networks are repetitively used to represent all development activities the MIS department performs.

The order of arrival for maintenance and development projects occurs in a random fashion, though the total workload mixture for resource requirements has been set such that 50% of the total work requirements represents maintenance activities and 50% represents development activities. As such, approximately 10% of the projects arriving to the simulator are development projects and 90% are maintenance projects. Project interarrival times are generated as random variates from a uniform distribution with a mean interarrival rate of one project every 2.25 working days. Activity durations are generated from gamma distributions. The probability of incurring rework requirements or looping back to an earlier point in a project due to design deficiencies is generated from a uniform distribution [0,1] and varies across activities.

These parameters are justified as follows. The random mix of development and maintenance projects in the job stream models an arrival process in which the maintenance requirements are not predictable. This pattern was found in the firms which participated in the model development process of Phase 1. Seasonality is not modeled because firms tend to commence new development projects whenever the designer resources necessary become available, i.e. seasonal patterns are not evident from the data collected during Phase 1. The workload mix is representative of that reported by the firms that were visited. It also represents a midpoint for the wide variance of workload mixes reported in the literature which generally ranges from a low of 20% to a high of 80% for resources committed to maintenance activities. The project networks are used repetitively to simplify controlling the workload mixture.

Sufficient data was not available from the firms visited during Phase 1 for "fitting" theoretical distributions to describe activity durations or rework requirements. There is also insufficient data for defining "empirical" distributions. In such instances, Law and Kelton provide some limited guidance on the selection of an appropriate probability distribution for modeling purposes [LAW82, pgs 155-175]. The Gamma distribution was selected for modeling activity durations because it has been used in applications that describe the time required to complete tasks, i.e. activity durations. The shape and scale parameters of the Gamma distribution may be set to provide a density function with a wide variety of shapes. This enables an approximation of the shape of activity duration distributions described in the literature ([NORD70], [PUTN78], [SPRA86]). Data was not available on the amount of rework incurred in any of the firms visited. The uniform distribution was selected since only subjective estimates on rework were available from the managers interviewed [LAW82].

Managerial Decisions and Policies. One policy an MIS department might utilize would be to temporarily assign designers from one project team to work on projects for another team. As the focus of this research is not the effect of interteam flexibility of resource assignment, this capability is not modeled. Similarly, overtime and subcontracting are not allowed.

Maintenance requirements may preempt ongoing development activities depending upon the policy management has established. The effect of various preemption policies on meeting established due dates is of interest. Therefore, preemption policy was selected as a treatment factor with two levels which are described in Section 3.2.3 below.

Due dates may be established internally by the MIS department, or they may be established externally by some form of corporate steering committee or by higher level management. Dumond did not find a significant difference in effect of various due date setting heuristics when a portion of projects had externally set due dates. The general effect across due date rules resulted in a higher percentage of projects with internally set due dates being completed late [DUMO85]. In this research, due dates for all development projects will be established internally. The policies represented by these parameters can be justified as being generally representative of the interview results described in Chapter 4.

**3.1.2 PHASE 2.** Phase 2 is divided into two subphases; Phase 2A and 2B. In Phase 2A, a discrete event simulation program representing the MIS department model was developed. The program is written in FORTRAN 77 to enhance its portability. The program was validated and the output analyzed to verify its accuracy and ability its to simulate the MIS department model. The program design is discussed in detail in Appendix A.

Phase 2B consists of a three-factor, full-factorial experimental analysis. The first factor has four levels corresponding to four due date setting heuristics used in estimating development project due dates. Chapter 2 suggests that heuristics from the project management and job shop areas offer good potential for achieving accurate due date estimates for MIS projects. The due date setting heuristics selected are discussed in detail in Section 3.2.1.

The second factor has five levels corresponding to five scheduling heuristics used in prioritizing activities for the allocation of resources. In simulating the development of projects within an MIS department, it is necessary to employ techniques which reflect the policy that management follows in regard to the assignment of resources to activities. One facet of this policy concerns which activity will have priority when two or more activities are simultaneously available for resource assignment. Scheduling heuristics are used to determine which activity from the available set will receive resources. Good scheduling heuristics tend to allocate resources to activities using some information about the current state of the development process in the MIS department. For example, an activity on a project critical path might receive priority for resources over an activity from a path possessing slack. The literature reviewed in Chapter 2 suggests several scheduling heuristics which might be used to implement this facet of a managerial policy on resource assignment. The scheduling heuristics are discussed in detail in Section 3.2.2.

Another facet of managerial policy addresses resource allocation across competing types of projects. This facet concerns the preemption of resources from development work in process whenever maintenance work arrives to the MIS department. While the preemption of resources is not a new problem, the characterization of preemption in the MIS multiple project problem is somewhat unique and introduces the third factor to the experiment. The degree to which preemption is allowed reflects the value management places on the ability to respond immediately to maintenance requirements, management's willingness to delay the completion of development projects, and other aspects of MIS departmental performance. The third factor has two levels representing two resource preemption policies which management might follow. The resource preemption heuristics are discussed in detail in Section 3.2.3.

## 3.2 DUE DATE, SCHEDULING, AND RESOURCE PREEMPTION HEURISTICS.

This section describes the three experimental factors examined in the research. Section 3.2.1 describes the four due date heuristics to be examined. Section 3.2.2 describes the five scheduling heuristics which are used to schedule activities for resource allocation. Two resource preemption policies which management might adopt are developed in Section 3.2.3. Section 3.2.4 outlines the procedure used to estimate various parameters, termed "F" and "K" parameters, which are integral components of the due date heuristics examined in the research.

**3.2.1 DUE DATE HEURISTICS.** There are four treatment levels for this experimental factor representing four different due date setting heuristics. This factor is of primary interest in the research. A due date setting heuristic which utilizes more information about the status of projects under development in an MIS department should perform better than one which utilizes less information. The due date setting heuristics tested vary from the naive to the more sophisticated in terms of their use of information. The heuristics examined set the due date for a development project when it arrives to the simulation.

[1] Historical Mean Completion Time (FLOW) is representative of due date estimating practices that rely on

analogy to past project completion performance. Such practices are often employed in MIS departments. The due date is set as the current date plus the historical mean project completion time, "F".

$$DD_i = CLOCK + F_j$$

where $DD_i$ = due date estimate of project i

CLOCK = the present date

$F_j$ = historical mean project completion time using scheduling policy j

"$F_j$" is partially dependent on the policy management follows in scheduling activities for the assignment of resources. For this research, "F" must be estimated for each of the scheduling heuristics tested in the MIS department model. The procedure for estimating "F" is described in Section 3.2.4.

[2] Number of Activities (NUMACT) sets the due date as the current date plus a function of the number of activities in the project. The function is computed by multiplying a parameter "K", which is a constant, times the number of activities in the project. "K" is computed based on historical data. Since historical data is not available for the simulation, the "K" parameter will be estimated for this heuristic and others by running simulations to generate a historical pattern of performance. The procedure for estimating the "K" parameters is discussed in Section 3.2.4.

$DD_i = CLOCK + (K_1 * NUMACT_i)$

where $NUMACT_i$ = number of activities in project i

$K_1$ = a parameter estimated for this research

<u>[3] Critical Path Time</u> (CPTIME) sets the due date as the current date plus a function of the computed project critical path time. Again, the function is computed by multiplying a parameter "K" times the computed critical path length to add sufficient slack to account for the historical constraint imposed by scarce resources.

$DD_i = CLOCK + (K_2 * CPTIME_i)$

where $CPTIME_i$ = critical path time of project i

$K_2$ = a parameter estimated for this research

<u>[4] Scheduled Finish Time</u> (SFT) utilizes more information concerning the state of system development in the MIS department than do the three previous due date assignment rules. This rule finitely schedules new projects into the system as they arrive. The scheduled finish time of the last activity of the project is adjusted by a parameter "K" to again add sufficient slack to account for the dynamic nature of the environment.

$DD_i = CLOCK + (K_3 * SFT\ ESTIMATE_{ij})$

where $SFT\ ESTIMATE_{ij}$ = scheduled finish time of activity j of project i

$K_3$ = a parameter estimated for this research

Clearly the estimation of values for the various "F" and "K" parameters is important. In the real world, managers could always insure the completion of a project prior to the estimated due date simply by selecting large values of "F" or "K". By manipulating resources, the manager could then achieve near zero deviation between the estimated due date and the actual completion date. There exist, however, factors which prevent managers from acting in such a fashion. One factor is the ever present backlog of development projects which form the queue of work awaiting the availability of resources. Secondly, MIS management is typically monitored by a corporate steering committee or strategic planning committee. This committee reviews the progress of projects and often establishes project priorities. Within the simulation, it is critical to select appropriate values for the "F" and "K" parameters to achieve a fair evaluation of heuristic performance. The method of selecting the parameter values is explained in detail in Section 3.2.4.

**3.2.2 SCHEDULING HEURISTICS.** There are five treatment levels for this experimental factor. These heuristics are used to allocate available resources to competing development activities based upon a priority index value. The priority index value is an attribute of each development activity that is ready for resource allocation.

When a development project arrives to the MIS department, the first activity is ready for resource allocation. The priority index value is computed according to the scheduling heuristic in use for a particular simulation run. The first activity is placed in sequence in a "Ready For Resources" queue based on the priority index. Activities move from the "Ready For Resources" queue to a "Work In Process" queue when resources are available and allocated. All other activities of the project are placed in a "Wait" queue. Activities move from the "Wait" queue to the "Ready For Resources" queue when all preceding activities have been completed. It is at this time that the priority index value for the activity is computed. This provides a priority index value which captures the dynamic nature of the systems development environment.

[1] First Come-First Served (FCFS) allocates resources to development activities which have been waiting for resources the longest. This heuristic is often used for a baseline comparison of the performance of other experimental treatments. The priority index is computed by:

$$\text{Index} = \min_{j} (TA_i)$$

where $TA_i$ = time activity i becomes available for resource allocation

j = set of competing activities

[2] Minimum Slack (MINSLK) determines priority by allocating resources to the activity with the minimum slack. Slack is computed as the difference between the critical path late start time and early start time. Each time an activity completes, the slack for each activity in the project is recomputed. The priority index is computed by:

$$\text{Index} = \min_{j} (LST_{ij} - EST_{ij})$$

where $LST_{ij}$ = late start time of activity j of project i

$EST_{ij}$ = early start time of activity j of project i

[3] Minimum Slack as Modified by the Due Date (MINSLK[DD]) is the MINSLK heuristic as modified by Dumond [DUMO85]. This modified heuristic develops the priority index by using the project due date or the current value for late start time, whichever is earlier in the simulation. It then recomputes the slacks. When a project is delayed due to resource preemption or due to the arrival of competing development projects, the slack for an activity may become negative. Projects with activities on the critical path which have negative slack will be late if sufficient extra slack is not provided at the time the due date is set for the project. Note that projects which are late will have negative slack values for activities on the critical path. The activities in these projects with negative slack will receive priority for resource allocation. This should reduce the degree to which the projects are late. Again, each time an activity from a project is completed, the slack for all activities in the project is recomputed. The priority index is computed by:

$$\text{Index} = \min_{j} (\min_{ij} (LST_{ij}, LST[DD]_{ij}) - EST_{ij})$$

where $LST[DD]_{ij}$ = LST of activity j as modified by the due date of project i

[4] Minimum Late Finish Time (MINLFT) is often used in static project scheduling and assigns resources based on the late finish time for activities on the critical path. Resources are allocated to activities with the earliest late finish times. When an activity from a project is completed, the late finish times for all activities in the project is recomputed. The priority index is computed by:

$$\text{Index} = \min_{j} (LFT_{ij})$$

where $LFT_{ij}$ = late finish time of activity j of project i

[5] Minimum Late Finish Time as Modified by the Due Date (MINLFT[DD]) is the MINLFT heuristic as modified by Dumond to take into consideration the due date in allocating resources [DUMO85]. As with the MINSLK[DD] heuristic, the priority index is computed by using the earlier of the project due date or the late finish time of the project's last activity. Similarly, the late finish time as modified by the due date is recomputed for each activity in a project whenever an activity in a project is completed. The priority index is computed by:

$$\text{Index} = \min_{j} (\min_{ij} (LFT_{ij}, LFT[DD]_{ij}))$$

where $LFT[DD]_{ij}$ = LFT of activity j as modified by the project due date

In a multiple resource allocation problem, such as the MIS problem, an activity with a high priority index value may require resources from more than resource class. For example, an activity may require two units of designer type #1 and one unit of designer type #3. If the scheduling heuristic is strictly followed, resources would not be allocated to any other activity until the activity with the highest priority index value has received resources. This could result in significant resource idleness. For example, if two units of designer type #1 and zero units of designer type #3 were available in the above situation, the resources would lie idle. In practice, managers would not allow this to happen.

To prevent excess resource idleness in the simulation, a "look ahead" heuristic is used in conjunction with each scheduling heuristic. Whenever an activity with priority for resources cannot have its requirements satisfied, other activities are examined to determine if the resources available should be allocated to activities with a lower priority index value. To prevent activities with unusual or high resource requirements from never receiving resources, the "look ahead" heuristic is only used when those activities with high priority index values have not waited an excessive time period for resources. In the simulation, if high priority activities had waited in excess of five days, or one work week, the look ahead heuristic was not used.

**3.2.3 RESOURCE PREEMPTION HEURISTICS.** This third experimental factor has two treatment levels. The preemption of resources represents a major difference between this research and that of Dumond. This factor provides an examination of two policies which management might adopt relative to the priority that maintenance projects will receive in the allocation of scarce resources.

[1] Absolute Priority for Resources (APR) as a policy provides maintenance projects complete priority for resource allocation whenever a new maintenance project arrives to the MIS department. This includes priority over development activities which are in process at the time of arrival of the maintenance project. Maintenance projects are allocated resources as soon as they arrive to the system. When adequate resources are not available, resources are preempted from development work in process activities. The algorithm for preemption is to remove resources from development activities which have been in process the least amount of time. Activities can be preempted which have negative slack. If resources are not available through preemption, the first activity of the maintenance project is placed in a maintenance project "Ready For Resources" queue. The activity is assigned a priority index equal to the current simulation CLOCK time. This mechanism sequences competing maintenance activities on a first come-first served basis.

[2] Limited Priority for Resources (LPR) as a policy gives maintenance projects priority over some development activities which are awaiting resource allocation and over some work in process activities. If sufficient free resources exist, the first activity of a maintenance project is allocated resources as soon as the project arrives to the system. If sufficient resources are not available,

resources are preempted from work in process development activities only if the development activity has positive slack, i.e. the project can still meet the established due date. Where sufficient resources are still not available, maintenance activities are placed in a maintenance project "Ready For Resources" queue and are assigned a priority index in the same manner as in the APR heuristic described above. When resources become available, the development project "Ready For Resources" queue is first checked for activities that have negative slack. If a development project activity has negative slack it receives resources. The maintenance project "Ready For Resources" queue is next checked and receives priority for resources over any development project activity which has positive slack.

Other preemption heuristics are possible. These two are selected for examination because they reflect policies discussed by project managers in the Phase 1 interviews. The APR reflects a policy where management considers it extremely important to respond promptly to the maintenance needs of users regardless of whether the maintenance is classified as "emergency" work. The LPR recognizes that response to the maintenance needs of users is important, but limits the priority placed on that work. When projects are already late, temporarily preempting resources from such projects may reflect poorly on the managerial skills of the MIS department managers.

**3.2.4 SELECTION OF "F" AND "K" PARAMETERS.** Within the simulation, it is critical to select appropriate values for the "F" and "K" parameters to achieve a fair evaluation of heuristic performance. In an actual operating environment, these parameters would be estimated from historical data reflecting the MIS department's performance relative to meeting estimated due dates. In this experiment, the simulation program is run on a stream of projects to generate historical performance data. Since the parameters to be estimated are sensitive to various factors, e.g. the scheduling heuristic, resource levels, project arrival rates, etc., each combination of due date, scheduling, and preemption heuristic would represent an MIS department with different policies and different historical performance data. It is therefore necessary to generate a separate parameter value for each of the possible combinations, e.g. 10 different values for "F" and 30 different values for "K".

A good estimate of "F" will minimize the mean deviation of the actual completion date from the estimated due date, termed Mean Deviation of Due Date (MDDD), resulting in an expected value for MDDD equal to zero. This performance measure is calculated as follows.

$$MDDD = \left[ \sum_{i=1}^{N} (DD_i - AC_i) \right] / N$$

where MDDD = mean deviation of due date

$DD_i$ = estimated due date of project i

$AC_i$ = actual completion date of project i

N = number of development projects.

Using this performance measure, the estimation procedure for the "F" parameters for each of the 10 cells is straightforward. For each cell, a set of 20 simulation runs was conducted with the same run length (750 days) as was used in the main experiment. Each run used a different random number seed. The project stream for each of the runs was the same as for the main experiment and included 100 development projects. The use of different random number seeds resulted in varying project interarrival times and varying project activity durations, varying rework requirements, etc. The estimates of "F" resulted in a MDDD less than 0.05days for each cell. This represents a MDDD of less than 30 minutes indicating that the "F" parameter estimates provide a near zero mean deviation between the actual completion date and the estimated due date.

Estimating the "K" parameters is less straightforward. A method for conducting this process is attributable to Ragatz [RAGA85] and Dumond [DUMO85] and consists of a two stage approach. Again the objective is to minimize the MDDD. In the first phase, the method establishes an initial estimate for each "K" using an analytic procedure. For example, a formula for an initial value of "K" for the NUMACT due date heuristic is:

$$K_i = NUMACT / MCT$$

where $K_i$ is a value of "K" for the "ith" scheduling and preemption heuristic combination and can be computed for a given series of projects when the number of activities (NUMACT) of the projects and mean completion time (MCT) are known. These numbers are obtained in the same manner as that used in estimating the "F" parameters. A set of 20 simulation runs was made to obtain the estimates of NUMACT and MCT with the initial values for "K" set to unity so that no effect due to the value of "K" is obtained for the due date heuristic. This same approach is used for the CPTIME and SFT due date heuristics.

In the second phase, the value of "K" is confirmed or adjusted by actually running the simulation. The initial analytic value of "K" assumes that no interaction exists between the "K" value selected and the performance measure of minimizing the MDDD. This is not the case where the scheduling heuristic considers the due date. Therefore, it is necessary to examine other values of "K".

Dumond used a global search around the analytical value of "K". For example, he states that "if the analytical solution indicates a K value of 1.835, then a test of K values of 1.5, 1.7, 1.8, 1.9, 2.0, and 3.0 could be done" [DUMO85 pg 76]. A more efficient procedure is available, although it is CPU time intensive. It requires from 20 minutes to three hours of CPU time on an IBM 4381 computer for 20 simulation runs with run length set at 750 work days.

Recall that each value of "K" will result in a different value for the MDDD. The MDDD is a function of "K" as well as the effects of the other variables and factors that affect the MIS department's performance on this measure, plus random error. It is well known that a value of "K" equal to unity will result in due date estimates which are always short of the actual completion date. Further, values of "K" which are greater than those obtained in the analytical process should result in due date estimates which are generally greater than the actual completion date. A value of "K" which is a root of the function defined by all valid "K's" will provide a near zero MDDD. To solve for the root, the method of successive bisection was used [BYRK81]. Initial end points which guaranteed inclusion of the root were selected. For example, if the analytical value of "K" was 1.8, then appropriate end points might be 1.0 and 2.6. For each value of "K" tested, a series of 20 simulation runs were again conducted. The search was stopped when a value of "K" was confirmed which resulted in a MDDD less than 0.05 days. The values for "F" and "K" used in the simulation experiment are presented in Tables 3.2 and 3.3 respectively.

**Preemption Heuristic = APR**

| | FCFS | MINSLK | MINSLK[DD] | MINLFT | MINLFT[DD] |
|---|---|---|---|---|---|
| FLOW | 82.878 | 106.886 | 94.288 | 108.340 | 95.477 |

**Preemption Heuristic = LPR**

| | FCFS | MINSLK | MINSLK[DD] | MINLFT | MINLFT[DD] |
|---|---|---|---|---|---|
| FLOW | 76.576 | 113.379 | 65.926 | 129.731 | 67.376 |

**Table 3.2 Values of "F"**

---

**Preemption Heuristic = APR**

| | FCFS | MINSLK | MINSLK[DD] | MINLFT | MINLFT[DD] |
|---|---|---|---|---|---|
| NUMACT | 1.961 | 2.379 | 2.452 | 2.629 | 2.753 |
| CPTIME | 2.004 | 2.479 | 2.402 | 2.666 | 2.934 |
| SFT | 1.792 | 2.215 | 2.178 | 2.373 | 2.469 |

**Preemption Heuristic = LPR**

| | FCFS | MINSLK | MINSLK[DD] | MINLFT | MINLFT[DD] |
|---|---|---|---|---|---|
| NUMACT | 1.813 | 2.600 | 1.581 | 3.141 | 1.644 |
| CPTIME | 1.853 | 2.683 | 1.645 | 3.189 | 1.692 |
| SFT | 1.656 | 2.395 | 1.465 | 2.845 | 1.522 |

**Table 3.3 Values of "K"**

## 3.3 EXPERIMENTAL DESIGN.

**3.3.1 PERFORMANCE MEASURES.** Two sets of performance measures are required. One set is used to evaluate the effect of the experimental treatments on development projects and the second set is used to evaluate the effect on maintenance projects.

There are two objectives with which MIS management is primarily concerned. One is the general objective of minimizing the mean completion time required for projects to provide high throughput. This objective applies to both development and maintenance type projects. This same objective is found throughout the project management and job shop literature reviewed in Chapter 2. The second objective is only relevant to development projects. Here, management seeks to meet the due date estimate. There exist at least two reasons for this second objective. One is that higher level management, often in the form of a strategic planning committee, may dictate this as a good managerial practice. Second, the establishment of an estimated due date creates an expectation on the part of users. Failure to meet the estimated date adversely affects the image of the MIS department.

A single performance measure is used for evaluating the first objective given above. This measure is relevant to both development and maintenance projects. It is to minimize the mean project duration or <u>Mean Completion Time</u> (MCT). A separate measure of MCT will be calculated for development and maintenance projects. For each type of project, the MCT is calculated as follows.

$$MCT = \left[ \sum_{i=1}^{N} (TC_i - TA_i) \right] / N$$

where $TC_i$ = Time of Completion of Project i

$TA_i$ = Time of Arrival of Project i

N = Number of Projects

MIS departments evaluate the objective of meeting the due date estimate using various criteria. Some departments treat excessive earliness as well as lateness of actual project completions as a poor management practice. This managerial philosophy emphasizes exactness in estimating. The performance measure used to evaluate this philosophy is the Mean Absolute Deviation of Completion Time (MADCT). This measure also captures the degree to which the heuristics result in a wide variance of both early and late completions of projects. The MADCT is calculated as follows.

$$MADCT = \left[ \sum_{i=1}^{N} \left| (TC_i - DD_i) \right| \right] / N$$

where $TC_i$ = Time of Completion of Development Project i

$DD_i$ = Due Date Estimate of Development Project i

N = Number of Development Projects

Other departments would penalize lateness more heavily than earliness. This policy recognizes that users may complain about late project completions, but probably will not complain about early completions [SHAN85]. It still provides some penalty for excessive earliness, thereby capturing, to some degree, the managerial concern for exactness in estimating. The performance measure used to evaluate this philosophy is termed Mean Weighted Lateness (MWT). This measure weights the penalty for late completions of projects twice as much as early completions. The MWT is calculated as follows.

$$MWT = \left[ \sum_{i=1}^{N} X_i \right] / N$$

where $X_i = (TC_i - DD_i)$ if the development project is late

$= 0.5 * (DD_i - TC_i)$ if the project is early

**3.3.2 ANALYSIS OF RESULTS.** A three-way analysis of variance is used to analyze the performance measures. Analysis of variance requires certain assumptions concerning the data. These are: (1) the data are from normally distributed population(s), (2) homoscedascity of variances, (3) additivity (required for the F-test), (4) interval scale of measurement, and (5) independent samples [HUCK74]. Of these, assumption #2 may be violated in the experiment. Assumption #2 is violated if the cell error term variances are heteroscedastic. It has been argued that the simulation methodology is sufficiently controlled and analysis of variance procedures are sufficiently robust to allow minor violations of the second assumption [HUCK74].

Significant main effects will be investigated using Scheffe's multiple comparison procedure to determine where differences lie within main effects. Of more interest will be the possible interaction effects between the due date and scheduling heuristics. If a significant interaction effect exists, an investigation of "simple" effects will be made using a procedure analogous to one-way analysis of variance [KEPP82].

**3.3.3 SAMPLE SIZE ESTIMATION.** It was necessary to initially estimate the number of observations required in each cell of the experiment. A research design similar to that used in this research was used by Patterson in a static single project experiment [PATT73]. The results of his analysis of variance were used to provide an estimate of the error variance and interaction sum of squares required for a power computation. The computation was made for the three-way interaction because this provides the most conservative estimate of the sample size required [KEPP82]. With an alpha setting of .05, it was found that a power greater than .88 would be provided by a cell sample size of 15 observations resulting in 600 observations or runs for the entire experiment.

After the simulation program was written and validated, it was found that the CPU time required for an individual observation was not excessive (approximately 45 to 75 seconds). Each observation is for 750 days of operation for the MIS department. Twenty observations per cell were made, thereby raising the power of the F-test to be greater than .90 and requiring 800 observations in total.

**3.3.4 EXPECTED EXPERIMENTAL RESULTS.** With a three-factor, full-factorial experiment, there are several potential results. These are summarized in Table 3.4 which presents the experimental design. One potential result would be a significant three-way interaction between the due date, scheduling, and resource preemption heuristics. It could be argued that there is the potential for this interaction because the Limited Priority for Resources preemption heuristic recognizes when development projects are late and doesn't allow resource switching in these cases. It is not expected that this interaction will be significant. It is also not expected that the two-way interactions between the resource preemption and due date heuristics or between the due date and scheduling heuristics will be significant. Therefore, the effect of the due date heuristics may be analyzed independently of the other two factors as a main effect in the experiment. It is expected that the Limited Priority for Resources heuristic with the policy of not switching resources for activities with negative slack will result in maintenance projects being delayed. This will result in a larger mean duration for maintenance projects than with the Absolute Priority for Resources heuristic. This should also produce a corresponding smaller mean duration for development projects.

**Main Effects:**
- Due Date Heuristic (A) (4 Levels)
- Scheduling Heuristic (B) (5 Levels)
- Resource Preemption Heuristic (C) (2 Levels)

**Possible Interaction Effects:**
- A x B
- A x C
- B x C
- A x B x C

**Due Date Heuristics:**
- Mean Flow Due Date Rule (FLOW)
- Number of Activities Due Date Rule (NUMACT)
- Critical Path Time Due Date Rule (CPTIME)
- Scheduled Finish Time Due Date Rule (SFT)

**Scheduling Heuristics:**
- First Come, First Served (FCFS)
- Minimum Slack (MINSLK)
- Minimum Slack Modified by the Due Date (MINSLK[DD])
- Minimum Late Finish Time (MINLFT)
- Minimum Late Finish Time Modified by the Due Date (MINLFT[DD])

**Resource Preemption Heuristics:**
- Absolute Priority for Resources (APR)
- Limited Priority for Resources (LPR)

Number of Cells: 40
Number of Observations per Cell: 20

**Table 3.4 Experimental Design**

---

It is expected that the SFT due date heuristic will perform better than the other three due date heuristics as it uses more information concerning the workload and resource state of the MIS department in setting the due date. Although it is expected that the due date heuristic factor will not significantly interact with either of the other two factors, such an interaction is possible. The

SFT due date heuristic uses critical path computations in generating the due date. This produces the potential for interaction between this heuristic and four of the scheduling heuristics.

This chapter has presented the methodology used in this research. This included a description of the experimental treatments and the methods to be used in measuring and evaluating the performance of these treatments. The next two chapters will present the research results. Chapter 4 presents the results of the Phase1 MIS department interviews. Chapter 5 presents the results of the Phase 2 simulation experiment.

## 4.0 ANALYSIS OF PHASE 1 INTERVIEWS.

In Phase 1, interviews were conducted with a convenience sample of MIS professionals in eight different MIS departments. These interviews were conducted in order to confirm and/or modify a preliminary computer based systems development model. Recall from Chapter 3 that the research methodology selected as most appropriate for the examination of project due date heuristics is computer simulation. The simulation research requires the specification of a MIS department model which is generally representative of a large class of MIS departments. Early in this research, a literature review revealed that a current, comprehensive description of such a model was not available. This necessitated the development of an adequate model. This was accomplished during Phase 1 by reviewing literature relevant to the process used in designing and implementing MIS systems. The main objective of the Phase 1 interviews was the confirmation of the MIS department model.

This chapter presents the results of the interviews through a discussion of the MIS department model. This model is organized into five areas. Each area represents a set of factors or variables that impact on the manner in which firms develop new computer based system projects and accomplish the maintenance of existing systems.

The individuals interviewed include project team managers and the supervisors of these managers. Table 4.1 gives a summary of the types of MIS departments that participated in the survey. The firm number assigned in Table 4.1 is used as a reference in Tables 4.2 through 4.6 in this chapter. The participating MIS departments are in firms representing various private sector industries and public sector agencies.

| Type of Company or Agency | Firm Number |
| --- | --- |
| Large Regional Bank | 1 |
| Electronics Manufacturing Corp. | 2 |
| Pharmaceutical Manufacturing Corp. | 3 |
| U.S. Army Data Processing Center | 4 |
| U.S. Navy Data Processing Center | 5 |
| Electrical Power Company | 6 |
| Heavy Equipment Manufacturing Corp. | 7 |
| University Administrative Computing Center | 8 |

**Table 4.1 Companies & Agencies Participating in Phase 1**

## 4.1 DEPARTMENT ORGANIZATION.

Table 4.2 gives the factors and variables that are included in the Department Organization Area of the MIS department model. The number of system designers in the MIS departments range from a low of 23 to a high of 175.

| Factor or Variable | Firm Number 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Number of System Designers in a MIS Dept | 70 | 70 | 40 | 175 | 23 | 30 | 50 | 36 |
| MIS Dept Organization (1) | T | T | T | T | T | T/P | T/P | T |
| Team Alignment (2) | A | F | F | F | F | F | F | F |
| Number of Design Teams (3) | 2/Sec | 2/Div | 4 | * | 4 | 4 | 3 | 4 |
| Number of Designers/Team | 6-9 | 5 | 8-15 | 7-10 | 6 | 6-8 | 10-20 | 9 |

(1) T=Fixed Team Organization; P=Pool of Resources.
(2) A=Aligned according to applications systems; F=Aligned with business functional areas.
(3) *=Five Divisions within MIS aligned with functional systems, two Areas per Division, two teams per Area.

**Table 4.2 Department Organization**

Six of the MIS departments are organized with fixed design teams or sections. Two of the departments use a combined fixed team and resource pool organization. In this approach, team managers and analysts are assigned to fixed teams with programmers forming a pool of resources for allocation to design teams. Each design team is normally responsible for the concurrent development of several projects.

In seven departments surveyed, the design teams are aligned with functional business areas, e.g. Accounting, Marketing, Manufacturing Production, etc. The other MIS department is in a large metropolitan bank and aligns teams according to application systems which may cross functional areas. The purpose of this alignment, in either case, is to promote increased designer-user communications and productivity in the development process.

The number of design teams per section or division within the MIS departments varied from two to four. The number of designers per team engaged in development or maintenance activities varied from six to 20. This reflects a span of control greater than that generally expected for managers. In instances where larger numbers of designers were assigned to teams, the team managers stated that they appointed project leaders to assist in managing individual job assignments.

## 4.2 RESOURCE CHARACTERISTICS.

The factors and variables included in the Resource Characteristics Area are given in Table 4.3. MIS designer resources are heterogeneous. Managers classify designers into skill levels to facilitate maximizing the utilization of departmental designer skills available. The corporate MIS departments had established three to five skill levels for designers. These skill levels recognize experience and ability. Project team managers reported using these skill ratings in conjunction with personal knowledge of individual abilities when assigning designer resources to project activities. The government MIS departments use the government civilian employee rating system which recognizes many skill levels for employees based on experience and education. The government skill level system was not reported as being useful for assigning designers to project activities.

| Factor or Variable | Firm Number 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Number of (1) Designer Skill Levels | 3 | 2-P 2-A | 3 | GS | GS | 2-P 3-A | 4 | 4 |
| Specialty Skills Used | No | No | No | Some | Some | No | Yes | No |
| Interchangeability of Designers | Limited-----------------------------> | | | | | | | |
| Effect of (2) Technological Resources | L | S | L | L | L | L | L | L |

(1) P=Number of programmer skill levels; A=Number of analyst skill levels; GS=Government civilian employee skill level system.
(2) L=Little delaying effect from technological resources; S=Some delaying effect requires allocating technology.

**Table 4.3 Resource Characteristics**

In general, special skill designations, such as telecommunications programmer or database designer, are not used to allocate designer resources to project activities. One MIS department recognizes a separate skill level classification for specialists and uses this classification as part of a career progression track.

Each department recognizes that there is limited interchangeability of designers across skill levels. No consensus was reached on how this interchangeability should be modeled. One project manager even argued that designers with higher skill ratings might take longer to complete simple activities than employees with a lower skill level rating. This could occur if a substantial amount of time had elapsed since the designer last completed a similar activity with a resulting degradation of skills. It was generally agreed that designers with higher skill ratings could be assigned to a greater variety of activities. Therefore, higher skilled designers can work on simpler tasks, but the reverse is not generally true.

Only one MIS department regularly tracks technological resource requirements for projects. The other departments do not usually concern themselves with technology as a resource to be allocated to projects.

## 4.3 PROJECT CHARACTERISTICS.

The Projects Characteristics Area includes the factors and variables described in Table 4.4. Each MIS department recognizes two classes of projects, new development and maintenance. The workload mix between development and maintenance varies from a low of 15% to a high of 80% for maintenance. These are subjective estimates provided by the interviewees based upon significantly different definitions of "maintenance". A manager from the firm reporting 15-20% maintenance requirements defines maintenance as follows:

| Factor or Variable | Firm Number 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Types of Projects | Development & Maintenance ---------> | | | | | | | |
| Percent (1) Maintenance | 80 | 40-70 | 50-80 | N | 80 | N | 15-20 | 30-40 |
| Priority for (2) Development | M | S | S | S | S | M | S | S |
| Uncertainty of (3) Requirements | S | S | S | E | S | S | S | S |
| Number of Activities per Project | Small for Maintenance -------------><br>Various sizes for Development -----> | | | | | | | |
| Project Network Shape | Linear with milestones ------------> | | | | | | | |

(1) N=No estimate could be made for the percentage of the workload which was maintenance.
(2) M=Priority for New Development set by MIS; S=Priority set by steering committee or strategic planning committee.
(3) S=Some uncertainty on project requirements, but most projects are well defined; E=Excessive uncertainty and resulting redesign work.

**Table 4.4 Project Characteristics**

"If the system ABENDS (aborts abnormally), or is found or suspected to not perform in the manner for which it was designed, then this is maintenance." At the other extreme, a manager reporting 80% maintenance requirements defines maintenance as including all work except "the automation of a system which was previously manually operated or the development of a system where none existed previously." Other managers define maintenance as work which requires a short time period for completion (less than two weeks was mentioned as a time cutoff by several managers). This latter definition includes small enhancements to existing systems as well as emergency requests to repair systems which are not operational. This is the definition used in this research.

Managers in each MIS department reported that their firm has a formal procedure for proposing the development of new systems. The majority of the proposals come from functional users. MIS designers also propose a small percentage of development requests. Since development proposals compete for resources, a prioritization system is required which will sequence the projects for development. In two MIS departments, the MIS director sets project priorities. One of these managers reported that his decisions were reviewed by a corporate steering committee. Managers in the other six MIS departments reported that priorities were established by corporate steering or strategic planning committees. Managers from all of the departments reported a backlog of projects varying from six months to two years in size.

Uncertainty in systems development includes a lack of knowledge concerning the total new development and maintenance workload that will be required of the MIS department. It also includes a lack of knowledge as to individual project requirements. In six of the firms, a project priority list is established by a corporate steering or strategic planning committee. In the other two firms, the MIS director either establishes the project priority list solely, or else in conjunction with a steering committee. The projects to be undertaken by the MIS department are known in advance with some degree of certainty. This degree of certainty varies depending on the number of changes which occur in the priority list. For instance, governmental regulations may dictate that a new system, not previously scheduled for development, be implemented. The steering committee may also add projects to the list or change the priorities of projects during periodic review sessions.

Each MIS Department reported that projects exhibit some degree of uncertainty as to the actual features or requirements that a user desires. The degree of uncertainty is a function of how well the project has been defined during the initial detailed investigation of user requirements. Poorly defined projects will exhibit a higher variance in the accuracy of the due date. They will also exhibit rework requirements. Seven of the MIS departments reported that the degree of uncertainty was small. This resulted from their similar objectives of "turning out first class projects" by stressing the importance of a well-defined logical design. One MIS department reported excessive uncertainty. This government agency is developing systems for world-wide application. Users are indirectly represented by individuals assigned duties in a "Functional Coordinating Group" (FCG). The FCG operates in a liaison role between the actual users and system designers. The result is a requirements analysis which is often inaccurate.

The number of activities per maintenance project was reported to be small. Maintenance projects are often managed as a single activity for the assignment of designers. Maintenance projects with several activities may be represented by a network which is serial in shape. These projects are generally completed within two weeks, with the average duration reported as three to five days. This classification of maintenance projects includes enhancements to existing systems.

New development projects may vary from two weeks to several years in duration, although most managers stated that it is unusual for a project to extend beyond one year. Managers try to limit projects to a year or less because projects taking longer than a year to complete often result in the delivery of systems which have already reached a state of obsolescence. The number of activities varies with the project size. Seven MIS Departments stated that typical development projects take two to six months for completion.

Each firm recognizes that new development projects consist of definable activities and can be depicted as project networks. Development project networks were described as exhibiting a linear "shape". This "shape" is depicted in Figures 1.1 and 1.2 of Chapter 1. A consensus on the "shape" of development projects was achieved by having managers construct typical project networks graphs.

Milestone activities exist within development projects and represent bottlenecks. Most project managers agreed that project networks have more activities in the middle segments of the project network than in the segments forming either end of the network. One manager stated that, as a result of their development methodology, each phase of a project is distinctly separate with only a single activity connecting the segments representing the phases.

## 4.4 ACTIVITY CHARACTERISTICS.

The factors and variables included in the Activity Characteristics Area are given in Table 4.5. Managers in all of the MIS departments stated that the types of activities comprising a project depend on the methodology used in the development process. A life cycle methodology is generally used. This makes it possible to classify the activities into several categories. The categories described include: (1) planning and designing, (2) coding and testing, and (3) implementing.

The number of designer resources required by individual activities was reported to be small, usually a single designer to an activity. Activities such as structured walk-throughs of system designs or complex program coding may require more than one designer. These designer resources are often from different resource classes because managers tend to assign senior personnel to work with junior personnel on difficult activities. One firm reported that small development projects (approximately 2-3 weeks in duration) are often assigned to a single developer. Seven firms stated that maintenance projects are often assigned to individual developers.

| Factor or Variable | Firm Number 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Type of Activities | Design, Coding & Testing, Implementing -> | | | | | | | |
| Number of Resources Required | Small Number of Designer Resources/Activity | | | | | | | |
| Activity Duration | Variable, but Definable ----------------> | | | | | | | |
| Rework (1) Probability | L | L | N | H | L | L | N | L |
| Size of Rework Loops | Limited ---------------------------------> | | | | | | | |

(1) L=Low probability if good design; H=High probability; N=No estimate possible.

**Table 4.5 Activity Characteristics**

All eight MIS departments develop activity duration estimates by relying on experience or analogy to a similar project or activity. Five of the MIS departments also utilize software packages, such as PAC Micro, Super Project Plus, or SPECTRUM, for estimating. These packages include the ability to produce project network diagrams, examine the potential effect of resource balancing, produce GANT charts, etc. Three MIS departments use a manual tracking system to identify and manage the activities within projects.

Managers from five departments stated that it is common for activities to require some rework. The amount and duration of rework was generally estimated to be small. Several managers stated that rework loops may often involve only a single activity. It was the opinion of most of the MIS managers that loops representing the redesign and rework of entire phases of projects are rarely encountered. Large rework requirements represent poor designs or an inability to properly determine user project requirements. The consensus was that poor managerial practices resulting in large rework requirements would not be tolerated by higher management. One project manager provided subjective estimates of the probability of incurring rework based on the type of activity. These probabilities were: Design Activities - 10%, Code & Test Activities - 20%, and Implementing Activities - 1%. None of the MIS departments maintained data on the quantity of rework encountered.

## 4.5 MANAGERIAL DECISIONS AND POLICIES.

Table 4.6 describes the factors and variables included in the Managerial Decisions and Policies area. Most of the MIS departments maintain a policy that limits the interteam flexibility in the assignment of designers. This means that it is uncommon to assign designers from one team to another team to assist in project development activities. Two of the MIS departments do not allow any interteam flexibility of assignment. This policy reflects the fact that the teams are aligned with specific functional areas within the firm.

The managers interviewed stated that this policy promotes productivity because designers become very familiar with the type of systems being supported and maintained.

| Factor or Variable | Firm Number 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Interteam (1) Flexibility of Assignment | L | L | N | N | L | L | L | L |
| Preemption by (2) Maintenance | Y | Y | Y | Y | L | L | N | Y |
| Overtime (3) | N | L | L | L | N | Y | L | N |
| Contract (4) | N | L | N | N | L | N | N | L |
| Method of (5) Setting Due Date | M | I | I | E | M | M | I | N |

(1) L=Limited assignment of designers across teams; N=No assignment of designers across teams.
(2) Y=Maintenance preempts Development work; L=Maintenance does not preempt critical Development work; N=No preemption due to separate maintenance team.
(3) L=Limited overtime; N=No overtime; Y=Overtime used extensively.
(4) L=Limited contract or vendor system development; N=No contractors used.
(5) I=Due dates set internally by MIS; E=Due dates set externally to MIS; M=Mixture of internal and external due dates; N=No due date established.

**Table 4.6 Managerial Decisions and Policies**

Managers in seven of the MIS departments stated that the inflow of maintenance projects is unpredictable, but that it is important to respond to maintenance requirements quickly. In some instances, quick response is necessary because a critical system may be nonfunctional. At other times, it is important to respond quickly because it affects the corporate image of the MIS department. As a result, designers may be required to stop work on assigned development project activities to respond to maintenance requirements.

The preemption of resources may result in delaying projects beyond the originally scheduled due date. Historically, managers have attempted to prevent delays by scheduling overtime or by shifting resources to the late project. Managers in seven of the MIS departments stated that the use of overtime or resource shifting often causes additional problems. Three of the departments do not use overtime. Four other departments resist scheduling overtime because the designers are salaried professionals. As such, they do not receive overtime compensation. It was generally agreed that requiring designers to work a significant amount of overtime may result in an excessive turnover rate for designer personnel. Managers from one MIS department, however, claimed that "overtime was a way of life". They also reported a high turnover rate among junior designers.

Managers also reported that it is not productive to reallocate resources to late projects. For obvious reasons, only a certain number of designers can productively work on a coding/test activity. The presence of additional designers will not shorten the activity duration and may even increase the duration. The general policy is to avoid resource shifting for new development projects. It is better to rely on increased planning during the design stage of a project.

Three of the firms use contractors to meet some of their system development requirements. In some cases, contractors are also used to develop systems that are high on the priority list, or to implement turnkey systems which are purchased or leased. There were no reported joint design efforts involving both contractors and MIS department designer personnel. Resource requirements are usually limited to the assignment of a designer to perform liaison duties with contractors.

Development project due dates are established internally by the MIS director in three of the MIS departments. One manager reported that most due dates are set externally, while managers in three MIS departments reported a mixture of internally and externally set due dates. These latter three departments reported from 10 to 30 percent of the due dates to be established externally. Managers in one MIS department do not provide users with project due dates. The project managers in this department do establish due dates for internal use only. These due dates are used as targets for work completion by designers. Due dates are not established for maintenance projects since these require completion in the minimum amount of time possible.

**4.6 CONCLUSIONS.**

It is interesting to note that considerable consistency and a lack of variance exists for most factors or variables across the MIS departments examined. This lack of variance exists despite the fact that the eight MIS departments are representative of a wide variety of private businesses and public sector agencies. To some degree, this consistency can be attributed to the fact that only managers from MIS departments relying extensively on the use of a life cycle development methodology were interviewed. As a result of this lack of variance, interviews in additional MIS departments were not felt necessary. The consistent interview results made it possible to quickly achieve a satisfactory model of MIS systems development using life cycle development methodologies. This led to the selection of a general MIS department model which is used in the simulation research conducted in Phase 2.

Although unrelated to the MIS department model, another interesting observation concerns the use of information available on past performance in systems development when estimating the duration of activities for systems currently under development. While all of the firms maintain data on past project performance, most of the firms do not have a databank of past performance information stored in a computer accessible medium. None of the firms have a databank available for access by a database querying system, although managers in each MIS department stated that such a system would greatly assist in the due date estimating process. This appears to be another case of the "shoemaker's children doing without."

## 5.0 ANALYSIS OF EXPERIMENTAL RESULTS.

This chapter reports the results of the simulation experiment. The statistical procedure utilized to interpret the results is analysis of variance. The computer program used to conduct the analysis is the SPSSx Information Analysis System [SPSS86]. Section 5.1 discusses the assumptions required for the use of analysis of variance procedures and focuses on the issues surrounding the assumption of homoscedascity of cell variances. Section 5.2 presents the analysis of the results for the development projects. The measures used to evaluate performance for development projects include the Mean Absolute Deviation of Completion Time (MADCT), Mean Weighted Lateness (MWT), and Mean Completion Time (MCT). Since the MCT measure is also used to evaluate maintenance project performance, the term Mean Development Completion Time (MDCT) is used to differentiate the MCT for development projects. Section 5.3 presents the analysis of the results for maintenance projects. The MCT is the sole measure used to evaluate performance for maintenance projects. The term Mean Maintenance Completion Time (MMCT) is used to differentiate the MCT for maintenance projects.

## 5.1 ASSUMPTIONS REQUIRED BY THE USE OF ANALYSIS OF VARIANCE.

A three-way analysis of variance is used to analyze the effects of the treatments on the performance measures described. Chapter 3 presented a brief discussion of the assumptions which underlie the use of analysis of variance procedures. These are: (1) the data are from a normally distributed population(s), (2) homoscedascity of variances, (3) additivity (required for the F-test), (4) interval scale of measurement, and (5) independent samples [HUCK74]. This section limits its discussion to the issue of homoscedascity of variances since all other assumptions are satisfied.

Two tests for homogeneous variances, Cochran's C-Test and the Bartlett-Box F-Test, were conducted. The null hypothesis is that the cell variances are all equal. The alternate hypothesis is that at least one cell variance is different from the others. The results for these two tests are presented in Table 5.1.

| | | |
|---|---|---|
| Cochran's C = | .086 | P = .000 |
| Bartlett-Box F = | 19.241 | P = .000 |

n = 20/cell

**Table 5.1 Tests of Homogeneity of Variances**

These tests result in a rejection of the null hypothesis of equal variances and indicate that the assumption of homoscedascity is violated. Given these results, a pertinent question is: What are the ramifications of this violation and is it appropriate to use analysis of variance procedures?

In answering this question, consider the effect that heteroscedascity has on the F-distribution. When the variances between cells differ significantly, the sampling distributions of the F-ratio have more mass in the upper tail regions than indicated by the F-statistic. The effect is that the value selected as the cutoff for the critical region is larger than it should be [MILL86]. For example, a researcher may set a critical value for alpha = .05, when in fact, the computed F-value may identify a point such that alpha actually equals .06 or .07. The degree to which heteroscedascity affects the F-distribution is minimized for balanced, fixed effects experiments such as this one [MILL86]. Further, the effect is not large even when the assumption is grossly violated ([BOX54], [KEPP82, pg 86-87], [MILL86, Sec 3.3], [MONT84, pg 91], [SCHE59, Sec 10.3]). Several Monte Carlo experiments have demonstrated the robustness of the F-test to this violation ([CHUR76], [MART77], [ROGA77]). In one experiment examining the effect of extreme heteroscedascity, the authors reported that when an alpha value was selected to be .05, the actual probability of a type I error was between .07 and .09 [ROGA77].

One approach to correcting for heteroscedascity is to mathematically transform the data in an attempt to reduce the cell variances. A problem with this approach is that a transformation of the data may destroy an additive model and create interactions where there are none [MILL86, pg 141]. In this experiment, the possible interactions between factors are of the most interest, therefore the use of a transformation technique is ruled out. Alternatively, realizing that the alpha value will be larger than expected, a lower alpha value may be selected to denote the critical region [KEPP82]. Accordingly, the analysis of variance procedure is used and an alpha value of .001 is selected.

## 5.2 ANALYSIS OF DEVELOPMENT PROJECT PERFORMANCE MEASURES.

This section analyzes the results obtained for the development project performance measures. The formulas used for these measures are presented in Table 5.2.

Recall that there are three factors, or experimental treatments, under investigation in the research. These consist of due date rules, scheduling rules, and resource preemption rules. The three factors are labeled as indicated in Table 5.3 for ease of reference. Table 5.3 also provides numbers which are used to reference the levels within each factor. This system of labeling is used in tables and figures throughout this chapter and the chapter which follows.

$$MADCT = \left[ \sum_{i=1}^{N} \left| (TC_i - DD_i) \right| \right] / N$$

where $TC_i$ = Time of Completion of Development Project i
$DD_i$ = Due Date Estimate of Development Project i
$N$ = Number of Projects

$$MWT = \left[ \sum_{i=1}^{N} X_i \right] / N$$

where $X_i = (TC_i - DD_i)$ if the development project is late
$= 0.5 * (DD_i - TC_i)$ if the project is early

$$MCT = \left[ \sum_{i=1}^{N} (TC_i - TA_i) \right] / N$$

where $TC_i$ = Time of Completion of Project i
$TA_i$ = Time of Arrival of Project i

**Table 5.2 Performance Measure Formulas.**

| Label | Experimental Treatment |
| --- | --- |
| DDATE | Due Date Rules. |
| | 1 FLOW - Historical Flow Time of Projects. |
| | 2 NUMACT - Number of Activities in a Project. |
| | 3 CPTIME - Project Critical Path Time. |
| | 4 SFT - Scheduled Finish Time of a Project. |
| SCHED | Scheduling Rules. |
| | 1 FCFS - First Come, First Served. |
| | 2 MINSLK - Minimum Slack. |
| | 3 MINSLK[DD] - MINSLK Modified by the Due Date. |
| | 4 MINLFT - Minimum Late Finish Time. |
| | 5 MINLFT[DD] - MINLFT Modified by the Due Date. |
| RESPRE | Resource Preemption Rules. |
| | 1 APR - Absolute Priority for Resources. |
| | 2 LPR - Limited Priority for Resources. |

**Table 5.3 Factor Labels.**

**5.2.1 MEAN ABSOLUTE DEVIATION OF COMPLETION TIME.** The first development performance measure discussed is the Mean Absolute Deviation of Completion Time (MADCT). The MADCT weights the penalty for early or late completions of development projects equally. Table 5.4 presents the analysis of variance table for this measure. The table indicates the lack of a significant three-way interaction. Of the three possible two-way interactions, both the DDATExSCHED and SCHEDxRESPRE are significant in their statistical relationship to the MADCT. Further, each of the three main effects are significant.

The lack of a significant three-way interaction means that the results can be analyzed in terms of two-dimensional designs, in this case the DDATExSCHED and SCHEDxRESPRE

| Source | df | SS | MS | F | Signif. of F |
|---|---|---|---|---|---|
| **Main Effects** | | | | | |
| DDATE | 3 | 7,977 | 2,659 | 30.75 | .000 |
| SCHED | 4 | 256,834 | 64,209 | 742.57 | .000 |
| RESPRE | 1 | 12,511 | 12,511 | 144.69 | .000 |
| **2-Way Interaction** | | | | | |
| DDATExSCHED | 12 | 2,856 | 238 | 2.75 | .000 |
| DDATExRESPRE | 3 | 461 | 154 | 1.78 | .150 |
| SCHEDxRESPRE | 4 | 91,517 | 22,879 | 264.60 | .000 |
| **3-Way Interaction** | | | | | |
| | 12 | 1,380 | 115 | 1.33 | .195 |
| **Explained** | 39 | 373,639 | 9,580 | 110.80 | .000 |
| **Residual** | 760 | 65,715 | 86 | | |
| **Total** | 799 | 439,354 | | | |

**Table 5.4 Analysis of Variance for Mean Absolute Deviation of Development Time**

interactions [KEPP82]. A reasonable first step in analyzing the DDATExSCHED interaction is to plot the cell means of the two factors. Figure 5.1a presents a graph of the MADCT for the individual scheduling rule cell means plotted as a function of the due date rules. Figure 5.1b presents a transpose plot. An F-statistic of 2.75 for the DDATExSCHED interaction indicates a rather weak interaction considering the large sample size used in the research and the fact that the statistical power of the F-test exceeds 90%. This weak interaction is clear in Figure 5.1a. Further, it is apparent that this is a strictly <u>Ordinal</u> interaction because the relative performance of the scheduling rules do not result in intersecting lines that connect the cell means.

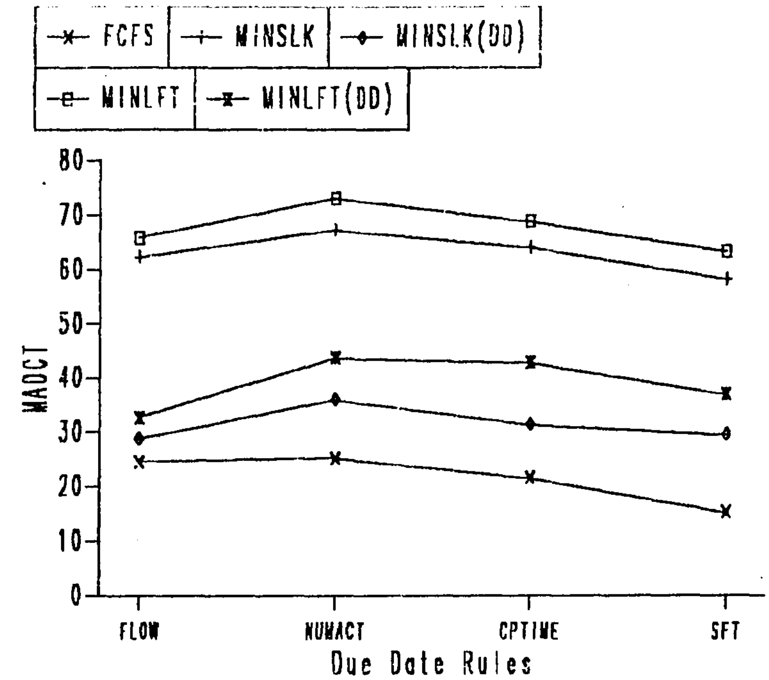



Fig 5.1a MADCT as a Function of
Due Date Rules and Scheduling Rules

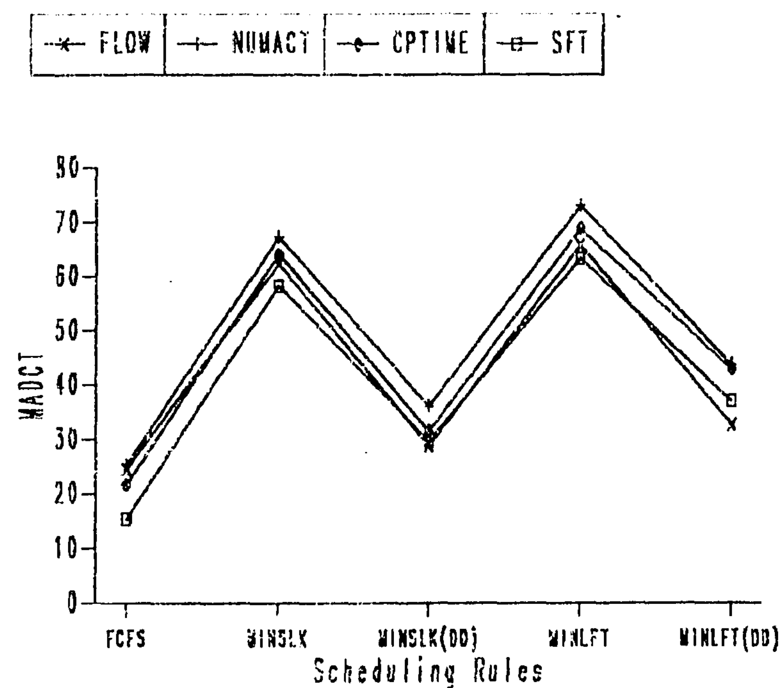



Fig 5.1b MADCT as a Function of
Scheduling Rules and Due Date Rules

It is unnecessary to perform further statistical tests to examine the DDATExSCHED interaction. Recall that one of the research objectives is to determine whether a particular combination of rules dominates all other combinations. With the ordinal interaction, it is easy to determine that the combination of the SFT due date rule and FCFS scheduling rule results in the lowest MADCT. Further, across all due date rules, the FCFS scheduling rule is dominant.

Figure 5.1b presents a picture which is less clear. A single due date rule does not dominate across all scheduling rules. However, since it is the interaction effect that is of interest, the fact that a single due date rule fails to dominate is not extremely distressing.

Figures 5.2a and 5.2b present graphs of the MADCT cell means for the SCHEDxRESPRE interaction. It is not clear from the graphs which combination(s) of scheduling and resource preemption rules dominate. The Minimum Slack as modified by the project due date (MINSLK[DD]) within the Limited Priority for Resources (LPR) rule provides the lowest value of MADCT at 17.77. However, the First Come-First Served (FCFS) and Minimum Late Finish Time as modified by the project due date (MINLFT[DD]) rules within with the LPR rule also provide low values of MADCT at 20.09 and 20.81, respectively. Further, note that the FCFS provides low values of MADCT across both the APR and LPR resource preemption rules demonstrating good consistency of performance.

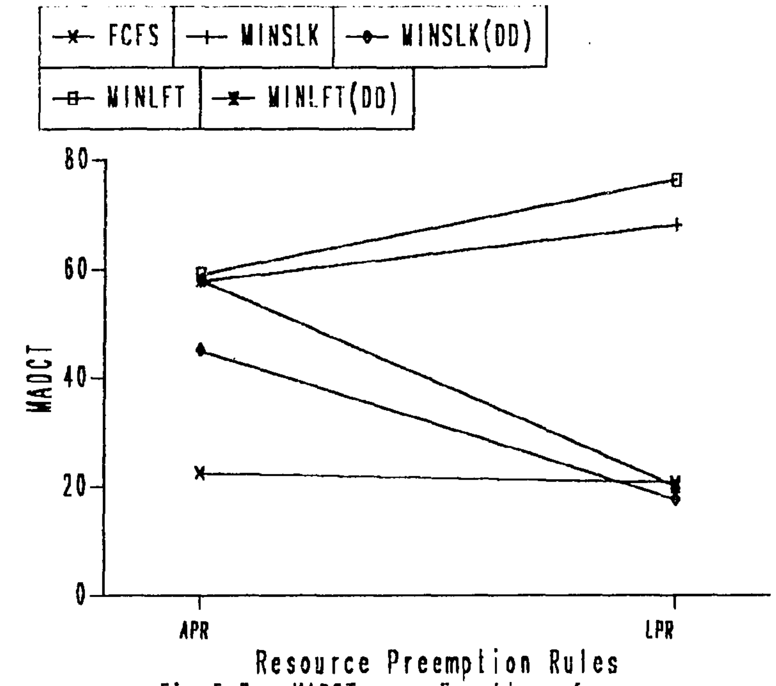



Fig 5.2a MADCT as a Function of
Resource Preemption and Scheduling Rules

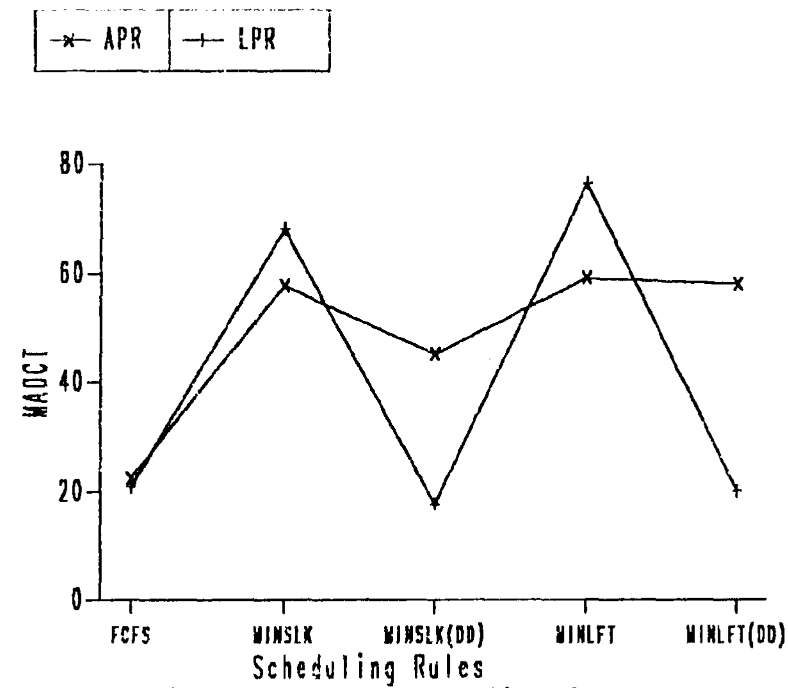



Fig 5.2b MADCT as a Function of
Scheduling and Resource Preemption Rules

A two-way analysis of variance with the cell observations collapsed across the due date factor revealed significant main effects for both the scheduling and resource preemption rule factors with p-values < .001. Scheffe's Method was used to contrast the effects of the scheduling rules within the LPR rule. The results are presented in Table 5.5. In the table, a star (*) denotes a pair of groups (scheduling rules) which are significantly different from each other. The groups are rank ordered according to their relative performance. The marginal means are also presented. An alpha value of .001 is used.

---

**Scheffe's Method**

| MEAN | GROUP | GRP 3 | GRP 5 | GRP 1 | GRP 2 | GRP 4 |
|---|---|---|---|---|---|---|
| 17.77 | Grp 3: MINSLK[DD] | | | | | |
| 20.09 | Grp 5: MINLFT[DD] | | | | | |
| 20.81 | Grp 1: FCFS | | | | | |
| 68.20 | Grp 2: MINSLK | * | * | * | | |
| 76.41 | Grp 4: MINLFT | * | * | * | * | |

alpha = .001

**Table 5.5 Contrasts of Scheduling Rule Performance on MADCT within the LPR Resource Preemption Rule.**

---

In this case, three homogeneous subsets of groups emerge. Of particular interest is the fact that there is not a significant difference between the performance of MINSLK[DD], MINLFT[DD], and FCFS. The performance of the non-due date oriented MINSLK and MINLFT rules are significantly different from this first group and from each other, but their performance is very poor and indicates a lack of usefulness in conjunction with the LPR resource preemption rule. When the scheduling rules are contrasted within the APR resource preemption rule, FCFS dominates in performance. As this result is evident from Figures 5.2a and 5.2b, a separate table is not provided.

In summary, although a two-way interaction exists between the due date and scheduling rules, the interaction is ordinal in its effect. It is clear that the FCFS scheduling rule dominates across all due date rules on the performance measure MADCT in the DDATExSCHED interaction effect. A significant difference in the simple effects of the due date rules within the DDATExSCHED interaction does not exist. Since the scheduling rules significantly interact with the resource preemption rules, it is necessary to consider this interaction when analyzing the effect of the scheduling rules. Again, the FCFS rule tends to dominate or perform at least as well as the other scheduling rules. This occurs despite the fact that the FCFS rule is naive and uses very little information concerning the status of the system. The due date oriented MINSLK[DD] and MINLFT[DD] perform as well as FCFS within the LPR resource preemption rule.

5.2.2 **MEAN WEIGHTED LATENESS.** The Mean Weighted Lateness (MWT) performance measure assigns half as much penalty for early completion of a development project as it does for late completion of the project. The analysis of variance table for MWT is presented in Table 5.6. A significant three-way interaction effect does not exist. Of the three possible two-way interactions, only the SCHEDxRESPRE interaction effect is significant. All three main effects are significant. Since the due date rules factor does not participate in a significant interaction, it is possible to unambiguously interpret the main effects of this factor.

| Source | df | SS | MS | F | Signif. of F |
|---|---|---|---|---|---|
| **Main Effects** | | | | | |
| DDATE | 3 | 4,977 | 1,659 | 15.03 | .000 |
| SCHED | 4 | 129,993 | 32,498 | 294.38 | .000 |
| RESPRE | 1 | 9,338 | 9,338 | 84.59 | .000 |
| **2-Way Interaction** | | | | | |
| DDATExSCHED | 12 | 948 | 79 | .72 | .737 |
| DDATExRESPRE | 3 | 231 | 77 | .70 | .553 |
| SCHEDxRESPRE | 4 | 44,583 | 11,146 | 100.96 | .000 |
| **3-Way Interaction** | 12 | 793 | 66 | .60 | .845 |
| **Explained** | 39 | 190,863 | 4,894 | 44.33 | .000 |
| **Residual** | 760 | 83,902 | 110 | | |
| **Total** | 799 | 274,765 | | | |

**Table 5.6 Analysis of Variance for Mean Weighted Lateness**

In analyzing the SCHEDxRESPRE two-way interaction, the cell mean values of MWT for the two factors are plotted in Figures 5.3a and 5.3b. The resulting graphs are extremely similar to those presented in Figures 5.2a and 5.2b. This

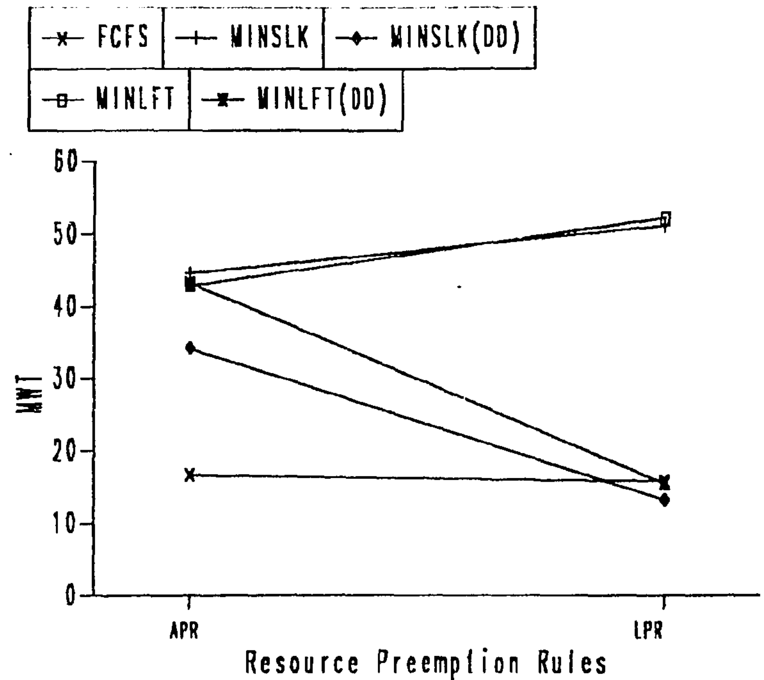



Fig 5.3a MWT as a Function of
Resource Preemption and Scheduling Rules

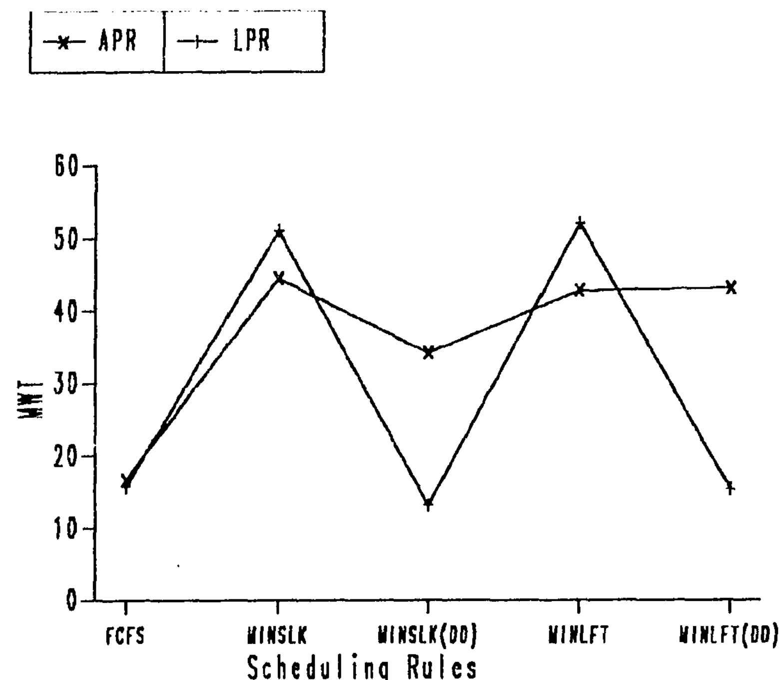



Fig 5.3b MWT as a Function of
Scheduling and Resource Preemption Rules

is not completely unexpected when one considers the nature of the MADCT and MWT performance measures. The difference between them results from projects which are completed ahead of the scheduled due date since the MWT measure only assigns half as much penalty for early completions as does the MADCT measure. A combination of scheduling and resource preemption rules which results in a small value for MADCT should also result in a small value for MWT when the distribution of late and early project completions is fairly balanced. A comparison of Figures 5.2a and 5.2b with Figures 5.3a and 5.3b reveals that, across most rule combinations, the decrease in performance penalty was approximately five to eight work-days.

Within the LPR resource preemption rule, the MINSLK[DD] scheduling rule provides the lowest value of MWT at 13.13. Additionally, the MINLFT[DD] and FCFS rules also provide very low values of MWT at 15.43 and 15.71, respectively. The performance of these three scheduling rules mirrors that obtained with the MADCT measure. Table 5.7 presents a contrast of the scheduling rules within the LPR rule using Scheffe's Method. Two homogeneous groups emerge. The first group includes the MINSLK[DD], FCFS, and MINLFT[DD] scheduling rules. These three rules are not significantly different from one another at the .001 level. The second group includes the MINSLK and MINLFT scheduling rules. The performance of this latter group on the MWT measure is relatively poor.

**Scheffe's Method**

| MEAN | GROUP | GRP 3 | GRP 1 | GRP 5 | GRP 2 | GRP 4 |
|---|---|---|---|---|---|---|
| 13.13 | Grp 3: MINSLK[DD] | | | | | |
| 15.43 | Grp 1: MINLFT[DD] | | | | | |
| 15.71 | Grp 5: FCFS | | | | | |
| 51.07 | Grp 2: MINSLK | * | * | * | | |
| 52.13 | Grp 4: MINLFT | * | * | * | | |

alpha = .001

**Table 5.7 Contrasts of Scheduling Rule Performance on MWT within the LPR Resource Preemption Rule.**

Within the APR resource preemption rule, the FCFS scheduling rule continues to dominate performance. The FCFS rule demonstrates consistency under both resource preemption treatments. A separate contrast within the APR rule was not performed since the FCFS rule clearly dominates in performance as seen in Figures 5.3a and 5.3b.

Next, the significant main effect of the due date rules factor for the MWT measure is analyzed. The question of interest is which due date rule(s) dominates in performance. Collapsing the observations of all cells onto this single factor results in four marginal means for the four due date rules. These are: FLOW = 32.53; NUMACT = 36.11; CPTIME = 33.79; and SFT = 29.20. A comparison of these means reveals that the SFT rule is significantly better than the FLOW and

CPTIME rules at the .05 level. The SFT rule is significantly better than the NUMACT rule at the .01 level.

While the due date rules factor does not significantly interact with the other two factors, there is still an interest in which combination of rules tends to dominate performance. This question arises because, in practice, the due date rules are always employed in conjunction with the other factors. Table 5.8 provides the 40 cell means representing all possible combinations of due date, scheduling, and resource preemption rules. An examination of this table

**Resource Preemption = APR**

| | FCFS | MINSLK | MINSLK[DD] | MINLFT | MINLFT[DD] |
|---|---|---|---|---|---|
| FLOW | 19.42 | 45.13 | 33.89 | 40.96 | 39.94 |
| NUMACT | 19.68 | 47.35 | 36.94 | 47.52 | 46.06 |
| CPTIME | 16.86 | 45.29 | 34.71 | 44.03 | 49.15 |
| SFT | 10.87 | 40.46 | 31.27 | 39.05 | 37.91 |

**Resource Preemption = LPR**

| | FCFS | MINSLK | MINSLK[DD] | MINLFT | MINLFT[DD] |
|---|---|---|---|---|---|
| FLOW | 17.92 | 49.91 | 12.56 | 51.59 | 14.00 |
| NUMACT | 18.63 | 55.05 | 16.33 | 55.62 | 17.90 |
| CPTIME | 16.06 | 52.24 | 12.68 | 52.94 | 13.92 |
| SFT | 10.22 | 47.06 | 10.92 | 48.38 | 15.89 |

**Table 5.8 Values of MWT for All Possible Combinations of Due Date, Scheduling, and Resource Preemption Rules**

reveals a clear dominance of the SFT due date rule when combined with the FCFS rule under either of the resource preemption rules. The SFT due date rule also dominates nine of the 10 column entries.

In summary, the results of the two-way SCHEDxRESPRE interaction for the MWT performance measure are very similar to those found with the MADCT performance measure. Within the LPR resource preemption rule, the MINSLK[DD], FCFS, and MINLFT[DD] rules all perform equally well. The FCFS rule continues to dominate the other scheduling rules within the APR resource preemption rule. The SFT due date rule dominates across both resource preemption rules when used in conjunction with the FCFS scheduling rule.

5.2.3 **MEAN DEVELOPMENT COMPLETION TIME.** The third development performance measure is the Mean Development Completion Time (MDCT). This measure is used to evaluate the mean duration of development projects and captures the flow of development projects through the MIS department.

Table 5.9 presents the analysis of variance table for MDCT. As in the previous two performance measures, a significant three-way interaction effect does not exist. Of the three possible two-way interaction effects, only the SCHEDxRESPRE interaction is significant. Of the three possible main effects, only the main effect of the scheduling and resource preemption factors are significant. The lack of a significant effect from the due date factor for the MDCT performance measure is expected. The only way in which the due date rules can influence the MDCT measure is through a significant interaction with the scheduling rules. Since only two of the scheduling rules, MINSLK[DD] and MINLFT[DD], consider information concerning established project due dates in allocating resources, this result is fairly predictable.

| Source | df | SS | MS | F | Signif. of F |
|---|---|---|---|---|---|
| **Main Effects** | | | | | |
| DDATE | 3 | 225 | 75 | .29 | .833 |
| SCHED | 4 | 125,427 | 31,357 | 120.96 | .000 |
| RESPRE | 1 | 37,795 | 37,795 | 145.79 | .000 |
| **2-Way Interaction** | | | | | |
| DDATExSCHED | 12 | 479 | 40 | .15 | 1.000 |
| DDATExRESPRE | 3 | 290 | 97 | .37 | .773 |
| SCHEDxRESPRE | 4 | 66,914 | 16,728 | 64.53 | .000 |
| **3-Way Interaction** | | | | | |
| | 12 | 1,006 | 84 | .32 | .985 |
| **Explained** | 39 | 232,136 | 5,952 | 22.96 | .000 |
| **Residual** | 760 | 197,026 | 259 | | |
| **Total** | 799 | 429,162 | | | |

**Table 5.9 Analysis of Variance for Mean Development Completion Time**

In analyzing the SCHEDxRESPRE two-way interaction, the cell mean values of MDCT for the two factors are plotted in Figures 5.4a and 5.4b. The graphs are remarkably similar to those presented in Figures 5.2 and 5.3. While this similarity was not expected, it is extremely interesting. The results depicted by the graphs demonstrate the consistency

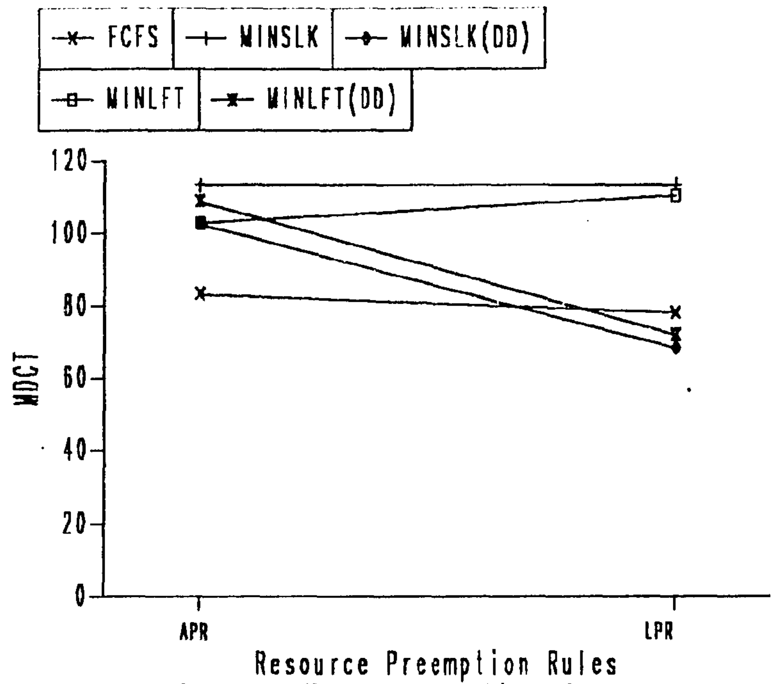



Fig 5.4a MDCT as a Function of Resource Preemption and Scheduling Rules

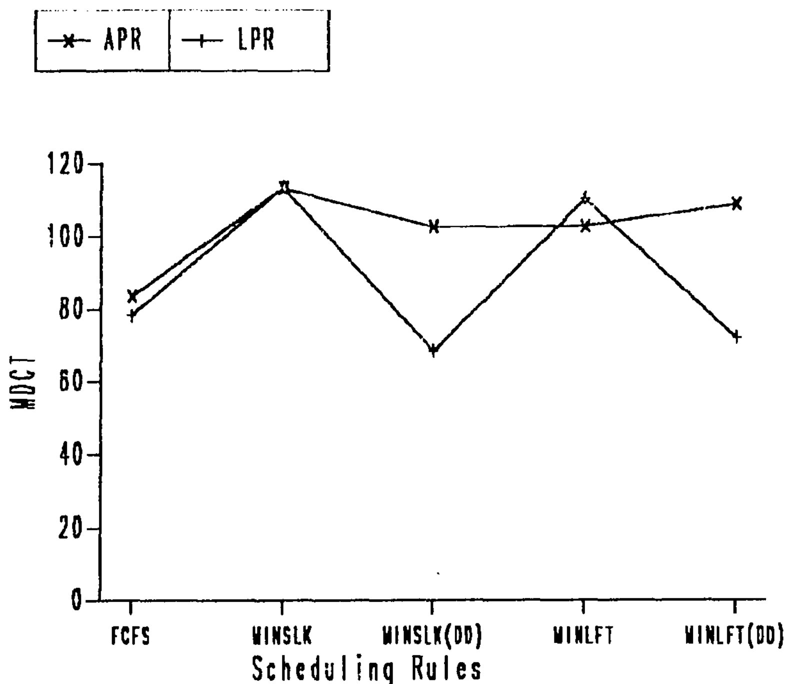



Fig 5.4b MDCT as a Function of Scheduling and Resource Preemption Rules

of the effects of the experimental factors across three different performance measures, MADCT, MWT, and MDCT.

Within the LPR resource preemption rule, the MINSLK[DD] scheduling rule again provides the lowest performance measure value of 68.41 for MDCT. This means that the average development project was completed in 68.41 work-days. The MINLFT[DD] and FCFS scheduling rules also perform well with values for MDCT of 72.09 and 78.11, respectively.

Table 5.10 presents a contrast of the scheduling rules within the LPR rule using Scheffe's Method. As with the MWT performance measure, two homogeneous groups emerge. The first group includes MINSLK[DD], MINLFT[DD], and FCFS. As occurred with the MADCT and MWT performance measures, these three rules are not significantly different from one another within the LPR resource preemption rule.

---

**Scheffe's Method**

| MEAN | GROUP | GRP 3 | GRP 5 | GRP 1 | GRP 2 | GRP 4 |
|---|---|---|---|---|---|---|
| 68.41 | Grp 3: MINSLK[DD] | | | | | |
| 72.09 | Grp 5: MINLFT[DD] | | | | | |
| 78.11 | Grp 1: FCFS | | | | | |
| 110.42 | Grp 2: MINSLK | * | * | * | | |
| 113.40 | Grp 4: MINLFT | * | * | * | | |

alpha = .001

**Table 5.10 Contrasts of Scheduling Rule Performance on MDCT within the LPR Resource Preemption Rule.**

---

Within the APR resource preemption rule, the FCFS scheduling rule dominates in performance on the MDCT measure. These results are comparable to those obtained on the MADCT and MWT measures.

In summary, the resulting effects from the experimental factors, when evaluated using the MDCT performance measure, are remarkably consistent with the results obtained for the MADCT and MWT performance measures. Within the LPR resource preemption rule, the MINSLK[DD], FCFS, and MINLFT[DD] scheduling rules dominate performance and are not statistically significant from one another. The FCFS performs consistently well within both resource preemption rules. The due date rules factor does not significantly affect the MDCT performance measure.

## 5.3 ANALYSIS OF MAINTENANCE PROJECT PERFORMANCE MEASURES.

The sole performance measure for maintenance projects is the Mean Maintenance Completion Time (MMCT). This measure captures the flow of maintenance projects through the MIS Department based upon their mean duration time.

Table 5.12 presents the analysis of variance table for this measure. Only the SCHEDxRESPRE interaction effect and the main effects of the scheduling and resource preemption rule factors are significant. The cell mean values of MMCT for the SCHEDxRESPRE interaction are plotted in Figures 5.5a and 5.5b. The interaction effects are strictly ordinal. These graphs demonstrate the overwhelming effect of the Absolute Priority for Resources rule. When the priority for resources is always given to maintenance projects, the scheduling rule used to prioritize resources for development projects does not significantly affect the duration of maintenance work.

| Source | df | SS | MS | F | Signif. of F |
|---|---|---|---|---|---|
| **Main Effects** | | | | | |
| DDATE | 3 | 59 | 20 | 1.16 | .322 |
| SCHED | 4 | 13,273 | 3,318 | 195.40 | .000 |
| RESPRE | 1 | 66,021 | 66,021 | 3,887.70 | .000 |
| **2-Way Interaction** | | | | | |
| DDATExSCHED | 12 | 111 | 9 | .54 | .887 |
| DDATExRESPRE | 3 | 59 | 20 | 1.17 | .322 |
| SCHEDxRESPRE | 4 | 13,355 | 3,339 | 196.59 | .000 |
| **3-Way Interaction** | 12 | 110 | 9 | .54 | .891 |
| **Explained** | 39 | 92,987 | 2,384 | 140.40 | .000 |
| **Residual** | 760 | 12,906 | 17 | | |
| **Total** | 799 | 105,894 | | | |

**Table 5.11 Analysis of Variance for Mean Maintenance Completion Time**

Significant differences between the effects of the scheduling rules are evident within the Limited Priority for Resources rule. From Figures 5.5a and 5.5b, it is obvious that the FCFS scheduling rule dominates. The FCFS rule is followed by a group consisting of the MINSLK and MINLFT rules. A contrast of the scheduling rules is presented in Table 5.13. The results of the contrast confirm those depicted by Figures 5.5a and 5.5b.

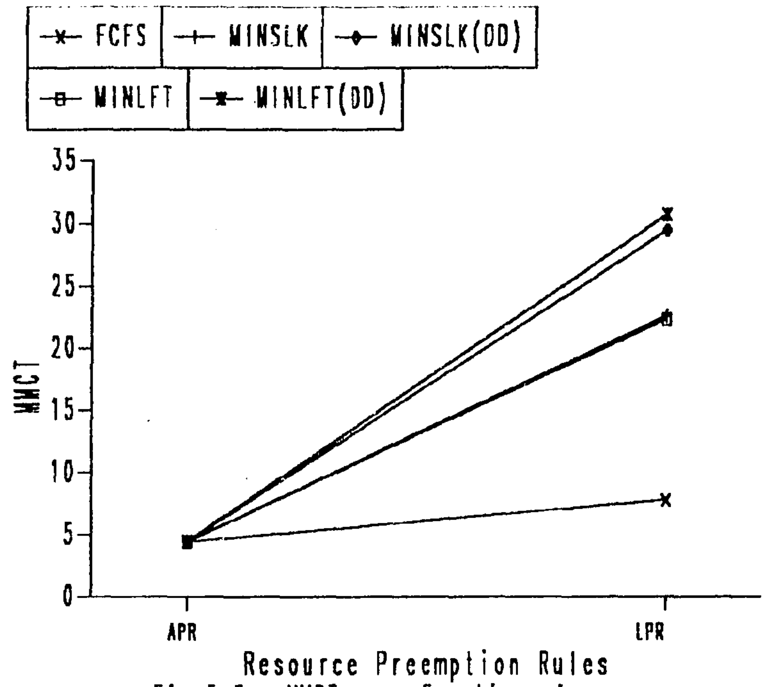



Fig 5.5a MMCT as a Function of Resource Preemption and Scheduling Rules

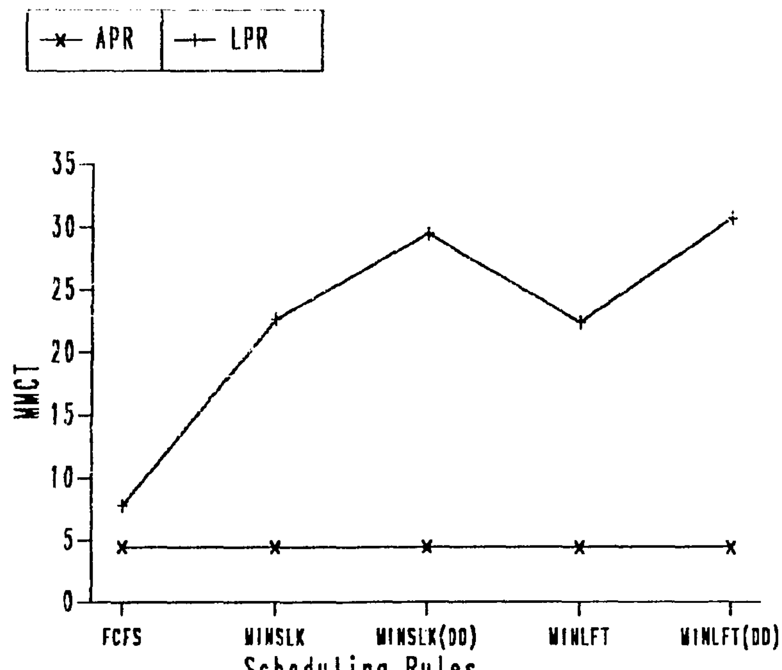



Fig 5.5b MMCT as a Function of Scheduling and Resource Preemption Rules

The results for the MMCT performance measure are intuitive given the previous analysis for the MDCT performance measure. Quite simply, when one relaxes the priority of resources for maintenance projects, the scheduling rule used to prioritize resources for development projects begins to also affect the flow of maintenance projects through the MIS Department. The effect is dramatic with the mean duration varying from a low mean of 7.8 work-days for the FCFS rule to a high mean of 30.74 work-days for the MINLFT[DD] rule. As would be expected, when the scheduling and resource preemption rules jointly consider the due date in determining whether or not to allow resource preemption, the duration of maintenance projects is greatest.

---

**Scheffe's Method**

| MEAN | GROUP | GRP 1 | GRP 4 | GRP 2 | GRP 3 | GRP 5 |
|---|---|---|---|---|---|---|
| 7.80 | Grp 1: FCFS | | | | | |
| 22.35 | Grp 4: MINLFT | * | | | | |
| 22.62 | Grp 2: MINSLK | * | | | | |
| 29.49 | Grp 3 MINSLK[DD] | * | * | * | | |
| 30.74 | Grp 5: MINLFT[DD] | * | * | * | | |

alpha = .001

**Table 5.12 Contrasts of Scheduling Rule Performance on MMCT within the LPR Resource Preemption Rule.**

---

One of the more interesting results for the MMCT measure concerns the continued dominance of the FCFS scheduling rule. When moving from the APR to the LPR resource preemption rule, the FCFS rule only increases the mean duration for maintenance projects by approximately 3 work-days. Recall that the FCFS rule also provides good performance on the MDCT measure for the mean duration of development projects. Thus MIS departments may obtain a reasonable trade-off between the competing objectives of minimizing mean duration for development and maintenance projects when using the FCFS rule. Further, this result is achievable within both the APR and LPR resource preemption rules.

In summary, the effect of the development project scheduling rules factor is not significant on maintenance project duration within the APR resource preemption rule. Within the LPR rule, the scheduling rules factor does affect maintenance project duration with the FCFS scheduling rule dominating performance. The duration of maintenance projects increases rapidly when a scheduling rule other than FCFS is utilized.

## 5.4 SUMMARY OF EXPERIMENTAL RESULTS.

This section summarizes the results of the performance measures for the three experimental factors. The main objective of the experiment is to determine the efficacy of a set of project due date rules. Additionally, the experiment provides the opportunity to examine the effects of a set of scheduling rules and a set of resource preemption rules.

Considering the due date rules factor first, it was not found that one due date rule dominated the MADCT and MWT performance measures. For the MADCT measure, the due date rules factor interacted significantly with the scheduling rules factor. The SFT due date rule, in conjunction with the FCFS scheduling and LPR resource preemption rules tended to produce the least error, however, a significant difference between the due date rule effects was not discernible.

For the MWT measure, the due date rules factor did not significantly interact with other factors. In analyzing the main effects, it was found that the SFT due date rule, in conjunction with the FCFS scheduling rule, produced significantly less error across both the APR and LPR resource preemption rules.

In analyzing the effects of the scheduling rules factor, it was found that consistent results were obtained for the MADCT, MWT, and MDCT development project performance measures. The scheduling and resource preemption rules factors significantly interact on each measure. Within the LPR resource preemption rule, the MINSLK[DD], MINLFT[DD], and FCFS scheduling rules consistently dominate. The scheduling rules in this group are not significantly different from one another across all three development project performance measures. Within the APR resource preemption rule, the FCFS scheduling rule is consistently dominant.

The results of the scheduling rule effects have a practical significance. Recall that the MADCT, MWT, and

MDCT performance measures represent different metrics which may be used to evaluate the sagacity of management. From a managerial perspective, the consistency of performance by the scheduling rules across these metrics should be appreciated. Management may select a scheduling rule without the need to consider the normal trade-offs which are expected whenever one's performance is subject to measurement on different metrics.

The MMCT measure was used to evaluate experimental effects on mean maintenance project duration. It was found that the scheduling rules factor did not affect the MMCT within the APR resource preemption rule. However, when the managerial policy of limiting resource priority for maintenance projects is in effect, the scheduling rules affect the MMCT. The FCFS scheduling rule produces the smallest MMCT. In fact, the use of any of the other four scheduling rules results in an increase in mean duration of maintenance projects which may be viewed by management as being unacceptably excessive.

## 6.0 SENSITIVITY ANALYSIS.

During the conduct of this research, two questions naturally arose as to the sensitivity of the results to certain input parameters. The first question concerns sensitivity to the selected values for the "F" and "K" parameters which are used in each due date setting rule. Recall that each of the due date rules examined in this research requires a "F" or "K" parameter. The function of the "F" or "K" parameter is to establish a development project due date that allocates slack time to the project. This slack is necessary because it is known that additional demands will be placed on the project design team. These demands take the form of additional development projects, maintenance project requirements, etc. Ideally, the slack provided will result in zero deviation between the due date and actual date of project completion. This deviation is termed Deviation of Due Date.

Chapter 3 indicated that the "F" and "K" parameters were selected for the experiment by running the computer simulation program to simulate a historical pattern of due date performance. A search procedure was then used to select parameter values which produce near zero Deviation of Due Date. Although it is unlikely that parameter values with this high degree of accuracy could be determined in an operational MIS department, it is reasonable to expect that values of "F" or "K" could be achieved which are not more than plus or minus 10 percent different from those values which produce near zero Deviation of Due Date. The sensitivity analysis, therefore, examines the effect of selecting "F" and "K" parameter values which are plus or minus 10 percent in error. The sensitivity experiment was conducted in exactly the same manner as the main experiment with the exception of this change.

The second question concerns the relationship between the "F" and "K" parameters and changes in resource utilization. Resource utilization is a factor which is likely to differ in another setting. The objective of this experiment is to investigate possible functional relationships between the "F" and "K" due date parameter values and resource utilization. Should such a relationship exist, it would enable managers to adjust the values of "F" or "K" in conjunction with changes in the resource utilization policy. For this question, the effect of increasing or decreasing the resource utilization level by increasing or decreasing the arrival of projects to the MIS department is examined. This experiment was conducted only for the Scheduled Finish Time due date and First Come-First Served resource scheduling rule combination within the Absolute Priority for Resources preemption policy.

The first three sections of this chapter address the question of under or overestimating the appropriate value for the "F" and "K" due date parameters. Section 6.1

examines the sensitivity of the Deviation of Due Date measure to changes in the "F" and "K" parameter values. Section 6.2 discusses analysis of variance results for the sensitivity analysis. Section 6.3 presents a detailed examination of the sensitivity of the primary performance measures to changes in the parameter values. Section 6.4 discusses the relationship between the "F" and "K" parameter values and the resource utilization level.

## 6.1 DEVIATION OF DUE DATE SENSITIVITY.

Since the parameter values of "F" and "K" are selected with an objective of achieving near zero Deviation of Due Date, it is desirable that this performance measure demonstrate consistency in response to errors in the parameter value estimates. By consistency, it is meant that when "F" and "K" parameter values are too low, the Mean Deviation of Due Date should be negative. This indicates the average project does not have sufficient slack and also gives the mean number of days by which projects exceed the estimated due date. Conversely, when "F" and "K" parameter values are set too high, the mean Deviation of Due Date should be positive.

Figure 6.1 and Table 6.1 give the results of the sensitivity of the Deviation of Due Date response variable in graphical and tabular form. The second column of Table 6.1 gives the results for the four due date rules in the main experiment. Columns one and three give the Deviations of Due Date for the 10 percent low and high values of the

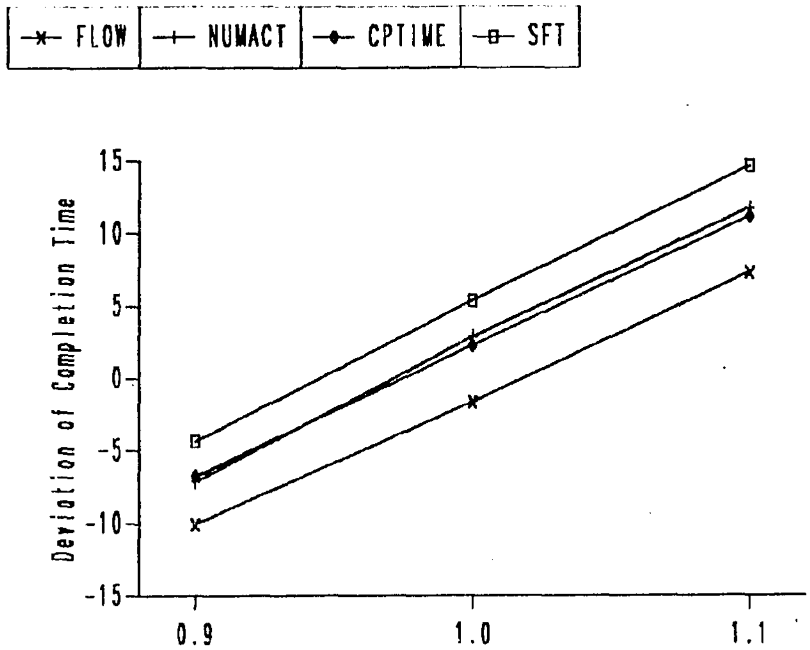



Fig 6.1 Deviation of Due Date for Development Projects for Four Due Date Rules

| | 0.9xK | 1.0xK | 1.1xK |
|---|---|---|---|
| FLOW | -10.09 | -1.67 | 7.28 |
| NUMACT | -7.18 | 2.86 | 11.72 |
| CPTIME | -6.78 | 2.24 | 11.15 |
| SFT | -4.38 | 5.34 | 14.65 |

Table 6.1 Deviation of Due Date for Development Projects for Four Due Date Rules

"F" and "K" parameters, respectively. Each of the due date rules have different values of "F" or "K" for each possible combination of due date, resource scheduling, and resource preemption rules.

It is necessary to explain why the values in the center column of Table 6.1 are not zero. During the main experiment, the same project stream used to establish historical values for "F" and "K" was again used for the actual data collection runs. During the data collection runs, however, the random number seeds were varied. This resulted in different project interarrival times, activity duration times, etc. The purpose of varying the random number seeds is to simulate a stream of future projects similar to, but not exactly like, past projects. With the change in random number seeds, the Deviation of Due Date is not as close to zero deviation as it was during estimation of the "F" and "K" parameter values. The deviation is still small, generally within a two to three percent error of the mean development project duration, reflecting the fact that acceptable parameter values were selected.

The results given in Figure 6.1 and Table 6.1 are quite intuitive. As expected, when less slack is provided by the "F" and "K" parameters, the Deviation of Due Date is negative for all four due date rules. Conversely, when additional slack is provided, the Deviation of Due Date is positive. It is also interesting to note that the Deviation of Due Date is nearly linear in its sensitivity to the values of the "F" and "K" parameters within the plus and minus 10 percent error evaluated. A change in the "F" or "K" parameter of 10 percent results in a change in the Deviation of Due Date of approximately nine to 10 days. This is consistent since the mean duration of development projects is approximately 95 days.

## 6.2 ANALYSIS OF VARIANCE.

One question of interest concerns possible changes in the significance of the interaction and main effects which were found for each of the performance measures in the main experiment. To review, these measures include: Mean Absolute Deviation of Completion Time (MADCT), Mean Weighted Lateness (MWT), Mean Development Completion Time (MDCT), and Mean Maintenance Completion Time (MMCT). While it has been shown that the results for the Deviation of Due Date are predictable and intuitive, this may not be true for the MADCT, MWT, MDCT, and MMCT performance measures.

To answer this question, two sets of analysis of variance procedures were conducted, one set for the data resulting from computer simulation runs using low values of "F" and "K" parameter values, and one set for the data resulting from high values of "F" and "K". The results of the two analysis of variance procedures are consistent with those from the main experiment. For each performance measure, the exact same main and interaction effects were statistically significant at the .001 level as occurred in the main experiment. For this reason, the analysis of variance tables are not presented.

## 6.3 ADDITIONAL SENSITIVITY QUESTIONS.

While the analysis of variance results indicate no overall change for the main and interaction effects, there are possible differences in the relative effectiveness of the rules examined. One question concerns the sensitivity of performance of the due date rules on the Mean Absolute Deviation of Completion Time and Mean Weighted Lateness measures. A second question of interest focuses on the effect on the performance of combinations of due date and scheduling rules within particular resource preemption rules. A third question concerns the sensitivity of mean development and maintenance project duration to changes in the "F" and "K" parameter values. These questions are addressed in the sections presented below.

**6.3.1 DUE DATE RULE PERFORMANCE.**, The results of introducing a 10 percent error in the "F" and "K" parameters for the four due date rules on the Mean Absolute Deviation of Completion Time and Mean Weighted Lateness measures are given in Tables 6.2 and 6.3, respectively. Figures 6.2 and 6.3 portray the results graphically.

Examining the sensitivity of the Mean Absolute Deviation of Completion Time measure first, one finds that the results are not entirely intuitive. An increase in the values of the "F" and "K" parameters produces the expected decrease in performance for each of the due date rules.

However, while performance should also deteriorate with lower values for the "F" and "K" parameters, performance actually improves. The sensitivity of the due date rules is approximately equal with the SFT rule providing the best performance in the 0.9xK category. It is apparent that the selection of "F" and "K" parameter values by seeking to minimize the Deviation of Completion Time does not necessarily result in the best values of "F" and "K" for the MADCT performance measure.

| | **0.9xK** | **1.0xK** | **1.1xK** |
|---|---|---|---|
| FLOW | 40.75 | 42.82 | 46.04 |
| NUMACT | 46.00 | 49.10 | 53.85 |
| CPTIME | 42.24 | 45.80 | 50.12 |
| SFT | 37.18 | 40.72 | 46.85 |

**Table 6.2 Mean Absolute Deviation of Completion Time for Development Projects for Four Due Date Rules**

| | **0.9xK** | **1.0xK** | **1.1xK** |
|---|---|---|---|
| FLOW | 33.08 | 32.53 | 32.71 |
| NUMACT | 36.29 | 36.11 | 37.46 |
| CPTIME | 33.38 | 33.79 | 34.80 |
| SFT | 28.98 | 29.20 | 31.47 |

**Table 6.3 Mean Weighted Lateness for Development Projects for Four Due Date Rules**

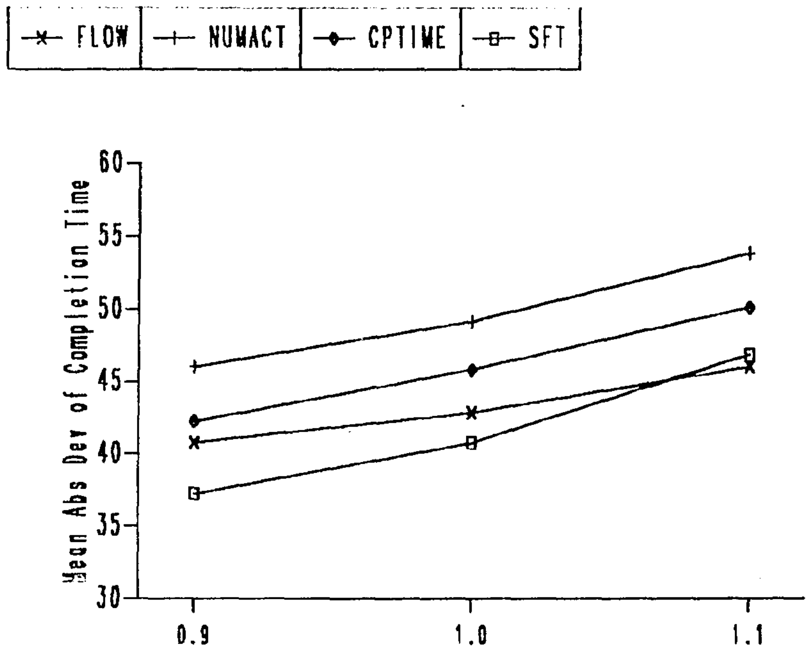



Fig 6.2 Due Date Performance on the
Mean Absolute Deviation of Completion Time
Measure for Development Projects

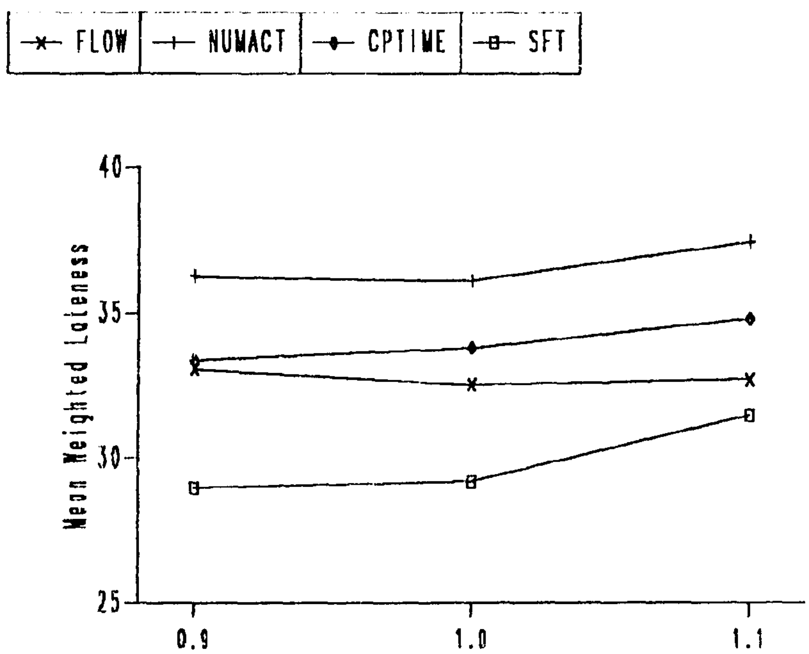



Fig 6.3 Due Date Performance on the
Mean Weighted Lateness Measure
for Development Projects

Performance on the Mean Weighted Lateness measure is more predictable. In general, performance deteriorates or exhibits relatively little change when lower or higher values of the "F" and "K" parameters are selected. Due date rule performance is relatively insensitive to 10 percent errors in the "F" and "K" parameter values for the Mean Weighted Lateness measure.

In summary, it is apparent that the minimization of the Deviation of Completion Time did not result in the best values of the "F" and "K" parameters for the Mean Absolute Deviation of Completion Time measure. Performance for all four due date rules improves on this measure when lower values of "F" and "K" are used. The four due date rules demonstrate relatively little sensitivity on the Mean Weighted Lateness measure.

**6.3.2 DUE DATE AND SCHEDULING RULE PERFORMANCE.** The main experiment revealed that the non-due date sensitive Minimum Slack (MINSLK) and Minimum Late Finish Time (MINLFT) scheduling rules perform very poorly in all possible due date and resource preemption rule combinations. In the sensitivity analysis, their performance is again consistently poor. For this reason, the discussion of scheduling rules is limited to the First Come-First Served (FCFS) and the two MINSLK[DD] and MINLFT[DD] due date sensitive scheduling rules.

First, the results of due date and scheduling rule combinations within the Absolute Priority for Resources

(APR) resource preemption rule are summarized. As in the main experiment, the FCFS scheduling rule dominates performance across all due date rules. For this reason, the performance of the other scheduling rules is not presented. Table 6.4 and 6.5 give the values of the Mean Absolute Deviation of Completion Time and Mean Weighted Lateness for the FCFS scheduling rule across all four due date rules. The results are not graphed because the similar values provide no additional information for analysis.

| | **0.9xK** | **1.0xK** | **1.1xK** |
|---|---|---|---|
| FLOW | 25.74 | 25.72 | 27.42 |
| NUMACT | 25.42 | 26.12 | 28.39 |
| CPTIME | 22.38 | 22.33 | 24.01 |
| SFT | 15.41 | 16.08 | 20.15 |

**Table 6.4 Mean Absolute Deviation of Completion Time for the FCFS Scheduling Rule and Absolute Priority for Resources Preemption Rule**

| | **0.9xK** | **1.0xK** | **1.1xK** |
|---|---|---|---|
| FLOW | 21.50 | 19.42 | 18.62 |
| NUMACT | 21.23 | 19.68 | 19.31 |
| CPTIME | 18.97 | 16.86 | 16.05 |
| SFT | 12.57 | 10.87 | 11.48 |

**Table 6.5 Mean Weighted Lateness for the FCFS Scheduling Rule and Absolute Priority for Resources Preemption Rule**

Tables 6.4 and 6.5 reveal two noteworthy points. First, there is little evidence of sensitivity by the FCFS rule to errors in the values of "F" and "K". This observation is true across all due date rules. Second, the FCFS and SFT Rule combinations dominate across all ranges of the "F" and "K" parameters.

Next, the results of due date and scheduling rule combinations within the Limited Priority for Resources (LPR) resource preemption rule are presented. Tables 6.6 and 6.7 give selected values of the Mean Absolute Deviation of Completion Time and Mean Weighted Measures measures for the SFT due date rule in combination with the MINSLK[DD], MINLFT[DD], and FCFS scheduling rules. These results are also graphed in Figures 6.4 and 6.5.

The MINLFT[DD] scheduling rule is somewhat sensitive to the errors in the "F" and "K" parameter values, however its

| | **0.9xK** | **1.0xK** | **1.1xK** |
|---|---|---|---|
| FCFS | 14.41 | 14.53 | 17.63 |
| MINSLK[DD] | 12.83 | 16.50 | 21.22 |
| MINLFT[DD] | 15.28 | 20.49 | 24.72 |

**Table 6.6 Mean Absolute Deviation of Completion Time for the SFT Due Date Rule and Limited Priority for Resources Preemption Rule**

| | 0.9xK | 1.0xK | 1.1xK |
|---|---|---|---|
| FCFS | 12.16 | 10.22 | 10.30 |
| MINSLK[DD] | 9.69 | 10.92 | 13.00 |
| MINLFT[DD] | 12.17 | 15.89 | 17.86 |

**Table 6.7 Mean Weighted Lateness for the SFT Due Date Rule and Limited Priority for Resources Preemption Rule**

performance is dominated by both the MINSLK[DD] and FCFS scheduling rules. The MINSLK[DD] scheduling rule is also somewhat sensitive to the errors. When the "F" and "K" parameters are reduced by 10 percent, the MINSLK[DD] rule dominates performance, however the difference is not statistically significant at the .05 level. The FCFS scheduling rule demonstrates the least sensitivity to errors in the "F" and "K" parameters.

In summary, the SFT due date and FCFS scheduling rule tend to dominate all other due date and scheduling rule combinations within the APR resource preemption rule. Within the LPR resource preemption rule, the performance of the MINSLK[DD] and MINLFT[DD] scheduling rules deteriorates with higher values of "F" and "K" and improves with lower values of "F" and "K". The FCFS scheduling rule is relatively insensitive to changes in the parameter values. The MINSLK[DD] and SFT rule combination dominate when the "F" and "K" parameter values are decreased by 10 percent.

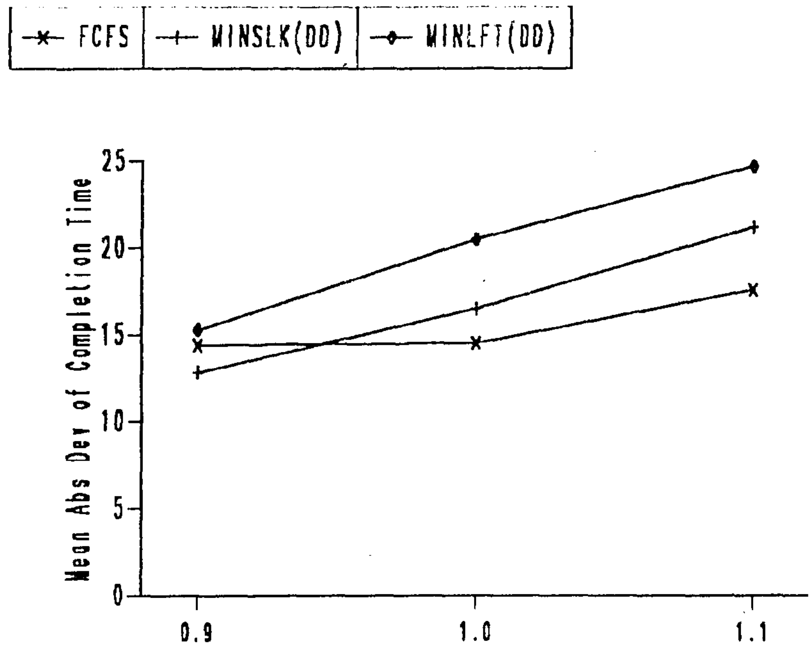



Fig 6.4 SFT Due Date Performance Within the LPR Resource Preemption Rule for the Mean Abs Dev of Completion Time Measure

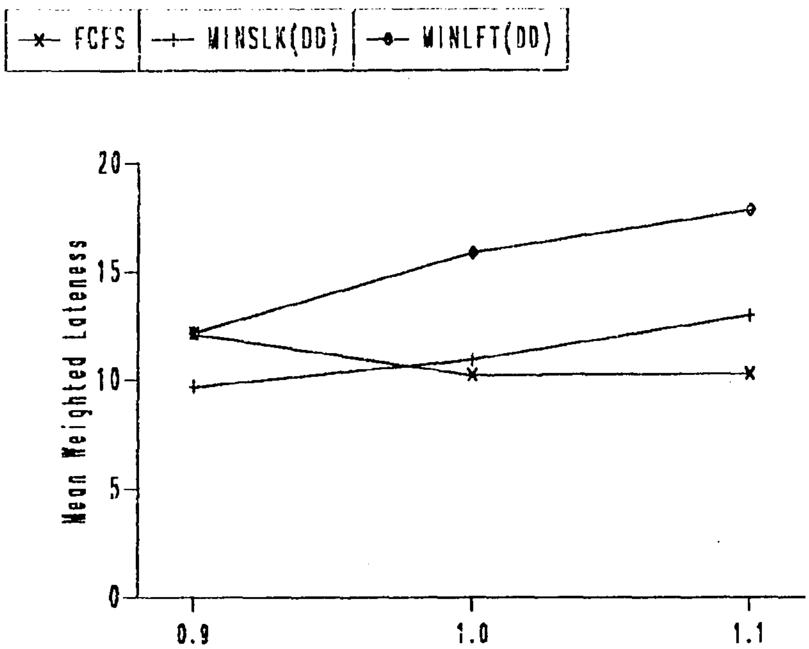



Fig 6.5 SFT Due Date Performance Within the LPR Resource Preemption Rule for the Mean Weighted Lateness Measure

**6.3.3 PROJECT DURATION.** Recall that the Mean Development Completion Time and Mean Maintenance Completion Time are measures of average project duration. The only manner in which these measures can exhibit sensitivity is through interaction with the MINSLK[DD] and MINLFT[DD] scheduling rules. Tables 6.8 and 6.9 give the mean duration values for development and maintenance projects, respectively. As expected, there is little sensitivity to errors in the "F" and "K" parameter values for either of these performance measures.

| | 0.9xK | 1.0xK | 1.1xK |
|---|---|---|---|
| FLOW | 94.77 | 95.75 | 96.21 |
| NUMACT | 94.84 | 94.46 | 95.25 |
| CPTIME | 94.75 | 95.45 | 96.40 |
| SFT | 95.73 | 95.76 | 97.39 |

**Table 6.8 Mean Development Completion Time**

| | 0.9xK | 1.0xK | 1.1xK |
|---|---|---|---|
| FLOW | 13.84 | 13.93 | 13.10 |
| NUMACT | 13.28 | 13.49 | 13.17 |
| CPTIME | 13.25 | 13.47 | 13.30 |
| SFT | 13.55 | 13.16 | 13.45 |

**Table 6.9 Mean Maintenance Completion Time**

## 6.4 RESOURCE UTILIZATION SENSITIVITY ANALYSIS.

In the main experiment, a factor termed the Interarrival Rate Factor is set at 1.50. This setting produces a mean arrival rate of projects of 1 project every 2.25 days. To examine the functional relationship between resource utilization and the selected values of "F" and "K", the Interarrival Rate Factor was varied from a low value of 1.30 to a high value of 1.70 at .05 intervals. This produced a mean arrival rate of projects which varied from a low of 1 project every 1.95 days to a high of 1 project every 2.55 days. The lower the value of the Interarrival Rate Factor, the faster projects arrive to the MIS department.

Data were collected for several statistics. These included the Mean Resource Utilization, Mean Development Completion Time (MDCT) measure of development project duration, Mean Absolute Deviation of Completion Time (MADCT) and Mean Weighted Lateness (MWT) performance measures, and Mean Deviation of Due Date (MDDD). Table 6.10 gives the values of these statistics at the selected Interarrival Rate Factors. Figure 6.6 presents the results graphically.

Table 6.10 and Figure 6.6 indicate that the duration for development projects (MDCT) increases as resource utilization increases. Similarly, while the average number of development projects arriving to the MIS department increases, the average number completed in the 750 work-day

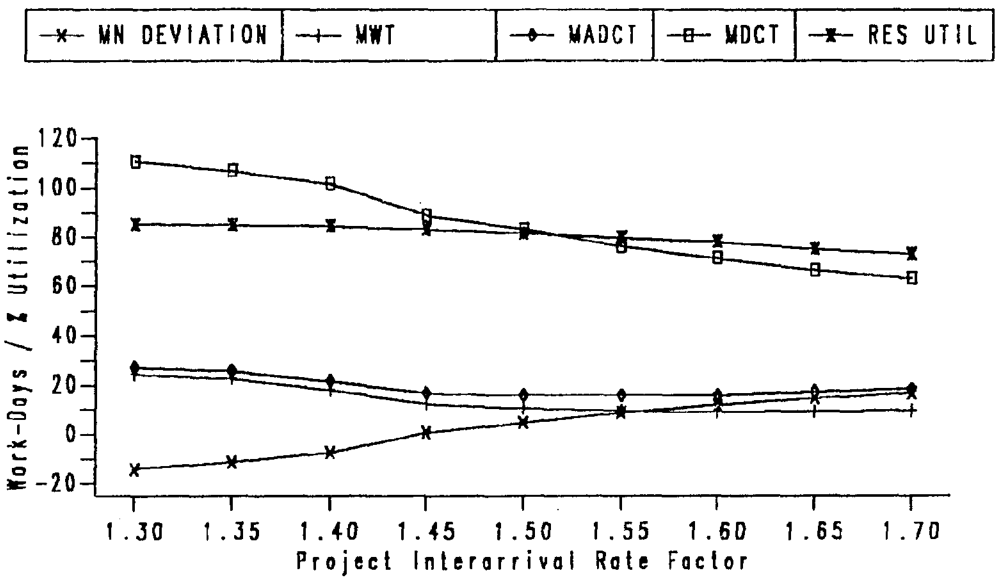



Fig 6.6 Effects of Changing Resource Utilization Level

| | **Interarrival Rate Factor** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **1.30** | **1.35** | **1.40** | **1.45** | **1.50** | **1.55** | **1.60** | **1.65** | **1.70** |
| Resource Util. | 85.4 | 85.0 | 84.7 | 83.3 | 81.7 | 80.0 | 78.2 | 75.2 | 73.2 |
| MDCT | 110.9 | 107.0 | 101.9 | 89.0 | 83.3 | 76.3 | 71.4 | 66.6 | 63.3 |
| MADCT | 27.37 | 25.81 | 21.80 | 17.00 | 16.07 | 16.01 | 16.03 | 17.44 | 18.60 |
| MWT | 24.5 | 22.2 | 18.2 | 12.6 | 10.8 | 9.8 | 9.0 | 9.4 | 9.7 |
| MDDD | -14.1 | -11.3 | -7.4 | 0.7 | 4.8 | 9.0 | 11.9 | 14.9 | 17.0 |
| Avg Devel. Proj Completed | 66.8 | 70.8 | 79.0 | 86.0 | 89.9 | 91.6 | 92.7 | 93.4 | 93.7 |

**Table 6.10 Selected Statistical Values at Various Project Interarrival Rate Factors**

duration of a simulation run decreases. This would appear to be counter-intuitive at first. Remember, however, that resource availability, the quantity of resources available, has not increased. Rather, management is attempting to increase designer output by increasing the workload on the MIS department and reducing idleness. The increase in congestion in the MIS department results in extremes in development project duration ranging from a low of 63.3 to a high of 110.9 work-days.

Table 6.10 and Figure 6.6 also indicate that, as resource utilization increases, the Mean Deviation of Due Date becomes negative. This means that projects are completed later than the scheduled due date. Similarly, if resource utilization decreases, projects are completed ahead of the scheduled due date. This may also appear to be counter-intuitive. Again one must consider increase in congestion in the MIS department when workloads are increased. The many activities which are concurrently competing for resources result in an inability to meet scheduled due dates.

The question of interest is: "How much additional slack should be allocated to a project to maintain due date performance at the level achieved prior to increasing or decreasing resource utilization?" When resource utilization is lowered (values for the Interarrival Rate Factor greater than 1.50), performance, as measured by the Mean Absolute Deviation of Completion Time and Mean Weighted Lateness, is almost unaffected. The MADCT varies from 16.07 to 18.55 while the MWT varies from 10.84 to 9.67. When resource utilization is raised, however, performance deteriorates on both of these measures. The MADCT increases to 27.37 and the MWT increases to 24.5.

An attempt to identify the functional relationship underlying these results was made by developing ratios of MADCT, MWT, and MDDD to resource utilization. The resulting ratios indicate that the relationship between the "F" and "K" parameter values and the resource utilization level is nonlinear. An examination of the functions plotted in Figure 6.6 would lead one to anticipate this outcome. Therefore, it was not possible to identify a functional relationship between the "F" and "K" parameter values and the resource utilization level which demonstrates the consistency necessary to generalize from the simulation experiment to an operational setting.

The experiment did provide some practical information. Managers can expect development project duration performance and the MADCT and MWT measures of performance to degrade markedly whenever attempts are made to increase resource utilization. Performance does not vary for the MADCT or MWT measures when resource utilization is lowered, however it is unlikely that MIS management will seek to move in such a direction.

## 6.5 SUMMARY OF SENSITIVITY ANALYSIS.

The purpose of this analysis was to answer questions regarding the sensitivity of the performance measures to changes in certain input parameters. Regarding sensitivity to errors in the values of the "F" and "K" parameters, within the plus or minus 10 percent error limits investigated, little change in the relative performance of the rules was found. It is comforting that the analysis of variance results are identical in regards to which main and interaction effects are significant for the various performance measures.

In analyzing the relative effectiveness of combinations of due date, scheduling, and resource preemption rules, very little change is evident. The only change occurred with "F" and "K" parameters set 10 percent lower than in the main experiment. At the lower setting, the SFT due date and MINSLK[DD] scheduling rule combination dominate within the LPR preemption rule where the SFT due date and FCFS scheduling rule dominate at higher values of "F" and "K". This change, however, is not statistically significant at the .05 level.

It is noteworthy that due date rule performance improved on the Mean Absolute Deviation of Completion Time measure when the "F" and "K" parameter values were lowered. However, due date rule performance was relatively insensitive on the Mean Weighted Lateness measure.

The experiment conducted to attempt to determine the functional relationship between the "F" and "K" parameter values and the resource utilization level was not successful. The examination of selected performance measure to resource utilization level ratios revealed that the functional relationship was non-linear in each case. A consistent functional relationship could not be identified. This makes it impossible to develop a general statement of the functional relationship which would be useful in an operational setting.

## 7.0 SUMMARY AND DIRECTIONS FOR FUTURE RESEARCH.

This chapter summarizes the results of the research. Certain combinations of the rules examined demonstrate superior performance. Based upon these results, suggestions are made as to the selection of an appropriate decision rule combination. The research results reported herein are also compared to results reported in the literature. The chapter concludes with a section which addresses recommendations for future research.

### 7.1 RESEARCH SUMMARY.

The research was conducted in two phases. The objective of Phase 1 consisted of the identification and verification of a model of the MIS systems development environment. Model verification was accomplished through a series of interviews with MIS professionals in eight different MIS departments. This phase resulted in a model MIS department suitable for experimentation by use of computer simulation methodology. The results of the interviews are detailed and summarized in Chapter 4.

In Phase 2, a three-factor, full-factorial simulation experiment was conducted to examine the relative performance of four due date, five resource scheduling, and two resource preemption rules. The results of these experiments are described in Chapter 5. Additionally, a sensitivity analysis was conducted to examine the effect of deviating from the "F" and "K" parameter values used in estimating due dates for the main experiment. Chapter 6 gives the results of the sensitivity analysis.

**7.1.1 SUMMARY OF PHASE 2 RESULTS.** The four due date rules examined, in order of increasing sophistication, are: Historical Mean Completion Time, Number of Activities, Critical Path Time, and Scheduled Finish Time. The five resource scheduling/allocation decision rules are: First Come-First Served, Minimum Slack, Minimum Slack Modified by the Due Date, Minimum Late Finish Time, and Minimum Late Finish Time Modified by the Due Date. The two resource preemption policies examined are: Absolute Priority for Resources and Limited Priority for Resources.

Four performance measures are used to evaluate the results of the simulation experiment. Three of the measures, Mean Absolute Deviation of Completion Time, Mean Weighted Lateness, and Mean Development Completion Time, are used to evaluate development project performance. The measure, Mean Maintenance Completion Time, is used to evaluate maintenance project performance.

The performance measure results for development projects are reviewed first. A significant two-way interaction between the due date and resource scheduling rules on the Mean Absolute Deviation of Completion Time measure was found. Although one due date rule does not dominate across all resource scheduling rules, the Scheduled Finish Time due date and First Come-First Served scheduling rule combination produce the best overall performance.

The due date rule factor does not significantly interact with the other factors on the Mean Weighted Lateness Measure. The Scheduled Finish Time due date rule consistently dominates the other due date rules on this measure. The due date rule effect is not statistically significant for the third development project performance measure, the Mean Development Completion Time.

The performance of the development project resource scheduling rules is remarkably consistent on all three performance measures. Within the Absolute Priority for Resources preemption rule, the First Come-First Served scheduling rule is significantly superior in performance. Within the Limited Priority for Resources preemption rule, the order of dominance is consistent on all three performance measures. This order is: Minimum Slack Modified by the Due Date, Minimum Late Finish Time Modified by the Due Date, and First Come-First Served. There is no statistically significant difference among the performance of these sequencing rules.

The discussion now turns to the results for maintenance projects. Within the Absolute Priority for Resources preemption rule, there is no significant effect of either the due date or resource scheduling rules for the Mean Maintenance Completion Time measure. Within the Limited Priority for Resources preemption rule, there is a statistically significant effect on maintenance project duration from the scheduling rules factor. In this case, the First Come-First Served scheduling rule dominates the rest. Under the First Come-First Served rule, maintenance project mean duration only increases by approximately three work-days when the resource preemption policies changes from the absolute priority to the limited priority to maintenance. When other scheduling rules are utilized, maintenance project duration increases markedly.

**7.1.2 UTILIZATION OF THE DUE DATE AND RESOURCE SCHEDULING RULES.** The utilization of any of the due date or resource scheduling rules examined is facilitated by their relative simplicity. Additional requirements are minimal. In using the due date rules, a database of past project performance must be available. This is necessary in order to achieve satisfactory estimates of the "F" or "K" parameter value for due date rule selected. Recall that of the eight MIS departments surveyed in Phase 1 of the research, none had a database available in a form which could be queried through the use of database software. Based on this survey, it appears that MIS departments will incur some database development startup costs if one or more of the due date rules are utilized.

Managers electing to use the Scheduled Finish Time due date rule must also maintain an accurate status of projects which are under development. This presents no problem as most of the MIS departments surveyed in Phase 1 reported that they maintain this information in some type of micro-based project control system. Many of these software packages are capable of performing the finite forward scheduling required by the Scheduled Finish Time rule.

Regarding the implementation of any of the resource scheduling rules, the same project control systems used to track the status of projects can be used to allocate resources to activities.

While the due date and resource scheduling rules are simple to implement, a pertinent and practical question which project managers may ask is: "What rule should be used to schedule resources or estimate development project due dates for a given performance measure?"" To answer this question, Table 7.1 presents a summary of which resource scheduling rules dominate when combined with particular due date estimating rules. Table 7.1 is divided into two columns representing the two resource preemption policies which were examined in this research. This table provides the dominant scheduling rule for each due date rule used. The dominating rule combinations are indicated by (**) for each performance measure.

If managerial policy provides an Absolute Priority for Resources to maintenance projects, then the First Come-First Served scheduling rule should be utilized regardless of which due date rule is selected. The combination of First Come-First Served and Scheduled Finish Time dominate on the Mean Absolute Deviation of Completion Time and Mean Weighted

| | Absolute Priority | Limited Priority |
|---|---|---|
| **Mean Absolute Deviation of Completion Time** | | |
| FLOW: | FCFS | MINSLK[DD] |
| NUMACT: | FCFS | MINSLK[DD] |
| CPTIME: | FCFS | MINSLK[DD] |
| SFT: | FCFS ** | FCFS or MINSLK[DD] ** |
| **Mean Weighted Lateness** | | |
| FLOW: | FCFS | MINSLK[DD] |
| NUMACT: | FCFS | MINSLK[DD] |
| CPTIME: | FCFS | MINSLK[DD] |
| SFT: | FCFS ** | FCFS or MINSLK[DD] ** |
| **Mean Development Completion Time** | | |
| FLOW: | FCFS ** | MINSLK[DD] |
| NUMACT: | FCFS ** | MINSLK[DD] ** |
| CPTIME: | FCFS ** | MINSLK[DD] |
| SFT: | FCFS ** | MINSLK[DD] |
| **Mean Maintenance Completion Time** | | |
| FLOW: | FCFS ** | FCFS |
| NUMACT: | FCFS ** | FCFS |
| CPTIME: | FCFS ** | FCFS |
| SFT: | FCFS ** | FCFS ** |

**Table 7.1 Summary of Due Date, Resource Scheduling and Resource Preemption Rule Performance**

Lateness performance measures. The First Come-First Served scheduling rule may be combined with any of the due date rules when the measure of performance is the effect on development or maintenance project duration.

When management limits the priority for resources to maintenance projects, the Minimum Slack Modified by the Due Date scheduling rule demonstrates consistent dominance. The First Come-First Served rule dominates for the values of "F" and "K" selected for the main experiment when the Scheduled Finish Time due date rule is used. The sensitivity analysis, however, revealed that the Minimum Slack Modified by the Due Date scheduling rule is superior when lower values of "F" and "K" are used in the due date rules. The Scheduled Finish Time due date rule in combination with either of these two scheduling rules appears to provide superior performance.

While the Minimum Slack Modified by the Due Date scheduling rule dominates the development project performance measures, the First Come-First Served scheduling rule performs best for the Mean Maintenance Completion Time measure when a limited priority for resources is given to maintenance. Managers may still elect to use the due date modified Minimum Slack rule since a policy of limiting the priority of resources for maintenance indicates that a degradation of mean maintenance project duration is not significant.

**7.1.3 COMPARISON TO PREVIOUS RESEARCH.** Recall from Chapter 1 that one of the primary objectives of this research is to extend our understanding of the multi-project, capacitated, multi-resource, dynamic, preemptive scheduling problem to include the environment indicative of Management Information System departments. The literature review given in Chapter 2 underscores the lack of research on the dynamic multi-project problem. The only research reported in this area is that of Dumond [DUMO85]. This research is thus an extension of Dumond's work. For this reason, the results obtained in our investigation will be compared primarily to Dumond's work.

Dumond evaluated the effectiveness of the same four due date rules examined in this research. He reported that the Scheduled Finish Time rule performed significantly better on all performance measures. In this research, the Scheduled Finish Time rule also dominates, though the performance is not nearly as superior as that reported by Dumond. The reason for the relative decrease in effectiveness results from the nature of the Scheduled Finish Time rule and the method of selecting activity durations.

The Scheduled Finish Time rule treats the due date estimation problem as a series of static problems by finitely scheduling all known projects. It bases the due date estimate on this schedule. As such, the rule is sensitive to any factor which would cause the schedule to vary when the projects are actually under development. Dumond utilized deterministic activity durations in his research. It is well known from Queueing Theory that deterministic activity durations result in less variance of system performance than do stochastic activity durations [BANK82]. In this research, activity durations were determined by generating a random variate from a Gamma probability distribution. In setting the due date, the Scheduled Finish Time rule utilizes the expected value of the activity duration. As projects are completed, the actual activity durations vary from the expected values used in developing the finite schedules. This adversely affects the performance of the Scheduled Finish Time rule. Despite the decrease in performance, the rule still tends to dominate among those examined.

The results of the effectiveness of the resource scheduling rules for this experiment are remarkably consistent with those reported by Dumond. He found that the First Come-First Served, Minimum Slack Modified by the Due Date, and Minimum Late Finish Time Modified by the Due Date rules dominate. He also found little sensitivity to deviations from the "F" and "K" parameter values used in the due date rules for the Mean Project Completion Time and Standard Deviation of Completion Time performance measures which were used. Finally, Dumond reported that the Scheduled Finish Time and First Come-First Served rules combination tended to outperform or perform equally well with other rule combinations on all performance measures. The results of this experiment are comparable.

From a research standpoint, one has to be quite pleased with the consistency of the results of this research and that of Dumond. The outcome strengthens the degree to which it is possible to generalize the relative effectiveness of the due date and scheduling rules examined. Further, the results tend to support the reports of many researchers in the static single and multiple project scheduling area. The Minimum Slack rule has often been shown to be among the superior resource scheduling rules ([DAVI75], [FEND68], [KURT82]). Both this research and that of Dumond emphasize the necessity to modify the Minimum Slack rule by the due date in a dynamic environment. This modification enables the Minimum Slack rule to use more available information in scheduling resources.

The most surprising result of this experiment is the consistent performance and dominance of the First Come-First Served scheduling rule. While such performance has not been reported in static project management research, the result is consistent with that reported by Dumond.

**7.2 SUGGESTIONS FOR FUTURE RESEARCH.**

While the static single and multiple project management areas have received considerable attention, the dynamic multiple project management area, of which the MIS due date setting problem is an example, has been relatively neglected by researchers. Further, the examination of a heuristic approach to MIS project due date estimation has not been previously reported.

The heuristic approach to due date estimation is replete with potential research questions. One set of questions includes research which would evaluate the potential of other due date, resource scheduling, or resource preemption rules. For example, several due date rules which have demonstrated excellent performance in job shop research may offer potential for application to the MIS due date setting problem. Ragatz and Mabert reported superior performance from the Jobs in Queue and Work in Queue due date heuristics [RAGA84]. The Jobs in Queue heuristic could be modified as an Activities in Queue heuristic for the multiple project environment. This heuristic could utilize information as to the number of activities which are waiting to be processed when estimating the due date for newly arriving projects. The Work in Queue heuristic could utilize information available on the amount of work waiting to be processed by the MIS department. For this heuristic, Work could be defined as the sum of the products of the individual activity expected duration multiplied by their resource requirements.

A second set of research questions concerns the assumptions which underlie this research. For example, it was assumed that the MIS department established all development project due dates internally. In many operational settings, various proportions of project due dates may be established externally. Although Dumond examined this question for a general class of project types, his results may not be generalizable to the MIS area. Another assumption of this research was that overtime would not be utilized. In fact, a small percentage of MIS departments may use overtime extensively, and all MIS departments use overtime to some degree. The effect of contracting a proportion of arriving projects could also be examined.

Another major assumption underlying this research concerns the system development methodology. It was assumed that a life cycle development methodology is utilized for all development projects. Other development methodologies such as the Prototyping approach, use of Fourth Generation Languages, joint user and MIS department project development methodologies, etc, could be investigated.

Still another area for research concerns modeling the project stream representing the work requirements. In this research, two types of projects, development and maintenance, were utilized. It would be possible to split the maintenance project type into multiple classes of maintenance. Some of these classes would require immediate resource allocation, such as when an important system becomes inoperative. Other classes could have maintenance requirements delayed, thereby modeling requests for minor changes to existing systems. The relative effectiveness of resource scheduling rules might vary with a change in the project stream model.

A very interesting research topic concerns the effect of the resource preemption policy. In this research, the Absolute Priority for Resources rule provides priority to maintenance activities. The Limited Priority for Resources rule provides priority to development activities, but only if the development activity possesses negative slack. These two rules produce significantly different effects on the mean duration of development and maintenance projects.

Under the Absolute Priority for Resources rule, the lowest mean development project duration was approximately 83 days when resources were scheduled using the First Come-First Served scheduling rule. Under the Limited Priority for Resources rule, the mean development project duration decreased to approximately 68 days when resources were scheduled using the Minimum Slack Modified by the Due Date scheduling rule. In contrast, the mean maintenance project duration is approximately five days under the Absolute Priority for Resources policy for all of the scheduling rules examined. However, when priority to maintenance is limited, the mean maintenance project duration increases to approximately 31 days. This six-fold increase occurs when the Minimum Slack Modified by the Due Date scheduling rule is used. This poor performance for maintenance projects relative to the First Come-First Served rule may result in managers rejecting use of the due date modified Minimum Slack rule.

The mixing of maintenance and development project requirements for project design teams is a significant organizational research issue. In the past, many MIS departments were organized with separate maintenance sections. The trend, as evidenced by the results of Phase 1 of this research, is to combine development and maintenance by allocating workloads to design teams which are aligned with functional areas. The primary argument for the combined organization is that the designers of the systems will also maintain the systems. The familiarity achieved should increase designer productivity on maintenance work. Some firms, however, are experiencing satisfactory results with a separate maintenance section organization. Removing maintenance requirements from design teams may significantly increase the quality of the systems produced and reduce many of the managerial problems which arise in the combined organizational approach. This is an excellent field research area because the impact of maintenance requirements on MIS departments is a significant managerial issue.

Some of the questions surrounding this issue may be amenable to research using computer simulation. A modeling approach would be to eliminate maintenance projects from the project job stream and model just the development project design teams. This research could also be combined with efforts to develop an MIS model which captures the effects of additional operational factors and variables.

## 7.3 SUMMARY OF CONTRIBUTIONS.

In summary, this research made two significant contributions. First, this research has documented a model of the variables and factors which impact the MIS manager's ability to set good project due dates. Such a systems development model was previously lacking in the MIS literature.

Second, it has provided an increase in the generalizability of the due date and resource scheduling rules examined. In particular, the project network shapes, the use of stochastic activity durations, the inclusion of activity rework loops, the modeling of a project stream which includes two different types of projects, development and maintenance, and the addition of resource preemption considerations provide an experimental environment which is markedly different from that examined by previous researchers. Even though the experimental environment was quite different, the results of this research are very similar to and support the results reported in previous research.

# APPENDIX A

## A.0 PROGRAM STRUCTURE, VALIDATION, AND VERIFICATION.

This appendix is divided into two sections. Section A.1 gives a description of characteristics of the simulation program and discusses the nature of "events" in discrete event simulation modeling. This section also provides a detailed discussion of the hierarchical structure of the simulation program developed. Section A.2 describes the validation of simulation model used and the verification of the accuracy of the simulation program output.

## A.1 SIMULATION PROGRAM CHARACTERISTICS, EVENTS, AND STRUCTURE.

**A.1.1 PROGRAM CHARACTERISTICS.** The computer program that performs the simulation of MIS department activities was written in FORTRAN 77 and consists of 1508 lines of source code and 700 lines of documentation comments. The program was compiled and run on an IBM 4381 computer. The program requires 76,084 bytes of memory. It uses a discrete event simulation approach. FORTRAN 77 was selected over other languages which are available for discrete event simulation modeling because it provides a greater degree of research control and portability.

Recall that the 4x5x2, three-factor, full-factorial research design requires 20 observations of the measures of performance per cell. The program main routine is designed to run the simulation 20 times. This facilitates collecting all of the observations needed for a single cell of the research design with a single batch job submission. Program run time per cell varies from a low of approximately 700 Central Processor Unit (CPU) seconds to a high of 10,600 CPU seconds. The cell run time is a function of several factors. It is dependent on the exact combination of Due Date, Scheduling, and Resource Preemption heuristics which are examined in the different cells. It also depends on the number of observations of the performance measures (dependent variables) which are collected for an individual cell run. Finally, the cell run time varies with the size of the project stream. The current limit on the number of projects for one observation is 1000 projects. This equates to approximately 750 work-days of simulation time.

**A.1.2 EVENTS IN THE SIMULATION.** Prior to describing the hierarchical structure of the program, a brief explanation of the discrete event simulation approach is in order. When modeling a particular operational setting, such as computer based system development, it is necessary to identify the "events" which can occur within the setting. Events represent "discrete" points in time when the status of a system changes.

In the simulation program there are three possible events. The first two events actually occur in operational systems. These include a "project arrives" event and an "activity is completed" event. Since all of the various requirements in which MIS departments engage can be categorized as "project" types, the arrival of any requirement can be modeled as a project. In turn, the project can be modeled as a network of activities and the activities can be assigned to members of a design team for completion. The completion of an activity event can also be modeled. The completion of a project is a special case of the activity completion event. In this case, the event happens to represent the last activity in a project. As these two types of events occur, several changes in the state of the system take place. For example, when an activity is completed, resources are released by the activity. These resources may, in turn, be allocated to activities which are awaiting resources.

Since the events represent discrete points in time, the state of the simulation system does not change between the occurrence of events. It is at these points in time when statistics concerning system performance are accumulated for later analysis.

The third event, which does not occur in operational settings, is the termination of the simulation. This event signals the program to compute and record the statistics which represent one observation of the measures which are later used to evaluate various aspects of system performance.

The purpose of this short discussion is to emphasize that modeling the events which occur in operational settings

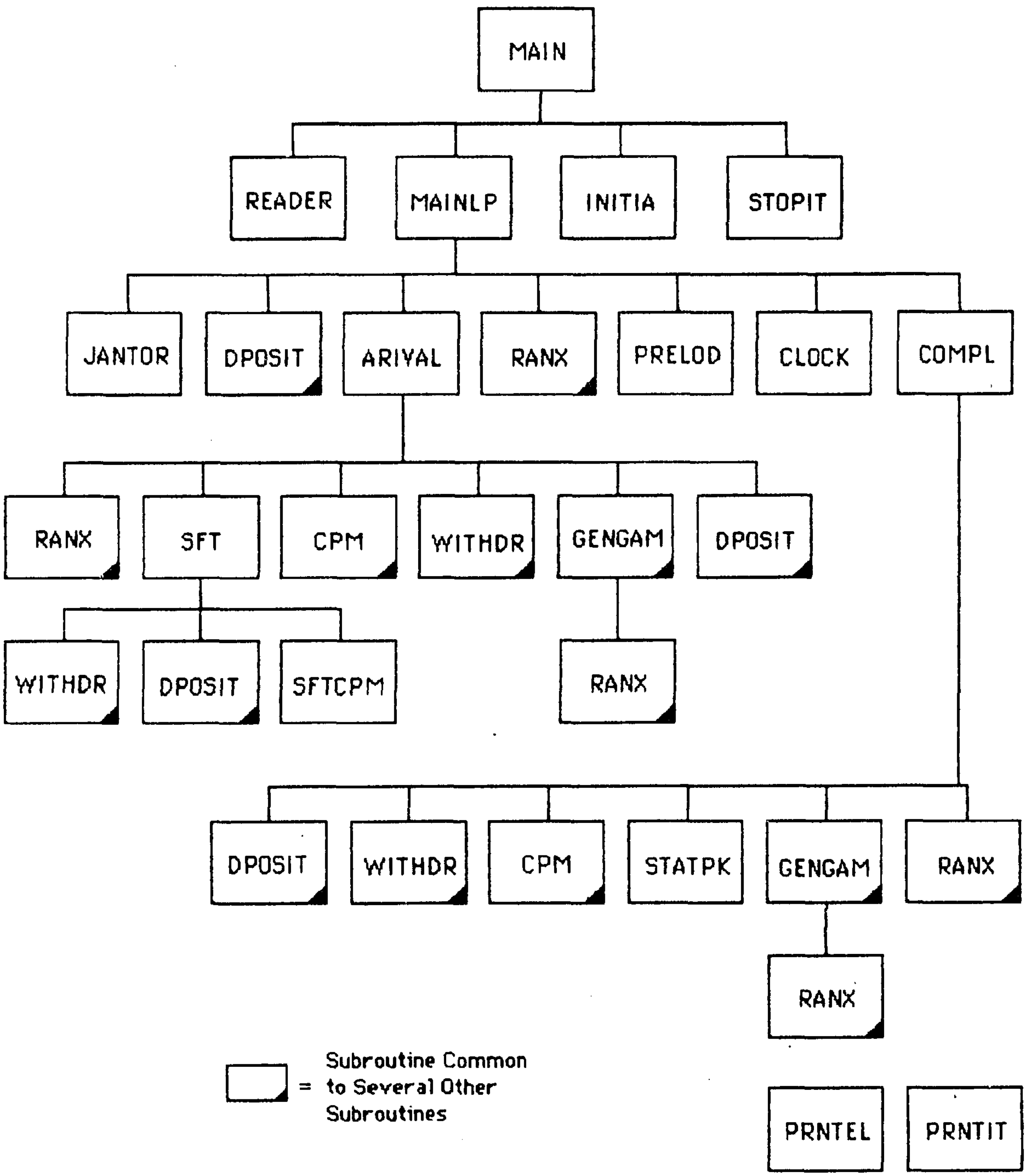



Figure A.1 Program Hierarchy Chart

is central to the development of a useful simulation model. In this program, the two events described above, project arrival and activity completion, form the central core of the simulation model. For a more detailed explanation of the discrete event simulation modeling approach, see [BANK84] or [LAW82].

**A.1.3 A DETAILED DESCRIPTION OF THE PROGRAM HIERARCHY.** The simulation program was written using a structured, modularized coding approach. Figure A.1 shows the hierarchy of the simulation program structure. The simulation program consists of a main scheduling program and 19 subroutines. Two of the subroutines, PRNTEL and PRNTIT, are not integral components in the logical design. They are used to perform debugging and trace functions during program verification. Table A.1 lists the subroutines in five categories and provides a brief explanation of their purpose. A detailed explanation of the subroutines is given below.

**A.1.3.1 CATEGORY 1: MAIN PROGRAM AND ANCILLARY SUBROUTINES.** The MAIN program performs the overall control function for the simulation. It performs several initialization and final processing functions to include opening input and output files, setting initial seed values for the random number generator, initializing statistical accumulators, printing a summary report of system performance, etc. These functions are only performed one time per batch execution of the simulation. The MAIN program calls four subroutines, READER, MAINLP, INITIA, and STOPIT.

### CATEGORY 1: MAIN PROGRAM & ANCILLARY SUBROUTINES

| | |
|---|---|
| MAIN | Main program level of control. |
| READER | Load input data files. |
| MAINLP | Control which Event subroutine executes next. |
| INITIA | Perform most of the initialization of program variables for an observation run. |
| STOPIT | Compute and write performance measure values to output files. |
| PRELOD | Preload the system to decrease the transient state of system operation. |

### CATEGORY 2: EVENT SUBROUTINES

| | |
|---|---|
| COMPL | Completion of an activity event. |
| ARIVAL | Arrival of a project event. |

### CATEGORY 3: SFT HEURISTIC SUBROUTINES

| | |
|---|---|
| SFT | Scheduled Finish Time heuristic routine. |
| SFTCPM | Compute critical path data on projects for the SFT routine. |

### CATEGORY 4: MISCELLANEOUS SUBROUTINES

| | |
|---|---|
| STATPK | Collection of performance measure statistics. |
| CPM | Compute critical path data on projects. |
| RANX | Generate uniform distributed random numbers. |
| GENGAM | Generate gamma distributed random variates. |
| PRNTEL | Print variable values for testing & debugging. |
| PRNTIT | Print activity queue states for testing & debugging. |

### CATEGORY 5: DATA MANAGEMENT SUBROUTINES

| | |
|---|---|
| JANTOR | Initialize bidirectional linked list code used to store activity data. |
| DPOSIT | Linked list insertion procedure. |
| WITHDR | Linked list removal procedure. |
| CLOCK | Control advancement of the simulation from one discrete event to the next. |

**Table A.1 Program Main Routine & Subroutines**

Subroutine READER obtains input data from three different data files. The first input data file, PROJECT DATA, contains detailed information about project networks which represent the work requirements to be assigned to the MIS department. This information includes resource requirements, predecessor and successor relationships, expected duration times, etc. The second input data file, JS DATA, contains a series of numbers representing the various types of development and maintenance projects that form the job stream for the simulation. The third input data file, SETUP DATA, contains parameter values which establish the number of observations required per research design cell, the number of projects to arrive to the simulator prior to recording performance measure statistics for a single observation, values for the "F" and "K" parameters which are integral components in the Due Date heuristics, and parameters which set the Due Date, Scheduling, and Resource Preemption heuristics to be used for a set of observations.

The primary control function of the main program is performed by calling the three subroutines, INITIA, MAINLP, and STOPIT. These subroutines are called in a loop that executes once for each observation of the performance measure statistics that is required.

Subroutine MAINLP (main loop) controls the operation of the simulation for one observation of the performance measure statistics. As is indicated in Figure A.1, this subroutine, in turn, calls a series of other subroutines. These include CLOCK, JANTOR, DPOSIT, ARIVAL, RANX, PRELOD, and COMPL.

MAINLP calls subroutine PRELOD to preload the simulation "system" with a set of projects. Preloading is performed using deterministic activity duration times. This guarantees that the same preloading occurs for each observation. The purpose of the preloading is to assist the simulation in reaching "steady state" conditions by eliminating most of the "transient state" time period. The simulation actually reaches "steady state" after approximately 30 projects have arrived in the simulation. This was verified by observing the time-average number of projects in process and average resource utilization at discrete points in the simulation.

MAINLP next calls subroutine CLOCK to obtain the type of event, i.e. project arrival, activity completion, or simulation termination, which will occur next. MAINLP updates the resource utilization matrix and calls either ARIVAL (arrival) or COMPL (completion) as appropriate. On termination of a simulation observation, control is returned to the MAIN program routine.

Subroutine INITIA (initialization) simply initializes variables for each observation of the performance measure statistics which are collected.

Subroutine STOPIT computes the ending values for the performance measure statistics and writes the information to three output files. One output file is a report which provides statistics describing the results for a set of observations. The second output file is produced for later use as an input data file to programs such as the Statistical Program for the Social Sciences for research analysis purposes [SPSS86]. The final output file is only produced during program validation and output verification procedures.

**A.1.3.2 CATEGORY 2: EVENT SUBROUTINES.** The two subroutines in this category are COMPL (completion of an activity) and ARIVAL (arrival of a project). Prior to discussing the event subroutines, it is necessary to describe the methods used to track activities in projects.

Project activities can exist in one of three states. First an activity may be work-in-process. In this case, the activity and its relevant attributes are stored in a "Work-In-Process" queue which is managed as a bidirectional linked list. The activities are maintained in ascending order within this list according to the scheduled time of activity completion. Second, an activity may have all predecessor requirements completed, but resources may not be available for allocation to this activity. Activities in this category are stored in a "Ready-For-Resources" queue based upon their priority index value attribute which is computed using the scheduling heuristic being examined in a particular set of simulation observations. This queue is also a bidirectional linked list. Finally, activities which do not have their predecessor requirements completed are stored in a "Wait" queue array. To reduce the storage requirements of the "Wait" queue array and eliminate unnecessary resorting of data when a project is completed, a pointer array and available space stack array are utilized to facilitate data management.

When an activity is completed in subroutine COMPL, three courses of action are possible. First, an examination of the work may result in the the activity being reworked. If this is the case, then the activity is reloaded into the "Work-In-Process" queue with a new activity duration and scheduled event completion time to simulate the rework requirement.

Second, an examination of the activity may result in the decision to rework a group of related activities. To simulate reworking a group of activities, the first activity in the group is reloaded in the "Work-In-Process" queue. The other activities are reloaded in the "Wait" queue pending rework of the unsatisfactory predecessor activities.

Third, the activity may be found to have been satisfactorily completed. In this case the activity is completed by removal from the "Work-In-Process" queue. The resources assigned to the completed activity are reallocated to activities in the "Ready-For-Resources" queue. The remainder of the COMPL subroutine updates the status of activity predecessor requirements and moves activities from the

"Wait" queue to the "Ready-For-Resources" queue and finally to the "Work-In-Process" queue.

A special case of activity completion arises whenever the activity completed is the last one in a project. The only additional processing required is a call to subroutine STATPK to collect performance statistics and the maintenance of the "Wait" queue pointer and available space stack arrays.

Subroutine ARIVAL processes the event called "arrival of a project" to the MIS department. The new project may be either a maintenance or a development project. Regardless of the project type, the subroutine assigns the project to a design team and assigns a project number for tracking purposes. The "Wait" queue pointer and available space stack arrays are updated to allow the loading of activities from the new project in the "Wait" queue. The subroutine also schedules the arrival of the next project and stores this event in the "Work-In-Process" queue. This queue also functions as the simulation event list.

The remainder of the ARIVAL processing is distinctly different for development and maintenance projects. If the project is developmental, an examination of available team resources is made to determine whether the first activity will be placed in the "Work-In-Process" or "Ready-For-Resources" queue. Project due date and activity priority index values are also computed. The remaining activities in the project are loaded in the "Wait" queue.

When the new project represents a maintenance requirement, a determination is again made on whether or not resources are available for the first activity of the project. If resources are not available, resources may be preempted from development activities in the "Work-In-Process" queue based on the resource preemption heuristic in use for the simulation run. If resources are not available through preemption, the first activity is placed in a first come-first served "Ready-For-Resources" maintenance queue. Maintenance activities are sequenced in this linked list queue based on their time of arrival in the simulation. If the maintenance project consists of multiple activities, the remainder of the activities are loaded to the "Wait" queue. If a development activity is preempted, the remaining development activity duration is updated and the activity is moved from the "Work-In-Process" queue to the "Ready-For-Resources" development queue.

**A.1.3.3 CATEGORY 3: SFT HEURISTIC SUBROUTINES.** This category includes the subroutines required to accomplish due date estimation using the Scheduled Finish Time due date heuristic. All other due date heuristics are simple enough to not require a separate compute program.

Subroutine SFT estimates a project due date by finitely scheduling all existing project activities for a team. To accomplish this, the subroutine copies all activities which have not been completed, i.e. those in the "Work-In-Process", "Ready-For-Resources", and "Wait" queues for the team assigned responsibility for the new project, to a set of temporary queues which are termed "SFT" queues. The new project also has its activities loaded into these SFT queues. The simulation runs these activities using expected activity durations to schedule events in the "Work-In-Process" queue. Subroutine SFTCPM is called to update critical path data on the activities in the temporary queues. Whenever the last activity of the new project is completed, the project due date is estimated and control returns to subroutine ARIVAL.

**A.1.3.4 CATEGORY 4: MISCELLANEOUS SUBROUTINES.** The first subroutine in this category is STATPK. This subroutine simply accumulates values which are required for later computation of the performance measure statistics representing the conduct of a simulation observation run.

Subroutine CPM computes critical path data on projects. It is called by several other subroutines. Whenever an activity is moved to the "Ready-For-Resources" queue, CPM is called to update the activity priority index value.

Subroutines RANX and GENGAM are used to generate the stochastic random numbers and random variates required throughout the simulation. They are called by several subroutines as indicated in Figure A.1. Subroutine RANX generates uniform [0,1] random numbers and is included to enhance the portability of the program. Subroutine GENGAM generates gamma distributed random variates from different gamma distributions. The scale parameter, beta, is fixed at one. The shape parameter, alpha, is passed to the subroutine as the expected value of the duration of the activity. All values for alpha are greater than or equal to unity. The random variates are generated using an acceptance-rejection method due to Cheng [CHEN77] as described by Law and Kelton [LAW82, pgs 255-258].

Subroutines PRNTEL and PRNTIT are used for program debugging and verification. The Debug facility provided with FORTRAN 77 could not be used to examine the values of variables representing the state of the system at discrete points in time because the number of variables represented by the program arrays was extremely large and prohibited such a volume of output. These two subroutines can be called at any point in the program to provide the trace functions necessary to verify status of the program queues and array variables. As such, they are are not logical components in the hierarchical structure of the program.

**A.1.3.5 CATEGORY 5: DATA MANAGEMENT SUBROUTINES.** The management of data within a simulation program can become a prodigious task. This is particularly true when one considers the requirement to track simulation events, such as the arrival of projects and the completion of activities, and insure that they occur in the proper time-order sequence. Two approaches are available for data management. One approach is to sort the data to maintain an "array of events" in the proper time-order sequence. However, in a simulation program such as the one developed in this research, the computer CPU time required for sorting may become prohibitive.

The approach selected for this simulation program maintains the event data in linked-list arrays. As events are scheduled, they can be inserted in their proper time-order sequence by the use of linked-list insertion algorithms. Several general simulation languages are based on this approach. One such language is called ASL. ASL is an acronym for "A Simulation Language". ASL consists of a set of subroutines written in FORTRAN 77 that facilitate the management of data. ASL is very similar to the SIMLIB simulation language [LAW82, Ch 2]. ASL was developed by Dr. Ashok Soni, Professor of Decision Sciences, Graduate School of Business, Indiana University. It is used with his permission and is not available in a published media. The ASL subroutines enhance a researcher's ability to develop simulation models by providing the code necessary to manage event data storage. This relieves the researcher of the task of writing and debugging the code necessary to perform the various data management functions.

Subroutine JANTOR (janitor) initializes all of the variables and arrays that exist in ASL. This includes a TIME variable which is used to track simulation time, and a 1000 row by 10 column array called the VAULT which is organized and managed by the use of bidirectional linked list code. The VAULT can be used to store up to 10 files or queues. File 10 is always the event list which this simulation also uses as a "Work-In-Process" queue.

Subroutine DPOSIT and WITHDR are insertion and deletion procedures, respectively. Entries are inserted or deleted into or from the VAULT array by the use of a 10-element transfer array called the TELLER. DPOSIT can be used to insert an entry into a file in the VAULT as either the first or last entry in the linked list, or in the middle of the list so as to maintain ascending or descending order. WITHDR can withdraw either the first or last entry from a linked list as desired.

Subroutine CLOCK is a timing routine which calls WITHDR to obtain the next event from the event file. CLOCK is called by subroutine MAINLP.

## A.2 VALIDATION AND VERIFICATION OF PROGRAM ACCURACY.

Several articles and texts are available which describe suggested procedures for validating and verifying simulation models. The validation procedure used in this research is described in a well known Management Science article by Naylor and Finger [NAYL67]. The verification procedure is described by Law and Kelton in their text, Simulation Modeling and Analysis [LAW82]. Validation is defined as "determining whether a simulation model (as opposed to a computer program) is an accurate representation" of the operational setting being modeled while Verification is defined as "determining whether a simulation model performs as intended, i.e. debugging the computer program." [LAW82, pgs 333-334].

**A.2.1 VALIDATION PROCEDURE.** A useful approach to validating a simulation model is attributed to Naylor and Finger [NAYL67]. Law and Kelton augmented this approach by providing specific recommendations on performing the validation procedure [LAW82, Ch 12]. The approach consists of three steps which are: (1) Develop a model with high face validity; (2) Test the assumptions of the model empirically; and (3) Determine how representative the simulation output data are.

One objective of this research is the development of a computer based systems development model which possesses a high degree of generalizability and face validity. The majority of the effort required to accomplish this objective is described in Chapters 3 and 4. The model development procedure is reviewed here to demonstrate how it relates to step one in the approach prescribed by Naylor and Finger. Specific techniques which Law and Kelton recommend during the first step of the validation procedure include: (1) conversing with "experts"; (2) utilizing existing theory; (3) relying on general knowledge; and (4) observing an operational system.

Early in the research, it was necessary to determine whether existing theory or general research knowledge could provide an adequate comprehensive descriptive model of computer based systems development. To this end a review of literature was conducted. While a comprehensive model was not found, the review was helpful in the initial formulation of such a model. The references most helpful included [DICK85], [DUMO85], [MART85], and [SPRA86].

Further refinement of the initial model was accomplished in two ways. First, discussions with members of the academic community were conducted. This incorporates the idea of "conversing with experts." Second, the literature review was broadened to include the examination of a number of practitioner journals and periodicals. The objective of this review was to further modify the computer based systems development model by incorporating the ideas and results reported by MIS practitioners. The periodicals reviewed included Communications of the ACM, Computer Decisions, Data Management, Datamation, EDP Analyzer, Information and Management, Journal of Systems Management, MIS Quarterly, and The Management of Information Systems.

The final refinement of the systems development model was completed by conducting a series of interviews with MIS project managers and their supervisors in different MIS departments. Conducting interviews combines ideas related to the suggestion to "converse with experts" and "observe an operational system". While the MIS project managers are not necessarily expert in all approaches to systems development, they are experts on the methodology utilized within their own MIS departments. Although operational systems were not directly observed, observations were collected from MIS managers in several operational MIS departments. This provided the face validity needed for the specification of a single MIS department model which is generally representative of a large class of MIS departments.

The recommendations by Law and Kelton are less specific for the second step in model validation: "testing the assumptions of the model empirically." Perhaps the most useful tool in this area is the conduct of a "sensitivity analysis". This technique was used to evaluate the effect of varying the "F" and "K" input parameters which are integral components of the due date rules examined in the research. The results of the sensitivity analysis are reported in Chapter 6.

There are additional assumptions which were made in modeling the systems development process. For example, it was implicitly assumed that the project networks used to represent the department workloads are generally representative of those projects which MIS departments must complete. In the validation process, it is often difficult or impossible to empirically test many of these additional assumptions. Rather, one must rely on intuition. The researcher must question the effect of these assumptions on the validity of the simulation model by asking: "Are the differences between the operational system and the model significant enough to affect any conclusions derived from the model?" [LAW82, pg 341]. This question of generalizability is discussed in Chapter 7.

The third step in validating the simulation model concerns the representativeness of the simulation output data to that which would be produced from an actual operational system. The use of classical statistical procedures for comparing the simulation output to operational systems output is ruled out in this research because there are no operational systems data available for the comparisons. Law and Kelton point out that this is often the case in simulation experiments.

The alternative approach to output data analysis selected for this research is the use of the "expert" option and researcher intuition as to whether the output data are reasonable. "Expert" opinions on the validity of some of the output data were obtained during the Phase 1 interviews. For example, MIS project managers were asked questions such as: What resource utilization percentage is expected in an operational MIS department? MIS managers generally agreed that system designers were occupied with activities not related to the completion of project work approximately 15 to 20 percent of the time. These "other" activities included attendance at training classes and seminars, vacations, informal meetings, breaks, etc. Based on this response, the project arrival rate was set such that mean resource utilization was approximately 82 to 83 percent. In this example, the fact that the simulation produces an 82 percent resource utilization rate does not guarantee the validity of the other simulation model output data. It does, however, provide additional confidence in the model's output data.

**A.2.2 PROGRAM VERIFICATION.** Program verification was accomplished using several techniques suggested by Law and Kelton [LAW82, Ch 10]. One such technique is "stub" testing. Stub testing requires the development of a well structured logical design prior to actually writing code. The subroutines, or "stubs", were dummy coded into the overall structure of the program. As the program subroutines were added, they were tested for accuracy. For example, the first subroutines added were those which read the data and print the status of the queues in the system. The READER subroutine was added and checked to insure that the data read were stored correctly in the various arrays. As another example, subroutine CPM was written and tested separately prior to adding it to the simulation program to insure that critical path computations were correct. After merging it into the simulation program, CPM was again tested to insure that the correct array data entries were used in the critical path calculations.

A second technique recommended is a program "trace". Subroutines PRNTEL and PRNTIT were written to allow projects and activities to be traced during the actual conduct of the program testing. This allowed a check of the program's performance during "unusual" conditions such as start up, arrival of a project when no resources were available, etc.

Finally, the program was run and tested under simplified conditions. During initial program runs, deterministic input data were used to insure that of the Due Date, Scheduling, and Resource Preemption heuristics were performing as required. The use of deterministic input data also facilitated the verification of the program output data.

## BIBLIOGRAPHY

[APPL86a] Appleton, David S., "Very Large Projects," Datamation, Vol 30, No 2, (Jan 86), 63-70.

[APPL86b] Appleton, David S., "Information Asset Management," Datamation, Vol 30, No 3, (Feb 86), 71-80.

[ARCH68] Archibald, R.D. & R.L. Villoria, Network-Based Management Systems (PERT/CPM), John Wiley, (1968).

[BAKE72] Baker, F.T., "Chief Programmer Teams, Management of Production Programming," IBM Systems Journal, Vol 11, No 1, (Mar 72), 56-73.

[BALL77] Bally, Laurent, John Brittan, & Karl H. Wagner, "A Prototype Approach to Information System Design and Development," Information & Management, (Jan 77), 21-26.

[BANK84] Banks, Jerry & John S. Carson, Discrete-Event System Simulation, Prentice Hall, Englewood Cliffs, New Jersey, (1984), 514 pages.

[BART86] Bartol, Kathryn M., David C. Martin, & Anne M. Lyons, "Human Resources Management: A Growing Necessity for DP Managers," Journal of Systems Management, Vol 37, No 1, (Jan 86), 32-36.

[BENB80] Benbasat, Izak & Iris Vessey, "Programmer and Analyst Time/Cost Estimation," MIS Quarterly, Vol 4, No 7, (Jun 80), 31-43.

[BOWM59] Bowman, E.H., "The Schedule-Sequencing Problem," Operations Research, Vol 7, No 5,, (Sep-Oct 59), 621-624.

[BOX54] Box, G.E.P., "Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems, I. Effect of Inequality of Variance in the One-Way Classification," Annals of Mathematical Statistics, Vol 25, (1954), 484-498.

[BROO74] Brooks, Frederick P. Jr., "The Mythical Man-Month," Datamation, Vol 20, No 12, (Dec 74), 45-52.

[BURT75] Burton, B.J. "Manpower Estimating for Systems Projects," Journal of Systems Management, Vol 26, No 1, (Jan 75), 29-33.

[BYRK81] Byrkit, Donald R. & Shawky E. Shamma, Calculus for Business and Economics, D. Van Nostrand Company, (1981), 372 pages.

[CANN77] Canning, Richard G., "Progress in Project Management," EDP Analyzer, Vol 15, No 12, (Dec 77), 1-11.

[CHEN77] Cheng, R.C.H., "The Generation of Gamma Variables with Non-integral Shape Parameter," Applied Statistics, Vol 26, (1977), 71-75.

[CHEN78] Chen, E.T. "Program Complexity and Programmer Productivity," IEEE Transactions on Software Engineering, Vol SE-4, No 3, (May 78), 187-194.

[CHRI81] Christensen, K., G.P. Fitsos, & C.P. Smith, "A Perspective on Software Science," IBM Systems Journal, Vol 20, No 4, (1981), 372-387.

[CHRY78a] Chrysler, Earl, "Some Basic Determinants of Computer Programming Productivity," Communications of the ACM, Vol 21, No 6, (Jun 78), 472-483.

[CHRY78b] Chrysler, Earl, "Programmer Performance Standards," Journal of Systems Management, Vol 29, No 2, (Feb 78), 18-25.

[CHUR76] Church, J.D. & E.L. Wike, "The Robustness of Homogeneity of Variance Tests for Asymmetric Distributions: A Monte Carlo Study," Bulletin of Psychonomic Society, Vol 7, (1976), 417-420.

[CONW74] Conway, R.W. "An Experimental Investigation of Priority Assignment in a Job Shop," Memorandum RM-3789-PR, The Rand Corporation, Santa Monica, CA (Feb 74).

[COOP76] Cooper, D.F. "Heuristics for Scheduling Resource-Constrained Projects: An Experimental Investigation," Management Science, Vol 22, No 11, (Jul 76), 1186-1194.

[CROS79] Crossman, Trevor D., "Taking the Measure of Programmer Productivity," Datamation, Vol 25, No 5, (May 79), 144-147.

[DAVI66] Davis, Edward W., "Resource Allocation in Project Network Models-A Survey," The Journal of Industrial Engineering, Vol 17, No 4, (Apr 66), 177-188.

[DAVI73] Davis, Edward W., "Project Scheduling Under Resource Constraints-Historical Review and Categorization of Procedures," AIIE Transactions, Vol 5, No 4, (Dec 73), 297-313.

[DAVI75] Davis, Edward W. & James H. Patterson, "A Comparison of Heuristic and Optimum Solutions in Resource-Constrained Project Scheduling," Management Science, Vol 21, No 8, (Apr 75), 944-955.

[DICK85] Dickson, Gary W. & James C. Wetherbe, The Management of Information Systems, McGraw-Hill, New York (1985), 493 pages.

[DONE76] Donelson, W.S., "Project Planning and Controls," Datamation, Vol 22, No 6, (Jun 76), 73-80.

[DUMO85] Dumond, John, Evaluation of Due Date Assignment Rules for Dynamically Arriving Projects, Unpublished Doctoral Dissertation, Indiana University, (Feb 85), 222 pages.

[DUMO84] Dumond, John & Vincent A. Mabert, "Evaluating Project Scheduling and Due Date Assignment Procedures: An Experimental Analysis," Working Paper, Indiana University, (circa 84).

[FEND68] Fendley, L.G., "Toward the Development of a Complete Multiproject Scheduling System," Journal of Industrial Engineering, Vol 19, No 10, (Oct 68), 505-515.

[FILL86] Filley, Richard D., "1986 Project Management Software Buyer's Guide," Industrial Engineering, Vol 18, No 1, (Jan 86), 51-63.

[GAYL71] Gayle, J.B., "Multiple Regression Techniques for Estimating Computer Programming Costs," Journal of Systems Management, Vol 22, No 2, (Feb 71), 13-16.

[GUEN64] Guenther, William C., Analysis of Variance, Prentice-Hall (1964), Englewood Cliffs, N.J., 199 pgs.

[HARR83] Harrell, Clayton Jr., Ralph L. Kliem, & Anthony J. Viscomi, "Sure-Handed Project Management: Parts I, II, & III," Computer Decisions, (Nov 83), 261-276.

[HENN69] Henney, A., "Techniques for Estimating Programming Effort and Assessing Programmer Performance," Data Systems, (Sep 69), 34-36.

[HOLL79] Holloway, C.A., R.T. Nelson & V. Suraphongschai, "Comparison of Multi-Pass Heuristic Decomposition Procedure with Other Resource-Constrained Project Scheduling Procedures," Management Science, Vol 25, No 9, (Sep 79), 862-872.

[HUCK74] Huck, Schuyler, William Cormier, & William G. Bounds, Reading Statistics and Research, Harper & Row, New York, NY, (1974), 387 pages.

[JOHN77] Johnson, J.R., "A Working Measure of Productivity," Datamation, Vol 23, No 2, (Feb 77), 106-110.

[KELL59] Kelley, E. & M. Walker, "Critical Path Planning and Scheduling," Proceedings of the Eastern Joint Computer Conference, (1959).

[KEPP82] Keppel, Geoffrey, Design and Analysis: A Researcher's Handbook, Prentice-Hall, Englewood Cliffs, New Jersey, (1982), 669 pages.

[KURT82] Kurtulus, I. & E.W. Davis, "Multi-Project Scheduling: Categorization of Heuristic Rules Performance," Management Science, Vol 28, No 2, (Feb 82), 161-172.

[LAW82] Law, Averill M. & W. David Kelton, Simulation Modeling and Analysis, McGraw-Hill, New York, (1982), 400 pages.

[MALC59] Malcolm, D.G., J.H. Rosenblom, C.E. Clark, & W. Frazer, "Application of a Technique for Research and Development Program Evaluation," Operations Research, Vol 7, (1959), 616-669.

[MANN60] Manne, A.S., "On the Job-Shop Scheduling Problem," Operations Research, Vol 8, No 2, (Mar-Apr 60), 219-223.

[MARK85] Marks, William W., "Teamwork Keeps Project Workloads Flying High," Data Management, Vol 23, No 8, (Aug 85), 12-17.

[MART77] Martin, C.G. & P.A. Games, "ANOVA Tests for Homogeneity of Variance: Non-Normality and Unequal Samples," Journal of Educational Statistics, Vol 2, (1977), 187-206.

[MART85] Martin, James, Fourth-Generation Languages, Prentice-Hall, New Jersey, (1985).

[MERE85] Meredeth, Denis C., "System Development Methodologies Provide Project Fire Power," Data Management, Vol 23, No 8, (Aug 85), 22-25.

[MILL86] Miller, Rupert G., Beyond ANOVA, Basics of Applied Statistics, John Wiley & Sons, New York, (1986), 317 pages.

[MIZE64] Mize, J.H., A Heuristic Scheduling Model for Multi-Project Organizations, Unpublished Doctoral Dissertation, Purdue University, (Aug 64).

[MODE70] Moder, J.J. & C.R. Phillips, Project Management with CPM and PERT, (2d Ed), Van Nostrand-Reinhold, 1970.

[MONT84] Montgomery, Douglas C., Design and Analysis of Experiments, (2d Ed), John Wiley & Sons, New York, 1986, 538 pages.

[MYER78] Myers, W., "A Statistical Approach to Scheduling Software Development," Computer, Vol 11, No 12, (Dec 78), 23-35.

[NAYL67] Naylor, T.H. and J.M. Finger, "Verification of Computer Simulation Models," Management Science, Vol 14, No 2, (1967), 92-101.

[NORD70] Norden, P.V., "Useful Tools for Project Management," Management of Production, M.K. Starr (ed), Penguin Books Inc, Baltimore, MD, (1970), 71-101.

[PATT76] Patterson, James H., "Project Scheduling: The Effects of Problem Structure of Heuristic Performance," Naval Research Logistics Quarterly, Vol 23, No 1, (Mar 76), 95-123.

[PATT73] Patterson, James H., "Alternative Methods of Project Scheduling with Limited Resources," Naval Research Logistics Quarterly, Vol 20, No 4, (Dec 73), 767-784.

[PETE79] Petersen, Perry, "Project Control Systems," Datamation, Vol 25, No 7, (Jun 79), 147-162.

[PRIT69] Pritsker, A. Alan B., Lawrence J. Watters, & Philip M. Wolfe, "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach," Management Science, Vol 16, No 1, (Sep 69), 93-108.

[PUTN78] Putnam, L.H., "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," IEEE Transactions on Software Engineering, Vol SE-4, No 4, (Jul 78), 345-361.

[RAGA84] Ragatz, Gary L., & Vincent A. Mabert, "A Simulation Analysis of Due Date Assignment Rules," Journal of Operations Management, Vol 5, No 1, (Nov 84), 1-13.

[ROGA77] Rogan, J.C. & H.J. Keselman, "Is the ANOVA F-Test Robust to Variance Heterogeneity Where Samples Sizes Are Equal? An Investigation Via a Coefficient of Variance," American Educational Research Journal, Vol 14, (1977), 493-498.

[ROMA86] Roman, David, "MS/DP Trapped Behind Backlogs," Computer Decisions, Vol 18, No 7, (Mar 86), 60-67.

[SCAC84] Scacchi, Walt, "Managing Software Engineering Projects: A Social Analysis," IEEE Transactions on Software Engineering, Vol SE-10, No 1, (Jan 84), 49-59.

[SCHE59] Scheffe, H., The Analysis of Variance, John Wiley & Sons, New York, (1959).

[SHAN85] Shanks, Jack G., "Inflating or Deflating Projects Depends on the 'Big Four'," Data Management, Vol 23, No 8, 18-21.

[SPRA86] Sprague, Ralph H. Jr., & Barbara C. McNurlin, Information Systems Management in Practice, Prentice-Hall, New Jersey, (1986).

[SPSS86] SPSSx User's Guide, SPSS Inc., 2nd Ed., (1986), 988 pgs.

[THAY80] Thayer, R.H., A. Pyster, & R.C. Wood, "The Challenge of Software Engineering Project Management," IEEE Transactions, (Aug 80), 51-59.

[TOEL78] Toellner, John, SPECTRUM-1: A Complete Project Management System, J. Toellner & Associates, (circa 1977), 10 pages.

[TRIP80] Tripp, Leonard L., & Pran N. Wahi, "How Much Planning in Systems Development?," Journal of Systems Management, Vol 31, No 10, (Oct 80), 6-15.

[WAGN59] Wagner, H., "An Integer Linear Programming Model for Machine Scheduling," Naval Research Logistics Quarterly, Vol 6, No 2, (Jun 59), 131-140.

[WEEK79] Weeks, James K., "A Simulation Study of Predictable Due Dates," Management Science, Vol 25, No 4, (Apr 79), 363-373.

[WEEK77] Weeks, James K. & John S. Fryer, "A Methodology for Assigning Minimum Cost Due-Dates," Management Science, Vol 23, No 8, (Apr 77), 872-881.

[WEEK76] Weeks, James K. & John S. Fryer, "A Simulation Study of Operating Policies in a Hypothetical Dual Constrained Job Shop," Management Science, Vol 22, No 12, (Aug 76), 1362-1371.

[WEIS73] Weiss, H.M., "Project Control," Journal of Systems Management, Vol 24, No 5, (May 73), 14-17.

[WHEE86] Wheelwright, James C., "How To Choose The Project Management Microcomputer Software That's Right For You," Industrial Engineering, Vol 18, No 1, (Jan 86), 46-50.

[WIES67] Wiest, J.D., "A Heuristic Model for Scheduling Large Projects with Limited Resources," Management Science, Vol 13, No 6, (Feb 67), B359-B377.

[WOOD85] Wood, Lamont & Don Leavitt, "Breaking the Applications Logjam, Part I," Computer Decisions, Vol 17, No 6, (Mar 85), 75-81.

[ZEMK84] Zemke, Ron, "Project Management by Computer," Training,Vol 21, No 3, (Mar 84), 48-53.

# VITA

| | |
|---|---|
| NAME: | Douglas Brian Bock |
| BORN: | Providence, Kentucky<br>December 5, 1950 |
| DEGREES: | B.S. Indiana University, 1973 |
| | M.B.A. Indiana University, 1984 |
| | Ph.D. Indiana University, 1987 |
| PROFESSIONAL ORGANIZATIONS: | The Decision Sciences Institute<br>The Association for Computing Machinery |