A methodology for automating the project management function in information systems design.

Sinan Baskan
A METHODOLOGY FOR AUTOMATING THE PROJECT MANAGEMENT FUNCTION IN INFORMATION SYSTEMS DESIGN

by

Sinan Baskan

A Thesis

Presented to the Graduate Committee of Lehigh University in Candidacy for the Degree of Master of Science in Industrial Engineering

Lehigh University 1982
This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science

August 25, 1982

(Date)

Prof. Larry E. Long
Professor in Charge

Prof. George F. Kane
Chairman, Department of Industrial Eng
ACKNOWLEDGEMENTS

The Author wishes to express his sincere gratitude to Professor Larry E. Long for his guidance and assistance during the course of this work.

Also, special thanks are due to Professor George R. Wilson for his suggestions on scheduling theory and integer programming problems.
TABLE OF CONTENTS

Acknowledgements .............................................. iii
Table of Contents .............................................. iv
List of Figures .................................................. v
Abstract .......................................................... 1
Introduction ....................................................... 2
General Methodology ........................................... 21
Resource Allocation ........................................... 47
PMCS Operation and Conclusions ......................... 97
References ......................................................... 119
Appendix .......................................................... 123
Vita ............................................................... 132
LIST OF FIGURES

Fig.1.1 Project Team Organization .............. 17
Fig.2.1 Network Representation .................. 22
Fig.2.2 Information available to PM ............ 25
Fig.2.3 I/O Information .......................... 28
Fig.3.1 Resource Constrained Schedule .......... 55
Fig.3.2 Precedence Diagram ....................... 56
Fig.3.3 Phased Precedence Diagram .............. 56
Fig.3.4 Installation Scheduling Problem ........ 62
Fig.3.5 Multi-Project Scheduling Model .......... 63
Fig.3.6 Project Scheduling For MIS ............. 67
Fig.3.7 Network Representation of a Stage ....... 76
Fig.3.8 Multi-Task Network ....................... 78
Fig.3.9 Matrix Representation ..................... 79
Fig.4.1 Decision Process ........................ 99
Fig.4.2 Resource Scheduling ....................... 101
Fig.4.3 Time-Phased Schedule .................... 101
ABSTRACT

The problem of directing an information system design project is investigated. A survey of existing methods and approaches to systems development is presented and a comprehensive methodology for the project management function and its automation is developed.

The basic premise of the methodology is the difficulty of setting quantifiable goals for each phase of the development process and the need to handle large volumes of data that are essential for successful monitoring of the activities. Hence an automated-iterative decision process that emphasizes qualitative decision-making is suggested. It is argued that if all the data pertaining to the activities of the project team are incorporated in "what...if" type scenarios, monitoring of on-going activities can be enhanced.

Finally, by forcing the user management and project management to evaluate the emerging system through scenario reviews and stage-wise decision-making an integrated approach to systems development and management becomes easier to accomplish.
CHAPTER ONE

INTRODUCTION

1.1 Management Information Systems

The arrival of computer-based information systems to the corporate management structure has created a new function (management information systems henceforth MIS) that has grown rapidly, both in size and in scope. In addition, the problems and intricacies it has brought to the organizations have been a major source of disappointment which has been so widely expressed by the user community.

Several studies have been made to explain the track record of the MIS function over the past two decades. The consensus is such that MIS grew out of simple data processing functions and its growth was primarily propelled by the rate of technological changes in computer systems rather than an overall conscious effort by corporate management.

More than a decade ago Diebold (10) complained about the fact that computers were being used for abstract applications. An eagerness to implement the state of the art and a lack of understanding on part of the management of the role and potential of MIS left the control totally to the data processing professionals. As the DP personnel
exploited the technology with ever increasing enthusiasm, the utility of information provided was never a primary concern in MIS implementation (10,25).

However, the changing business environment with increased uncertainties and the maturing of most DP departments prompted top management to review its concept of MIS. The growth of the MIS budget led to a demand on resources that now constituted a respectable amount (14) and pressures for a formal planning system arose due to the shortage of manpower, technical developments in hardware and software (27,28). The recognition of a need for formal strategic planning in MIS development initiated a debate on the role and potential of MIS. It became evident that given the amount of commitment required MIS should be more than a data processing activity. Simply, management had to get more out of its MIS department to justify the investment. The output of MIS began to be regarded as a management tool. Modelling the operations on the basis of information flow was introduced as an idea whose time has come (4,16,18).

Such a transformation of the DP-department would entail a broader concept of systems design. The decisions management makes on the basis of information provided would have to be scrutinized. The performance of MIS became closely linked to the performance of the
manager who had to rely on the information. Which and why certain pieces of information would be needed to make a certain decision, where and how this information would be generated, became very crucial questions. The user was invited to search for answers to these questions in cooperation with the MIS personnel. The user was drawn to the systems design process as a participant rather than remain at the receiving end. The user management is now expected to have not only an interest, but also a conception of things as they ought to be in MIS design. The systems analysis function had to cover the translation of user's conception into a workable framework for systems design.

Over the past decade various textbooks have been written to address the specifics of MIS design (5,9,32,36). The central theme is the fact that most MIS projects are "first"s and unique to their respective area of application. In the absence of historical data and previously established standards, the system specifications and performance criteria must be established in very general terms and have to be revamped continually. Structured systems analysis methodologies have been suggested to make such an evolution possible (9). Also formal planning procedures for the user as well as the MIS department and integration of corporate
planning and MIS planning have been suggested as means of ensuring user participation (27,28). The main point of these arguments is the assumption that a structured approach that starts from scratch inevitably ensures user participation.

Several researchers have indicated that MIS design by nature should be evolutionary since user perception of the system changes as the user becomes acquainted with the functional aspects of the emerging system (25,4,32).

This view has been substantiated by the empirical results acquired through various research experiments.

It became evident that the utility of MIS product is dependent on the information required, decisions to be made and delivery of information. Any shift in user perception of any one of these attributes may affect the performance of MIS even though the functional characteristics may not change (12,25,26,38).

The contention of most researchers and practitioners centered around the idea that a system that is designed to a great extent by the MIS professional may not serve to the satisfaction of the user. Even though the system may be functionally perfect, user's perception of its utility may be affected by factors other than those related to functional performance and therefore may never be used as it was originally intended. In addition, as
recent developments indicate, a merger between the old data processing and the quantitative analysis functions is taking place and the MIS department is being charged with applications development and planning as well (16,4,19). This development is a result of a growing consensus on the utilization of the MIS resource. The premise is that management information is more than tabulated data and a successful MIS should be capable of treating and interpreting data to the extent that policies can be formulated efficiently. This redefinition of the MIS function appears to strengthen the argument in favor of establishing a standard systems design methodology since the involvement is far more comprehensive and difficult to manage on a case by case basis.

In counterpoint, Dearden points out the futility of such a goal by noting that management functions in a dynamic environment that cannot be controlled by an overall methodology (7). He further stipulates that the MIS "product" is subject to the same budgetary considerations as any other product even though the MIS function is a staff function and supply/demand characteristics are quite different (8). In other words MIS function, as researchers have defined it; is impossible to establish and that the result will never be
the optimum. Also the MIS professional is expected to respond to contingencies and changing (sometimes unpredictable) needs.

The controversies reflected in these approaches to the study of MIS reveal the specific problems of design and management in the area. Basically the nature of MIS can be summarized as follows:

1. A typical MIS project takes several years from conception to completion. Since the requirements of the management it is designed for are constantly changing and shifting, the development process has to be scrutinized by the user management to ensure its effective utilization. However, this requires user proficiency in systems design.

2. Since most MIS projects are one at a time undertakings with very little prior experience (in the specifics of the application area) establishing standards and performance criteria can not be done at the time of initiation. Therefore the performance evaluation as well as the purpose and functions of the system evolves over the project cycle. These criteria and standards should reflect user's needs and therefore have to be developed by a trained and interested user.

3. A corporate-wide formal planning strategy is needed to monitor the evolution of the system throughout
the project cycle. However, since an optimum "product" is not possible, a planning system that can be based on a fixed goal may prove to be inhibitive. What is needed is a process that can monitor the changes and the evolution in requirements as well as the project work. Assuming a static set of objectives will not result in a successful implementation.

In the following section design strategies outlined to date will be reviewed. A procedure that can translate user needs to functional characteristics of the system (and vice versa) must be devised.

1.2 General Approaches To MIS Design

The preceding arguments point to the existence of demand for a general planning and management methodology to deal with MIS. Such a methodology is specifically urgent in this area because the beneficiary of the ultimate product and those who deliver it have widely varying perspectives. Closing the gap between these two perspectives have prompted the development of several schemes (13,15,17,20,22,37).

Since a MIS would aid the decision making process, most of the proposed planning systems center around the decisions the user is expected to make. Some systems suggest that the performance of the MIS as well as its
functional aspects be determined by unstructured and non-routine decisions (15, 20, 21, 22). In this case, the user is called upon to make the subjective judgements and shape the system specifications independently from the MIS professional. In terms of evaluation, user perception of utility is the basic measure. The systems analysts would automate as many of the structured, routine decisions as possible. In terms of performance, the system purports to relieve the user from routine, programmable decisions.

An alternative is a comprehensive planning and control procedure that covers all decisions (programmable and non-programmable) and establish decision-making patterns on basis of information flow (13, 37). Such a scheme is more involved and integrates institutional/environmental aspects to the planning. User participation may be extensive in the analysis of non-routine decisions but will be limited in functional/technical aspects. In this instance non-routine decisions are analyzed by the development personnel on a basis of institutional factors. In other words the user and systems analyst try to emulate and describe the general environment in an effort to facilitate the understanding and evaluation of factors that have bearing on the decisions.

In both cases the main feature of the system is to
have a procedure to assure user definition and specification of MIS objectives and functions. The latter includes routine decision making among the tasks of MIS.

The need for user involvement in the analysis of decisions supported by the MIS is also stressed by empirical research done in the area (11, 25, 26). Also surveys conducted indicate that as the unstructured decisions the MIS is intended to support increase, the early establishment of measurable project objectives and periodic audits contribute to the success of the project (11, 28). However, the availability of measurable objectives and performance criteria is a major problem. To alleviate this problem, user leadership of the project team has been considered to be essential (11, 22, 27).

A structured systems analysis approach, when used by such a project team serves to educate the user and enables the integration of changing perceptions into the system specifications.

Hence, a general philosophy for information systems planning and design has emerged and centered around the following points:

1. A formal planning system specifying,

   a. organizational objectives
   b. purpose and functions of MIS
   c. user decisions and user’s operating procedures
2. A structured decision/systems analysis procedure
   a. descriptive information flow analysis
   b. user specification of requirements

3. A project organization involving the user as the leader

The mere fact that such a general framework has been developed prompted questions dealing with the evaluation of MIS development process during the project. The basic question is: "Once the development project is initiated how do we know that we are making the right commitments?". One proposition on approaching the evaluation problem is to observe the direction the user is being led by the MIS (19). The contention is that a successful MIS should prompt the user to explore new applications as well as enhance his understanding of existing applications.

An alternative is to concentrate on the uncertainties involved and to commit the project team to monitoring certain pre-determined risk factors throughout the project cycle (1). In other words, the MIS should generate the kind of information and insight that helps resolve the uncertainties before a specific juncture is reached.

These two methodologies suggest that the project cycle be subject to periodic assessments and criteria for
such assessments be determined or at least outlined well in advance. However, the success of such a scheme depends on a standardized project cycle with all stages and tasks and decision points defined prior to the initiation of the project. Unfortunately there is little evidence in the literature that such a concept to define the project cycle has been developed. In fact even though the proponents of structured MIS planning advocate the use of formal planning tools such as PERT/CPM and scheduling models, they have not specified a methodology for implementation (1, 15, 27). The implementation of such formal planning and control tools is possible only if a formalized set of task descriptions and a project cycle are established.

In order to make a formal MIS planning system similar to those proposed, we also need a project initiation and control system to monitor the process as an integral part of MIS planning. It is not sufficient that such a system be a tracking procedure for activities that are performed over a period of time. The system must restrict the management to certain structured decisions and allow for evaluation and commitment assessment at pre-determined points of the project cycle. The following section will be devoted to outlining a project management system that will complement MIS.
planning systems.

1.3 Problem Statement and Project Management

Even though each management information system is unique, almost all system development projects follow the same life cycle. From conception to completion, the tasks and choices available to the user are generally the same.

Unfortunately, the emphasis on formalizing MIS design and development has not extended to detailing a general life cycle that could be applied to information systems development. This approach would definitely be superior to the structured systems design methodologies since these are intended to be design tools and not monitoring and evaluation systems (9).

An attempt is reflected in the development of ISMS (Interpretive Structural Modeling System) (17). Intended as a design and analysis tool, this computerized system uses graph theory to describe information-decision relationships. The resulting information flow diagrams and system flowcharts are suggested as tools by which a development project could be managed.

However, resources are committed on basis of individual tasks and not on information flow. Budgeting and task accounting procedures are known at the
initiation of the project. But the user perception of utility, user decisions and information requirements may change during the course of the project. To concentrate on such dynamic variables for project management introduces additional uncertainties.

The ultimate objective of a project management system is to maintain standards and eliminate uncertainties so that alternative approaches to the accomplishment of a task can be evaluated. The major features of the system essential to its purpose are itemized as follows:

1. A standard system development cycle: all tasks will be described in terms of resource requirements and time frame; interdependencies and precedence relationships will be established.

2. Structured decisions: at various points in the development cycle several options will be made available to the user. The user is forced to make a choice between these decisions. This purports to eliminate possible deviations from the original objective.

3. Quantitative models: To depend totally on PERT/CPM based techniques are risky since these models assume that the project manager knows enough about the activities involved to make estimates. In information systems this is definitely not true. However, given a
development cycle comprised of standard tasks, quantitative models can be employed for descriptive purposes. By confining the application of models to small subsets of tasks, alternative scenarios can be generated. One such approach has been developed by considering the organization as a marketplace where different departments compete for computing and information services (23). However, this model is intended to describe the operations of a MIS department and not serve as a design tool. Nevertheless, using quantitative models at various stages of development to provide "snapshots" to the user will not involve the risks and uncertainties of using them as a formal planning tool.

4. User involvement: The fact that the system is capable of confining the project team to structured decisions and also providing descriptive scenarios is bound to increase user involvement in all phases of development. The decisions reflect not technical capabilities available but the functional aspects of the emerging system. All structured decisions will ultimately deal with the question of "what kind of decisions do you want this MIS to lead you to?" or "how do you expect to integrate this information into your decisions?". Clearly only the end user of the system can be expected to answer these questions.
5. Finally, the project management system will have to rely on a computerized data base to provide all the information necessary for task accounting and scenario generation. Such an automated system will be more responsive, accurate and versatile. It is also hoped that better turnaround and extensive report generating capabilities of an automated project management system will facilitate user involvement and education.

1.4 Organizing for Project Management

The outline of the proposed project management system in the preceding argument did not address to the composition of the project team. However, the discussion on approaches to MIS planning methodologies did raise questions on user involvement and participation. At this point it is appropriate to define an organizational structure that would:

1. allow the execution and maintenance of the five features of the project management system presented in section 1.3
2. integrate user preferences and views as design criteria into the development process
3. and contribute to user awareness and education as the system evolves

The proposed organizational chart is given in figure 1.1 and position descriptions are as follows:

Data Processing Manager (DPM): Top level liaison
Fig. 1.1 Project Team Organization
between the user and the computer center personnel. Has line control over systems analysts, programmers and operations and staff position at top management level.

Systems Coordinator (SC) : responsible for various development projects and to ensure compliance with the overall corporate strategy on systems development.

Project Manager (PM) : responsible for a particular project oversees the activities of systems analysts and programmers working on the project and is responsible to maintain contact with users and user training.

Operations Manager (OM) : responsible for the administration of the machine room, scheduling tests and computing functions. Becomes part of the project team in determining hardware/software requirements and during parallel operation and conversion.

The vertical lines indicate line control over procedures, implementation strategy and the enforcement of decision originating from the ISPC. The horizontal lines indicate the communication with the user concerning system specifications and system performance.

It is important to note that at this point the system specifications and objectives have been approved by the ISPC at the conclusion of the feasibility study. The interaction between the project team and the user pertain to the modifications to and monitoring of the
system development.

For each major function or service that is intended to be an outcome of the project a systems analyst is assigned to be responsible. He/she will be directing the provision of those functions/services to the satisfaction of the user. His/her counterpart will be the user who will be using the function/service to make a decision.

The programmers who produce the necessary programs will be placed under the authority of the systems analyst who supervises the relevant function.

It must be noted that the organization is conceived to facilitate user leadership in shaping functional aspects of the system. The lateral relationships of users, systems analysts and programmers allow the evaluation of the system by its functional effectiveness as perceived by the user. In addition, the entire development process and resource requirements are exposed to the user increasing his/her awareness.

At decision points of the project cycle user and project members assigned to the user are required to go over their objectives by responding to a set of questions and reviewing system documentation produced to date and project documentation by PMCS.

The premise of the decision points lies in the recognition of DP professional's inability to translate
user needs into technical specifications. By exposing the user continuously to the development process it is hoped that the user will be able to converse with the DP professionals more effectively and still be the major decision-maker in determining functional specifications and objectives.
CHAPTER TWO

GENERAL METHODOLOGY

2.1 General Structure of the Project Cycle

As noted earlier, the key to success in information systems development depends on the ability to decrease the uncertainties and ambiguities involved and to educate the user. Hence, the computerized development procedure that will be outlined in this chapter is structured to achieve these goals.

Whenever decisions have to be made over an extended period of time the information that is necessary needs to be updated continuously. The essence of making the "right" or the "safest" decision depends on the ability to organize and filter the pieces of information that are immediately relevant to the decision at hand.

To illustrate; it would be worthwhile to examine the network given in figure 2.1. If each node represents a stage in the development process and each arrow a course of action that is a result of a decision made at the node the arrow originates from, a user at any given node would have to make a decision primarily on basis of what PM learned prior to the present stage. The objective is to reach to node-9 as soon as possible with minimum cost. The manager has started out from node-1 and reached to
Fig. 2.1 Network Representation of a Project
node-4 through any one of the alternative paths. PM has expended a certain portion of the limited resources available thus far. There is no reliable information available as to what it would take to reach to node-9 through the alternative paths. It is conceivable that mistakes may have been made prior to node-4. However, the decision will not remedy these mistakes. The manager must decide solely on basis of what can be done given the present situation. PM can choose to proceed to node-5 but cannot assess if it is easier to reach to node-9 once node-5 has been reached than it would be had an alternate route been chosen. Even though insight gained during the previous stages may be helpful, there will not be enough information to determine the relative resource requirements of possible routes with an acceptable level of accuracy.

In most environments (such as manufacturing, construction) past experience and the nature of the activity permits the gathering of quantifiable standards. Also the performance (and/or quality) evaluation can be based on the physical characteristics of the product (as in mechanical design). However, this is not the case in information systems development. The performance is evaluated on basis of utility, applications are almost always "first"s in that particular organization and
resource management is not based on prior experience. On the other hand, the fundamental tasks that need be performed to render a functioning information system are known and by and large standard. In other words the decision making environment of a project manager in charge of IS-development is very similar to the one depicted by figure 2.1. PM knows in general terms what needs to be done and in what sequence, but no idea on the necessary resource commitment and time duration for each specific task. In addition there is no reference to judge the intermediate results during the progress of the project. A judgement on basis of utility requires the use of the system for a period of time.

The information available to the project manager is depicted in figure 2.2 at a typical stage. The PM acquires user requirements and system specifications from user interviews or reports. PM is aware of the resources available and knows what needs to be done to design an information system. PM is expected to outline a plan for the proposed system development. The proposed plan will be scrutinized by the user and management before a decision is made as to whether the investment is worthwhile or not. However, in order to provide an accurate description of what is involved the project management should have information to answer questions
Fig. 2.2 Information available to the Project Manager
regarding the following.

1. Specifications for the proposed system commensurate with existing information/data structures and capabilities of the MIS department.
2. Familiarity with the application at hand.
3. User awareness and confidence in the information requirements necessary to make a decision.
4. User Head and PM agree on data flow and decisions to be made.
5. User’s ability to detail the required functions and awareness of the costs and time involved.
6. Agreement between UH, PM and OM on the operational aspects.
7. Reliable estimates on the overall time frame.

Without these issues being addressed, a full fledged commitment to initial system objectives and specifications cannot be expected. It would be safer to assume that no one can possibly have definite answers to those questions at the outset. The PM would increase the chances of success by encouraging a periodic review of these questions and modifications to the answers already given.

The technical expertise of system development personnel is required in two aspects of the development. First is the outline of the development cycle that would have to include all possible tasks, courses of action and choices available to the user. Second, the determination of the decision rules that direct the project team in the
most feasible approach given a set of conditions.

It is important to note that; at this point, it is assumed that the feasibility study has received a favorable response and the concern is to control budget outlays and to make sure that the specifications are adhered to.

Figure 2.3 illustrates the information that will be available to the PM at a given time.

At node I the PM is expected to assess the performance of the project effort vis-a-vis the overall system objectives. The information that is available at this point is itemized below:

1. Tasks to be performed: required tasks necessary to bring the project to completion. Derived from a detailed list of standard tasks that must be carried out in any systems project.
2. Results of previous tasks: budget outlays and time frame for the tasks that have been executed since the initiation of the project.
3. Resources used: type and quantity, % utilization, expenditures involved for each resource have to be known.
4. User input: user evaluation of the emerging system characteristics to keep track of changes, shifts in user needs as well as new applications that may be possible.

The output from the decision process at node I is a plan for the next stage which will end at a node J where a similar process will take place, until the project is brought to a conclusion.

Given this structure of the decision process in a
Fig. 2.3 I/O Information at a Decision Point
systems development project, an automated project management system would require the integration of following features:

1. A file of predefined tasks and task-descriptions which when put together would comprise the entire development process.
2. General standards for each of these tasks in terms of time, personnel and expenditures needed.
3. Pre-specified points in the progression of standard tasks which the development cycle can be divided into stages.
4. A set of conditions and decision rules indicating the general direction in which the project should proceed, should these conditions arise. These rules would be helpful in highlighting the strategies or choices with the highest probability of success with the information and results acquired up to that point.
5. A report generating program to make the information pertaining to nodes 1, 2, 3, 4 available and a resource accounting and scheduling program for possible schemes of resource allocation.

At any given decision point the project management system would be activated by the PM and would operate on tasks that have to be executed before the next decision point.

As stated before, the objective of this thesis is the development of such a project management system (henceforth referred to as PMCS).

2.2 Foundations of PMCS

The essential characteristic that distinguishes
The systems development methodology that is proposed here is its dependency on a task based development cycle. In other words, the development process is defined by the tasks that comprise the entire effort and are considered to be standard for all systems development projects, as opposed to a model defined by milestones and a time schedule. The advantage is the capability to monitor and direct task execution with respect to the overall objective. This must be the case since a major portion of the investment for systems development goes to the personnel costs. The difficulty in quantifying the output of these tasks necessitates close scrutiny of the tasks before and during execution rather than post-facto assessment.

The success of this approach to the definition of the development cycle depends on the completeness and detailed breakdown of the process into individual tasks. The breakdown should be in accordance with two general principles:

1. The tasks individually should not involve a major commitment in terms of funding, rather a series of interrelated tasks when taken together should comprise a fundamental commitment. This requires a more detailed breakdown than might have been regarded sufficient.

2. Each task must be segregated in such a way that a
definite distinction between the input information necessary, the information generated during the task execution and the final output can be facilitated. This is an essential factor since the departures and revisions to the original system specifications and goals should be identified as early as possible. If aforementioned distinctions cannot be made easily a gradual evolution away from the norm may take place without adequate scrutiny.

A development cycle that has come very close to observing these two principles has been suggested by Long (24). The details for tasks are thorough since the original intent of the author was to develop a documentation methodology. In order to render the development cycle suitable to stage-wise decision-making certain modifications are necessary. However, before going any further an overview of Long's development cycle will be given.

2.2.1 Standard Development Cycle:

Since the premise of Long's development methodology was to design comprehensive, consistent documentation procedures, the emphasis in breaking down the entire
process into phases and tasks was placed on activities that would give a profile of the system most accurately. Each task leads to definite contribution to the overall system documentation. The series of standard system documents are then used as the basis for review sessions that bring the user and the project management together. In this way both the PM and the UH can monitor the functional aspects of the emerging system. The detailed flowcharts of the three phases are given by Long (24) and presented in the appendix. In the following discussion these flowcharts will be referred to frequently.

The preparation of the standard system documents at the right time during the development process is the ultimate objective of this methodology. The decisions that have to be made are basically review oriented. It is assumed that system specifications and objectives that were approved at the end of a feasibility study will not be changed in accordance with a shift in user aspirations. The only possibility of change in system definition occurs when a cost overrun arises.

As a result the systems analysis phase is not geared to forcing the user to review his/her requirements. User training consists of teaching the user the function of the emerging system. With justification, no procedure to expose the user to alternatives in system design has been
suggested. As the system grows, it may become apparent that new applications are possible and feasible. The user through the PMCS should be exposed to these possibilities and have the chance of exploiting them. The documentation produced, provides a convenient tool to increase user awareness and when supplemented with alternate scenarios will create enough information for the user to make a more extensive set of decisions.

In addition, the documents that are specified to be the outputs of certain activities can be used as a vehicle of control by the PM. A review of these documents can enable the PM to set performance standards for the oncoming tasks. In particular, the up to date and organized nature of the information gathered would be useful in estimating the resource requirements for the next stage. Hence the development cycle as proposed already has the first and third features that are considered to be fundamental for a project management system. Furthermore, because of its documentation-oriented nature, necessary information for integration of standards and decision rules can be generated and organized systematically.
2.2.2 Modeling and Definition of PMCS:

The first step in the definition of the process is to integrate a set of decision rules into the overall cycle. The assumptions that are made in choosing the decision points in the cycle are listed below.

1. It is assumed that a feasibility study has been completed and the original system specifications have been approved.
2. A fixed amount of budget outlay has been authorized. The PM has the authority to allocate funds at his discretion. The breakdown of expenditures offered by the feasibility study is not necessarily binding.
3. A maximum level has been decided for each resource as far as resource commitment on part of the user and operations go. Again the PM has the final word on the allocation and scheduling of these resources.
4. A certain number of the project team members will have other responsibilities not related to the project. But PM, chief S/A and the chief Programmer will be on the project full-time.
5. Users will assign a representative who works with the PT full-time and user heads will be kept up to date on developments. At the time a decision point is reached all the user heads will participate in making the decisions.

In order to define the PMCS the phase flow diagrams of phases 2 and 3 will be used as reference. Certain points in these flow diagrams will be selected as decision points. The following rules apply to the choice of decision points.

1. A point in the flow diagram that may require a choice that may increase the potential of the system (by making additional applications
possible and/or enlarging the scope of present applications)

2. A point where it becomes obvious the resource commitment may have to be revised.

3. A point where it becomes apparent that the time frame may have to be revised.

4. A point where user input and commitment for development and/or operation of the system proves to be insufficient (this is mainly related to the system specifications and data flow within the organization)

5. A point where user aspirations and perception of the system may change.

Observing these rules, a walk-through of the phase flow diagrams would suggest the following points as decision points.

Definition of system objectives, scope and specification: corresponds to the 2nd block of the development flowchart. The main concern is the review of the outcome of the feasibility study and achieve concurrence among the user and the project team.

Personnel commitment: corresponds to block-6 of the development cycle. PM and UH must agree on the range and cost of involvement of their personnel.

Designing the data structure: corresponds to block-25. The emphasis is on confirming the logical relationships between the individual data-items and source documents and the records, files and output documents. The user at this point goes through the most significant period of self-education since the system may be confined or flexible to allow additional applications,
depending on the insight and foresight exercised at this point.

Cost review: corresponds to block-47 of the development cycle. At this point the development costs incurred to date are sunk costs and attention should be given to the operation and conversion costs. However, the performance data accumulated over the project cycle can be used to estimate and revise the costs for the following stages.

Planning for programming: corresponds to block-101. Resource allocation schemes must be reviewed and a time frame and cost plan must be decided upon.

The decision making procedure at these points will center on a series of questions that the PM, S/A and UH are expected to respond. The questions will require a "yes" or "no" type answers and are based on the logical progression of a systems evolution. Depending on the response to each question a course of action is suggested. These rules are intended to serve as guidelines and also as tools for exposing the user to a variety of possibilities.

The premise behind such a procedure is to provide an understanding of the interdependencies between user needs and the work required to satisfy those needs. Since decision rules are constructed on basis of what needs to
be or what can be done given the present objectives and specifications, they are not necessarily confining but rather descriptive in nature.

However, at this time a word of caution is in order. Prior to using PMCS, the PM and UH must understand clearly that the system is designed to aid them in making decisions by consolidating and organizing the information available. The emphasis is on highlighting the logical relationships between the user requirements and the resources and work involved. PMCS is by no means intended to be an automated design tool. The output reports of PMCS should be the basis of discussions and interviews between the PM and UH.

2.3 The Case for Computerization

The methodology outlined in this chapter has certain features that may render it fairly ineffective if its implementation were to be carried out manually. Since the basis of the system is the decision points and review procedures, manual implementation would imply several meetings, inter-office correspondence, debate and conceivably trigger confrontations among individuals. But in order to preserve the structured nature of the entire methodology the user and the MIS professional have to be relieved from these procedures. Also the
significant amount of information generated must be handled and stored effectively. The experience and expertise acquired throughout the development process as well as through other development projects can be documented by computerization of the entire process.

The user regardless of the application and the period of development is exposed to the same set of choices and guided by the same logic. The records of the project thus provide both a training tool and a basis for comparative analysis of various systems developed.

Each decision point (depicted in the flowchart as D-1, D-2, ... etc.) involves two kinds of decisions.

1. A qualitative assessment of the work accomplished so far
2. Allocation of resources and determination of the strategy for the next phase.

The discussion on the latter type of decisions will be left to the next chapter. Here, a checklist of questions concerning qualitative aspects of the system to be covered at each decision point is given. This checklist should serve as a tool for educating the user and evaluating the success of the emerging system relative to the user's initial conception.
D1 System Objectives and Specifications:

Input Documents:

I10: Applications Description
I20: Definition of Proposed System
I30: System Design Constraints
I40: System Reports and Documents
I50: System Data Files
I60: System Data Elements
P1: Data-flow Diagrams

Major Considerations:

1. Decisions that will be supported by the system
2. The information required for these decisions must be generated
3. The data structure must cover all pieces of information
4. Programmable vs. non-programmable decisions
5. System capability to support the data structure
6. Given the capability of the proposed system, possibility of additional applications

Procedures:

1. Use documents P1, I10 and I20 to answer the following questions:

1.1 Do the original nodes in the data-flow diagrams cover all the decisions the user is expected to make? (Y or N)
   If (N) complete the set of data-flow diagrams
2.2 Are all the applications covered by the decisions that can be supported? (Y or N)
   If (N): review I10 and I20 for missing or redundant applications
1.3 Are there any decisions that can be programmed? (Y or N)
   If (Y) review I40 for redundant output.
1.4 Given the data-flow pattern are there any additional decisions that can be made without modification to the data structure? (Y or N)
   If (Y) review P1 to reflect the data-flow for the new decision.

2. Use P1, I40, I50, I60 for the following questions.

2.1 Can all reports and documents be generated from the data-files?
   If (N) revise P1, I40, I50
2.2 Can any of the data elements be entered on-line? (Y or N)
   If (Y) revise I50, I60
2.3 Is it possible to generate data-files from a set of permanent files? (Y or N)
   If (Y) revise I50, I60, P1

3. Refer to I30, I40, I50, I60, P1

3.1 For each report or document; is it possible to confine source data to one data file? (Y or N)
   If (Y) revise P1, I50
3.2 For each data file; is it possible to confine the access to one? (Y or N)
   If (Y) revise P1

D2 Personnel Requirements and Operations:
-----------------------------

Input Documents:

I70 : Present System General Flowchart
I80 : Proposed System General Flowchart
I90 : Scope of System Development
I140 : System Status Report

40
P2 : Functional Analysis Matrix
I110 : Benefits Report
I120 : Cost Report
I1300 : Estimated Personnel Requirements
I1310 : Personnel Training Requirements

Major Considerations:

1. Assignment of S/A's to functions and services of the system
2. Availability of personnel over a time frame
3. Identifying special requirements (programming experience, software know-how etc.)
4. User commitment of personnel and time
5. Operational personnel requirements

Procedures:

1. Use documents P2, I1300, I1310 to answer the following questions:

1.1 Can a S/A be assigned to oversee each function?
   If (N) review P2 to combine two or more functions
1.2 Prepare a schedule of available S/A and programmer times over the time frame of the project; are the total man-hrs of personnel required equal to those available by category? (Y or N)
   If (N) review P2 to prioritize functions
1.3 Is a non-standard programming language required? (Y or N)
   If (Y) answer questions on training
1.4 Is there a new software/hardware required? (Y or N)

   If (Y) answer following:
   1.4.1 Any one in organization knowledgeable on the software/hardware to be used? (Y or N)
   If (N) investigate cost and
schedule of training by vendor /outside agency

1.4.2 Is there a training program available? (Y or N)
If (N) develop an in-house training program
If (Y) answer questions on training

2. Use I70, I80, II300, II310 and P2 to answer the following

2.1 Once the system is in operation, would there be an increase in the time user needs to provide input data?
If (Y) identify the additional involvement on I80 and justify
2.2 Would additional training be necessary for the user to provide input?
   If (Y) refer to questions on training
2.3 Weekly operations schedule require any revision to the routine operations of the user?
   If (Y) draw up an operational plan on a weekly basis

3. Training questions: P2, I80, I70, II310, II300

3.1 Can the training for programmers be carried out during systems analysis?
   If (N) revise time frame of development opportunity cost of delays etc.
3.2 Is the cost of in-house training included in overall budget?
3.3 Standard training available on a routine basis after system becomes operational?
3.4 User training possible during systems analysis?

4. Use I80, I90, I110, I120, P2, I140

4.1 Any revisions to II310, II300
necessitate budgetary changes?

4.2 Any revisions possible to I80, I90, P2 to alleviate inflating the budget

D3 I/O Requirements and Data Structures

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

Input Documents:

P1: Data flow analysis
II330: Output Reference List
II340: Output Description
II350: Data Element Matrix
II360: Data Element Codes
II370: Data-base/file description
II380: Record Layout
II390: Data-base Schema

Major Considerations:

1. Ability of the user to provide all the data necessary to produce output reports
2. Identifying sets of data that must be organized into a file
3. Output generation schedule and procedures
4. Maintainability of the data-base
5. Possibility of additional applications

Procedures:

1. Use documents P1, II330, II340 to answer the following

1.1 Under the present system are all the source data required for the proposed system available?
   If (N) define and discuss the procedures to complete the input data
1.2 Are there any output documents that use more than one source document?
   If (Y) investigate the feasibility of consolidating all source data that go
into an output in one file or record

1.3 Do the data flow diagrams indicate the use of the same data items at various stages of output generation?
   If (Y) investigate the duplication of data items in different files

1.4 If any one file is accessed for only one report would it be possible to merge the file into another file?

2. Use Pi, II350, II360, II370, II380, II390

2.1 Do all files require updating?
   If (Y) answer the following

   2.1.1 Can all updating be done at once?
       If (Y) consider one updating program

   2.1.2 Is updating of certain records/files depend on conditions?
       If (Y) Can these conditions be programmed?
       If (Y) consider random access or DBMS

2.2 Are updating sessions involve large number of changes in each file?
   If (N) consider DBMS

2.3 Are modifications possible in file/record structure?
   If (Y) consider DBMS

2.4 Are additional applications desirable?
   If (Y) answer the following

   2.4.1 Do the applications require additional files/records?
       If (Y) consider DBMS

   2.4.2 Do the applications require additional programs but same files?
       If (Y) consider DBMS
2.4.3 User-oriented on-line/real-time application? If (Y) consider random access, DBMS

2.4.4 Do the applications center on modifications to the reports? If (Y) consider report generators

D4 Cost Review

Input Documents

P1 : Data Flow Analysis
P2 : Functional Analysis
P3 : Functional Cost Breakdown
II400 : Detail System Flowchart
I120 : Cost Report

Major Considerations

1. Development costs
   a. deviation from estimates
   b. time frame

2. Operational Costs
3. Operation cycle
4. Budgetary changes
   Procedures:

1. Use P1, P2, P3 and I120

1.1 Is the total deviation from the budgeted amount more than the tolerance level? If (Y) answer the following

1.1.1 Is there a single function
responsible for the deviation?
If (Y) revise the function
If (N) answer the following
1.1.2 Is there a specific resource that is used excessively?
   If (Y) justify its use
   If (N) is there a unit cost increase?

1.2 Is there a difference in the cost of executing the operational tasks and the development tasks?
   If (Y) update operational cost report
CHAPTER THREE

RESOURCE ALLOCATION

In the preceding chapter the emphasis was placed on the problem of meeting project objectives and maintaining its utility. The present chapter deals with the management of the resources. The resources that will be the subject matter of this chapter are man-hours of various personnel such as systems analysts, programmers, etc. Monitoring the time of personnel is considered to be an effective way of keeping track of cost as well as the emerging product.

3.1 Resource Allocation Problem

The specific definition of the resource allocation problem depends mainly on the nature of the project. In general, the resource allocation problem is to determine the timing and the amount of usage of a known set of resources for a series of tasks that comprise the project.

Thus most resource allocation problems involve scheduling as well since resources are limited and must be assigned over a time period. Hence tasks must be scheduled in accordance with:

1. precedence relationships
2. resource availability
3. deadlines or milestones

In the context of information system development the information needed for the type of scheduling just described is not readily available.

The project manager has no standards to refer to in estimating the performance parameters. He relies on his experience and therefore needs to revise his initial expectations periodically.

In addition the resources available to him are not necessarily idle. The programmers and systems analysts have on-going duties. The data processing center operates on a regular basis whether the project team uses its services or not. Even though the resources may be committed to the project for a time period, the resource allocation decision still has to take the opportunity cost into account. Under such circumstances it is appropriate to think of resource allocation as resource sharing where success depends on the marginal improvement brought about by the additional assignments.

It is also important to note that the development cycle dictates the use of resources towards an objective rather than producing a quantity of output. In other words the resources must be used to achieve a given objective rather than to produce given quantity. In this
case the objective is to achieve an accurate profile of the system so that user and the PM can make the decisions specified in the development process. Therefore in allocating resources emphasis is on the earlier assignment of resources, use of which will produce the most reliable information.

In a traditional resource allocation model the PM may choose to attain one or several of the following objectives:

1. Minimize makespan
2. Minimize throughput time
3. Minimize idle time

where makespan time refers to the total time required to complete the project, throughput time refers to a single task and is the time elapsed between its initiation and completion and idle time is the duration of time that an assigned resource is not utilized. When the evaluation of output is quality-oriented and the plans are not made on the basis of a time frame, those objectives are not very useful and furthermore divert the attention of PM from other important aspects.

As was discussed in the previous chapter, confining the development tasks to a rigid time schedule is not advisable. In fact this may serve to impair the quality of the work. The main objective of the PM is an accurate
description of how the resources are used and what results are obtained. The cost-effectiveness is secondary to attaining the utility desired.

Hence, the resource allocation problem in information systems development centers around the problem of resource accounting (i.e. the capability to associate an expended resource with a specific output of the system).

Now it is time to state the resource allocation problem in formal terms:

Given a series of standard tasks that lead to a decision point and a set of resources, what are the possible combinations of resource assignments that can generate the most output required for the decision point as early as possible while observing precedence relationships and resource availability.

It is obvious that within the context of PMCS, the major concern is not meeting deadlines but to attain the output as early as possible. But the two remaining characteristics of the traditional scheduling problems are applicable here.

In scheduling tasks of the development cycle the following information is available for purposes of resource allocation.

1. function the task is associated with
2. cost of resource types needed for the task
3. precedence relationships

Before assigning a resource unit to a given task a scheme to prioritize the tasks requiring the same kind of resources at the same time is needed. The suggested list of priority rules are listed as follows:

1. If a function is expressed to be more important by the user the tasks of the functions can be assigned earlier.
2. For each task the resource that costs the least can be assigned first (or vice versa).
3. For each task the resource that contributes most to the output can be assigned first.
4. For parallel tasks the one that requires the least time must be scheduled first.

These priority rules should be determined by the PM and should reflect the preferences and strategy of the user management. It is true that each priority rule may produce a different schedule. However, each schedule can be used as an alternative solution to the resource allocation problem based on different assumptions. Once a resource is assigned, it will not be released until the task is completed. In other words a resource committed to two different tasks will not be allowed. (This is especially true for personnel).

The premise behind this scheme is the completion of "cheaper" jobs as early as possible so that their output
can enhance the understanding of the nature of the project under development.

This is not to say however that if the costs escalate the "more expensive" tasks will be cancelled. All the tasks that lead to a decision point will have to be performed regardless of cost-performance. Once the decision point is reached though an assessment of project feasibility can be made.

In accordance, with the objective of increasing user awareness and presenting available options several schedules will be generated with a time frame that results from the scenario. The user will then be in a position to determine where the resources are being committed.

Smith and Wechsler (35) have recommended use of descriptive scenarios as a design tool, forming a basis of discussion between the user and the project team. The resource allocation module of PMCS uses the same concept to establish

1. a time frame
2. an acceptable sharing of resources
3. resource accounting

3.2 Resource Allocation Models

The solution to the problem outlined in the previous
section can be obtained in several ways. Since the problem inherently involves scheduling, a combinatorial optimization approach can be employed. This would enable the analyst to set a time frame in terms of weeks and divide this time frame into smaller periods. Then resource units can be assigned to these time slots according to certain criteria and a set of resource constraints.

Another alternative is to resort to deterministic models. Linear integer programming, zero-one integer programming and in cases where more than one objective exist integer goal programming are commonly employed for resource allocation problems.

Still a third approach is the use of heuristic algorithms; approximate optimization methods which in comparison to other techniques give suboptimal results but are computationally more manageable.

In the following sections, these methods will be compared and their applicability to systems development projects will be discussed.

3.2.1 Combinatorial Optimization:

This technique in its simplest form generates
arrangements of elements (or tasks) and then finds an optimal arrangement.

For purposes of illustration, figure 3.1 depicts a typical work-week with each day comprised of eight man-hrs. \( R(i,j,k) \) denotes the amount of resource \( k \) available at the \( i \)th hour of week day \( j \). Assuming there are three resources with available quantities being \( R(1) \) \( [i=1,2,3] \) and six tasks that need these resources with precedence relationships as per figure 3.2, the scheduling problem requires a match between the tasks and the time periods when resources are available. If resource \( (k) \) is not available on \( j \)-th day of the week at \( i \)-th hour then obviously no task that requires resource \( (i) \) can be scheduled for that time slot. In the meantime, the precedence relationships would have to be observed also.

At this juncture, it is important to note that the resources are total man-hours of various types of work available to the project manager. However the effectiveness of scheduling will be judged qualitatively. This matter will be addressed in detail later on. In order to solve this problem following strategies are applicable:

1. All permutations of tasks can be generated and then those permutations that violate the precedence
Fig. 3.1 Resource Constrained Schedule
Fig. 3.2 Precedence Diagram

Fig. 3.3 Phased Precedence Diagram
relationships can be left out creating a set of schedulable activities. Then the problem is one of matching periods of availability with series of tasks. Some widely used matching algorithms have been described in detail by Minieka (30). However, as the problem increases in size these algorithms become increasingly time consuming.

2. In order to ensure that precedence conditions will not be violated the network of tasks can be grouped into divisions (or categories) as per figure 3.3. This means that tasks in group I must be completed before tasks in group II can be initiated and so on. Hence resources are shared between a smaller number of tasks. An examination of the network would indicate that tasks 1, 3 and 4 can be executed without executing task 2. However, the tasks of group II would not be completed when task 4 a member of group III is initiated. Depending on resource availability and time duration of the tasks, there may be cases where this sequence may be advantageous. On the other hand the amount of processing required for tracking the individual activities increases.

In the context of PMCS allocation and scheduling problems must be solved in an interactive environment with a rather limited number of alternatives. It pays to
bear in mind that the relative importance or even the urgency of the tasks in the eyes of the user must be built into the algorithm. In so doing another drawback is introduced to the use of the combinatorial approach (in addition to computational efficiency). The major advantage of this approach is the ability to scan through all possible solutions. However, the introduction of the user-oriented conditions, this ability would be seriously undermined. The fact that the number of tasks to be scheduled may be limited will not alleviate these disadvantages entirely. On the other hand developing a model along the lines argued here would involve considerable amount of work without insuring a significant advantage.

3.2.2 Integer Programming:

Integer programming, especially zero-one integer programming is the most widely used technique in scheduling. Generally, almost all multi-resource, multi-project scheduling problems can be formulated as zero-one integer programs.

The typical problems that has been solved by integer programming are production planning, job shop scheduling
and staffing/manpower planning problems. The common characteristics of these problems are listed below:

1. A series of projects (or tasks) exist with known execution times and precedence relationships.

2. A set of resources (or categories of personnel) with available number of units and requirements for each task known.

3. A time schedule by which all or a subset of the tasks are expected to be completed.

4. A cost/benefit schedule that lists the relative costs of completing jobs on time, late or ahead of time as well as the cost of additional resources (or overtime).

Most of the models developed to date have assumed that the data describing task characteristics, precedence relationships and resource requirements were reliable and would not change over the time frame the model was developed for.

These models are usually developed to optimize a function that can be quantified and translated into monetary terms. The objective function may be a reduction in project duration while increasing costs (33) or minimization of idle time for resources (2a) and minimization of total project time (36).

Although these models may perform satisfactorily in
decision environments such as transportation, manufacturing management, facilities planning, inventory management, systems development process does not lend itself to modelling in those terms. First of all the verification of data on tasks and resources is difficult and in most cases such data does not even exist. Also feasible solutions can be obtained when an objective function is formulated in terms of the variables the constraints are formulated in. However, qualitative objectives are very difficult and complicated to formulate in this manner. Furthermore, as stated earlier, emphasizing quantifiable benefits may impair the essential objectives.

The only advantageous use for integer programming approach without running the risk of oversimplification or diverson is in the area of budget scheduling and cost control.

Two models suggested for different purposes but still relevant to the area of systems development approach cost control by minimizing the total cost of resources incurred over the entire time frame.

In one of the models the installation scheduling problem has been formulated as a mixed integer program (6). The installation scheduling problem is defined in general terms as follows:
There are a set of investment to be made over a time interval which is divided into stages. With each stage a cost function and a benefit function are defined. There is an initial capital position at the beginning of the time interval. The objective is to find a schedule of investments that will maximize the capital position at the end of the time interval.

This approach can be accepted as a tool for the budget reviews that the project team and the user would have to make. The problem formulation is given in figure 3.4

where \( N \) is the number of types of investments, 
\( m(i) \) is the number of investments of type \( i \),
\( C(i,t) \) is the cost of investment of type \( i \) in time period \( t \),
\( B(i,t,r) \) is the capital available at time period \( t \) for an investment of type \( i \) that had been made at time period \( t-r \),
\( K \) is the initial capital investment,
\( x(i,t) \) is the number of investments of type \( i \) up to and including the period \( t \),
\( P(t) \) is the capital position at the end of time period \( t \) and
\( S \) is a constant.

The one problem with integrating this model into PMCS is its computational efficiency in comparison to the available computing power.

Another use of integer programming is the multi-project scheduling model suggested by Pritsker et al (29). The formulation is given in figure 3.5. Any one
Maximize: \( P_t \)

\[
\begin{align*}
\text{subject to:} & \\
\sum_{t=1}^{T} x_{it} &= m_i \quad (i=1, \ldots, N) \\
\sum_{l=1}^{N} c_{il} x_{lt} + p_{lt} &= k_l \\
\sum_{l=1}^{N} \sum_{r=l}^{t-1} b_{itr} x_{it-r} &= s_l \\
\sum_{l=1}^{N} a_{ir} x_{it} - p_{lt-1} + p_{lt} &= s_l \\
\sum_{k=1}^{t-1} x_{ik} - d_{it} &\leq 0 \quad (t=1; \ldots, T-1) \quad (l=1, \ldots, N) \\
\sum_{k=1}^{t+1} x_{ik} - d_{it} &\geq 0 \quad (t=1, \ldots, T-2) \quad (l=1, \ldots, N) \\
d_{it} &\leq m_i \quad (i=1, \ldots, N) \quad (t=1, \ldots, T-1) \\
\text{all } x_{it}, d_{it}, p_{lt} &\geq 0 \\
\text{and } &\text{ integer}
\end{align*}
\]

Fig. 3.4 Installation Scheduling Problem
I) \[ \text{MAX.} \sum_{i=1}^{I} \sum_{t \in E_i} x_{it} - \frac{1}{M} \sum_{i=1}^{I} \sum_{t \in E_i} \sum_{l \in E_i} t \cdot x_{il} \]

\[(M > \sum_{i=1}^{I} \sum_{t \in E_i} L_{it})\]

II) \[\text{MAX.} \sum_{t = \text{MAX.} E_i}^{\text{MAX.} G} x_t \]

\[(x_t = \begin{cases} 1 & \text{all projects complete by } t \\ 0 & \text{otherwise} \end{cases})\]

III) \[\text{MAX.} \sum_{i=1}^{I} \sum_{t = q_i + 1}^{\text{MAX.} G} (p_{i_t})(x_{i_t}) \]

\[(q_i : \text{desired due date for project } i)\]

I) \[\sum_{t \in E_i} x_{it} = 1\]
2. \[ \sum_{q \in E_{i,t}}^{t-t} \chi_{i,qt} = 1 \]
\[ \frac{1}{N_{i}} \sum_{I=1}^{N_{i}} \sum_{q \in E_{i,t}}^{t-t} \chi_{i,qt} \geq \chi_{k,t} \]
\[ (t = \max(E_{i}), \ldots \max(G_{i})) \]

3. \[ \sum_{t \in E_{i,m}}^{L_{i,m}} \{ t \chi_{i,mt} + a_{i,n}^{m} \} \leq \sum_{t \in E_{i,n}}^{L_{i,n}} t \chi_{i,nt} \]

4. \[ \sum_{I=1}^{I} \sum_{J=1}^{N_{i}} \sum_{q=t}^{t+d_{i,t} - 1} (r_{i,k}^{m})(\chi_{i,qt}) \leq R_{k,t} \]
\[ (t \leq q \leq t + d_{i,t} - 1) \]
\[ (t = \min(a_{i,n}^{m}), \ldots \max G_{i}) \]
\[ L_{i,t} \leq t \leq E_{i,t} \]
\[ \chi_{i,qt} = 0 \]

Fig. 3.5 Multi-Project Scheduling Model (continued)
of the objective functions I to III can be adopted and solved subject to job completion constraints, project completion constraints, precedence constraints and resource constraints which are expressed in equations 1 to 4 respectively,

where i is the number of a project, i=1,......,I;
j is the no. of a job of a project, j=1,......,N(i);
I is the number of projects to be scheduled;
N(i) is the number of jobs in project i;
G(i) is the absolute due date for project i;
L(i,j) is absolute due date for job j of project i;
a(i,j) is the arrival period for job j of project i;
d(i,j) is the number of time periods required to process job j project i;
E(i,j) is the earliest possible period in which job j of project i could be completed;
E(i) is the earliest possible period by which project i could be completed;
X(i,t)=1 if all jobs of project i are completed on or before t=1 and X(i,t)=0 otherwise;
X(i,j,t)=1 if job j of project i is completed in t and X(i,j,t)=0 otherwise;
k is the number of a resource, k=1,......,K;
r(i,j,k) is the number of units of k required for job j of project i;
R(k,t) is the number of units of k available in t.

In order to render this model meaningful to the project management, an objective function that will minimize costs must be added. In this case a set of assignment schedules and their relative costs could be generated. The modified formulation is given in figure
3.6.

This model is also a mixed integer programming formulation and its basic objective is to track the cost of resource expended on each task through time periods, thus increasing the accountability for each resource during each time period. Again \( f(i,j,k) \) is an integer (1 or 0) indicating that resource \( k \) is assigned at time \( t \), \( d(i,j) \) is the number of time periods required for task \( j \) of project \( i \), \( r(i,j,k) \) is the number of units of resource \( k \) assigned to task \( j \) project \( i \), \( R(k,t) \) is the number of units of resource \( k \) available for time period \( t \), \( C(k) \) cost of resource \( k \) per unit, \( N(i) \) is the number of jobs in project \( i \), \( N(k) \) is the number of projects in stage \( k \).

Once more the computing power available prohibits the integration of this model into PMCS because the processing requirement increases at a very fast rate as the number of variables and constraints increase. However, the point has to be made that budget allocations can be scheduled over the entire period of development by employing the first model. Then at each stage the Pritsker model can be activated to monitor the cost behavior of the project.

These models are intended to be descriptive as far as systems development goes. In other words, they explain how the resources should be assigned to minimize
\[ \begin{align*} 
\text{Min.} & \quad z = \sum_{i,k} (c_k)(r_{i,k}) \\
\leq t & \quad R_{kt} = \sum_k r_{i,k} = \sum_i \sum_j t_{ij} + \sum_j d_{ij} \quad \left( r_{i,k} \right) \quad \left( s_{j,k} \right) \\
T & = \sum_i \sum_j t_{ij} \quad \left( d_{ij} \right) \quad \left( s_{j,k} \right) \\
\text{eq.} & \quad d_{ij} = \sum_i s_{i,j} \\
\hline 
\text{Fig. 3.6 Project Scheduling for Information Systems} \\
67 \end{align*} \]
cost. Also since the data employed are merely estimates that lack historical basis the inferences that may be drawn from the results should be secondary to the qualitative judgements of the user and the PM on the utility of the system.

The last model offered is the simplest in the sense that it concentrates on the fundamental problem of information systems design; accounting for resources in the absence reliable forecasts or experience on resource use. Over a period of time the information generated by using this model may provide an adequate data-base and a descriptive perspective of the process. Nevertheless, the decision on which model should be used almost all the time will center on the cost of implementation.

3.3 Resource Allocation Algorithms of PMCS

The size and context of the resource allocation problems that are expected to arise during the information systems development projects do not justify the use of large scale quantitative models given the present state of the art. The range of activities both in number and in scope are limited and even if a feasible and reliable model were to be developed it is questionable if all the benefits of large scale modelling can be ensured in such an environment.
One justifiable strategy of using deterministic models would be the scheduling of all activities and resources for the entire development process at once. But this would have refuted the premise and major advantage of PMCS as explained in the second chapter by committing the project team to a long-term plan with no vehicle of review and adjustment.

Therefore, the resource allocation will be carried out by heuristic procedures, the development of which is explained in the following section.

3.3.1 Criteria for Scheduling Activities:

For each activity (or task) the following attributes are stored in the data-base of PMCS and are standards for use in scheduling and resource assignment:

1. resource type (systems analysis time, programming time, etc.)
2. quantity of the resource required subject to a standard rate for cost calculations (measured in man-hrs)
3. precedence relationships

Initially all tasks are schedulable, meaning none of the tasks has a reserved time slot in the overall time frame.

Scheduling a task implies that the necessary
resources have been allocated to the task and its place in a sequence of activities has been identified.

For any activity in the systems development process one of the resources is personnel time expressed in man-hours. The time required for the execution of an activity is the sum total of man-hours spent in analysis and programming. The personnel time and its monitoring is a very convenient tool in assessing the qualitative evolution of the system. The case for this argument was made in the second chapter. On this premise, using the total man-hours as the execution time of a task is thought to be justified.

In assigning resources to tasks, the tasks to be scheduled may be selected according to the following priority rules:

1. Assign the available resources to the tasks with longest (shortest) completion times.
2. Assign resources to tasks that lead to the production of a system document
3. Assign resources to tasks that are the precedents of the largest number of tasks
4. Assign resources to the tasks that require most of the most expensive resource.

Given an initial hypothetical source node S from which all other tasks originate, the set of schedulable tasks are those that can be immediately executed. If the resources to meet the requirements of all these tasks are not available then one of the priority rules must be
used. When all tasks are scheduled then the tasks that originate from those already scheduled will comprise the set of schedulable activities.

The application of the priority rules separately will generate various schedules. The time frame and cost distribution (hence budget requirements) over the project cycle will be derived from the schedule. Then the UH and PM can investigate those alternate schedules and make decisions accordingly.

It is important to note that the schedules pertain to tasks that are to be executed between two decision points. Therefore a commitment to perform the tasks is about to be made when resource allocation schedules are generated. The performance of the previous resource assignment must be scrutinized in order to use the present schedules effectively.

3.3.2 Project Objectives and Priority Rules:

In general, several objectives can be adopted when project scheduling algorithms are being used. Some of these have been listed in section 3.2 (minimizing makespan etc.) The choice of such objectives have a bearing on the choice of the priority rules.
In information systems, the choice of objectives centers on sharing the resources (especially manpower) and generating as much information (and understanding) about the emerging system as early as possible. Under these circumstances priority rules that schedule tasks at the earliest time they are allowed to start, are preferred.

On the other hand, a tight budget would prompt PM and UH to choose priority rules that schedule tasks which use "cheaper" resources initially while the results of these tasks enhance insight. When user perception is weak and user involvement needs to be improved, scheduling document-oriented tasks must be favored.

The choice of the priority rules are left entirely to the PM and UH. However, unless strong convictions exist for or against specific objectives or rules, a set of schedules for each of the rules is recommended.

PMCS provides a "menu" of possible objectives (or circumstances) which may justify the use of a specific priority rule. However, this facility may not be used at all.

The PM and UH are responsible for reviewing the results at the end of each stage and update the standards for the tasks that have been executed. Unless this is done the inaccuracies of the previous resource
assignments will be repeated during the next stage.

In establishing the criteria for scheduling and setting priorities the PM faces two major questions.

1. in assigning resources to a set of tasks that must be performed repeatedly for a number of different projects which of the projects should be favored?
2. once a priority among the projects has been established, what rules must be used to time-phase the allocation of limited resources?

Before a methodology to address these two questions can be developed, a detailed definition of the problem will be given.

Before the project management process is initiated, the PM has three sets of information available as computerized files.

a.) standard tasks that comprise the project cycle: standard resource requirements for each task, input and output requirements and precedence relationships are pre-defined.
   b.) resource requirements: all available resources and their availability by periods (weekly, monthly etc.) are kept up to date. This information reflects the overall capacity of the organization and is beyond the control of the PM.
   c.) system projects information: The projects that must be undertaken by the project team and the estimates of time and resources needed with a ceiling on budget allotment constitute the third input file.

The first file is and must be standard for any given organization. The last two are updated as new investments bring in new resources and new developments.
necessitate new systems to be designed.

For all design projects added to the third file, the standard development cycle is valid. In other words all the standard tasks must be performed for each project. Whenever this is unjustified, zero resources and completion time can be assigned. However by adopting this hypothesis control over the projects can be maintained routinely.

In this context, it is possible to think of the standard development cycle as a PERT/CPM type network. If there are N system design projects then it is possible to envision N separate project networks. However, from a resource management and project control standpoint it would be more feasible to deal with one network that encompasses the entire work to be done. This is so since the work for the N separate projects must be performed by the same people/depot etc. and it is this work that needs to be monitored. The overriding objective is to manage the information system function within an organization with consistent, favorable results. Having N different project teams and N different work assignments does not guarantee an effective control of the aggregate output. Accounting for the possible slacks, surpluses and shortages of resources that may arise in and vary from project to project would be possible.
Hence, a project network containing all possible tasks regardless of the project they belong to must be constructed. To illustrate consider the following example.

Example-1: A standard development cycle has been developed consisting of ten standard tasks. Three systems will be designed using these standard tasks. If \( S(j) \) denotes the project \( j \) and \( T(i,j) \) the tasks then the network for the project \( j \) will be as in figure 3.7.

Assuming \( j=1,2,3 \) this scheme produces three different networks. If one network for all systems under development were to be built each \( T(i,j) \) would be a vector of three components each component corresponding to one of the projects. This structure is similar to the one that might be used to depict the multi-resource, multi-project model suggested by Pritsker et al.

However, a problem arises in constructing this network. Given a task vector \( T(1,j) \) \( (j=1,2,3) \) what is the best method to establish precedence between \( T(1,1) \), \( T(1,2) \), \( T(1,3) \)? In a broader context is it really necessary to complete a specific task for all systems or can \( T(1,1) \), \( T(1,2) \), \( T(1,3) \) be performed at different phases? These two questions render most of the standard scheduling and resource allocation algorithms difficult to use in their application to information systems since they concern the sharing of a human resource i.e. personnel.

In attempting to define a procedure to prioritize the various projects the multi-network structure must be transformed to an aggregate network that reflects the entire set of tasks to be executed. In the previous sections decision points were defined to divide the development cycle into stages. In the aggregate network, the same stages are retained except there are as many of
Fig. 3.7 Network Representation of one Stage in the Development-cycle
each task as there are systems under development. This is obtained by duplicating the standard development cycle for each system and retaining the precedence relationships as defined by the standard-cycle.

Once the aggregate network is constructed it appears to the PM as if one network composed of m*n tasks where n is the number of systems to be designed and where m is the number of tasks in the standard project cycle.

The next step is to time-phase the project cycle to assign and monitor resource commitments over a period of time. Time-phasing refers to the re-definition of the project-life in terms of time periods during which resources can be assigned to a specific task and be accounted for only in terms of those tasks, over the the time period. Consider the following example.

Example-2: Continuing with the problem of example-1 consider the entire network as shown in figure 3.8.

Now, a time-period t can be chosen to divide the project cycle into equal intervals Each interval would have a certain quantity of the resources assigned to it. Then the general scheduling problem would be to determine the assignment of the tasks of the aggregate network to these intervals. If these periods are to be thought of as the columns of a matrix then it would be possible to think of rows of the matrix as functions or systems to be designed.(figure 3.9).

The assignment of tasks T(i,j) of the aggregate network to a specific time period would depend on resource availability and system requirements (such as minimum makespan, minimum delay, minimum cost etc.) and precedence
Fig. 3.8 Multi-task network representation of a time-period in the development-cycle
Fig. 3.9 Matrix representation of a multi-task network. This representation simplifies computer manipulation and storage of data for each time period.
relationships. All $T(1,j)$ need not be assigned to the same period and the tasks for any $j$ need not follow on another period by period. This approach would facilitate optimizing the resources for each period since at each period there is a wider selection of schedulable tasks than would be the case if each period was required to have one task of each system.

At this point, it is important to note that the definition of period depends solely on the unit of resources that the PM uses to account for the resource commitments. In other words if the contention of the PM is such that his evaluation require weekly usage of resources, then time period is a week. Another word of caution is not to confuse the stages with periods. The stages of the development cycle are pre-determined and are defined by the decision points of the standard development cycle. Periods are simply units of time during which certain resources must be expended.

Reflecting on what has been proposed so far, it would be worthwhile to consider the automation aspects of the process. Two files, one containing the standard development cycle and another having the system projects information, have been processed against one another to produce a third file that contains all the tasks of all the projects in network form (i.e. precedence relationships). This file would contain the following information on each task.
1. System code and task-code, descriptions
2. standard resource usage for each resource required
3. weight of the task: standard % amount that should be applied to the budget figure to determine the total cost
4. duration of the task.

In addition to this file the PM would have his time-table defined in terms of the unit-periods. Now to complete the circle the PM has to answer the questions raised but left unanswered earlier:

1. How to set priorities between various projects?
2. How to schedule resources?

The following section develops the basic approach to each one of these questions.

3.3.1.3 Determination of project task priorities:

Since one of the primary objectives of the thesis is to provide a procedure to monitor and control the project effort, the approach to establishing priorities must be capable of increasing insight and understanding of the possible consequences due to the courses of action. To facilitate decisions of this nature, descriptive information and a projection of the various possible schemes must be provided. This, in fact amounts to developing a series of scenarios each reflecting the
projected results of a feasible priority scheme.

There are several cases that may necessitate consideration of a limited number of priority rules. Each of these cases are reflective of the general environment where a PM is forced to a trade-off. The nature of these various cases may confine the choice of scheduling schemes to a few. The following is a brief description of some general cases:

1. The required completion time of a project may be sooner than those of the competing projects or its actual completion time may be significantly shorter. Under these circumstances completing this project as soon as possible and reducing the demand on the resources may be advisable.

2. The total budget allotted to a given project may be rigid and tight. In cases where this is a reflection of uncertainty of the utility and/or success of the system under consideration then early work on the system may provide the insight and information for review of the commitment to the project. If the tight budget is a reflection of a revenue shortage than a lower priority may be justified. This is based on the assumption that a project under a tight budget is less likely to succeed hence it pays to commit limited resources to systems that has a higher probability of success.
3. A certain project may be urgently needed. If its importance relative to the remaining projects can be assessed then the priority is a direct result of such assessment. Probably the best way to assess such relative importance is to determine the break-even point between the penalty costs of not completing the remaining projects in time and the estimated benefits from the system under consideration. Of course when most perceived benefits are intangible this process is substantially more difficult.

4. A project may require an impropportionate amount of certain resources that may not be available at a later stage (or period). In this case, all other things being equal; it may be wise to assign a higher priority to the project. Conversely, a system may require resources that are either not readily available or the commitment of which may create problems in a given period. A lower priority would push this project to later periods where resources may become available.

5. Finally, systems that have a large budget should be initiated as early as possible because the returns on investment can begin to be realized earlier.

Although this list can be expanded further, it is considered to be sufficiently representative of cases that arise in practice.
It is essential to note that a capability to process the information contained in the schedulable task file and to organize it in accordance with PM's choice of a priority scheme exists but no decision-making capability has been generated. The PM and the user would have to decide on one or a combination of the five cases as their priority rule. So far the advantage PMCS boasts is its ability to generate aggregate network schedules of various configurations. In the following section a similar capability for generating resource allocation schemes will be outlined.

3.3.1.4 Resource Scheduling Schemes:

Once the aggregate project network is finalized resources can be assigned to each period and then tasks from the network will be assigned to unit-periods. To visualize the procedure consider two sets one having resource quantities the other having tasks. Each task would have its own resource requirements. If a quantity of resources is chosen and placed in a third set, then in choosing a task that would be placed into the third set, the quantity of the resources would be an upper bound. Conversely, if a task is chosen first, then in choosing a resource quantity the resource requirements of the task chosen would be a lower bound. Now, considering that the third set is a unit period, two basic approaches to
scheduling arise. It is now appropriate to dwell on the premises and circumstances that may justify these approaches.

Case-1: Fixed Resource Commitments

When resources are committed to a series of on-going activities and resources cannot be reassigned, scheduling tasks into unit periods and creating lower bounds that may not be met is not a feasible approach. Purely from a data processing standpoint, such a strategy increases the necessity of updating schedule files and necessitates an iterative processing that cannot guarantee an optimum schedule. However, when the resources that can be made available to the PM for each unit-period is determined beforehand, scheduling tasks into the unit-periods is a straight-forward procedure. A set of resource constraints exist for each unit-period and as long as these are observed, optimizing the schedule is a matter of choosing the tasks in a way that minimizes either of a combination of delay, idle time, through-put time etc.

This approach assumes that no time constraints exist on the completion of the individual tasks. Even when such constraints are imposed, the fact that the period is fixed is the overriding constraint. A secondary objective of minimizing the penalty cost due to late completions can be imposed.
Case-2 : Limited Time Requirements on Tasks

The tasks have fixed deadlines or there is an overall through-put time requirement. In this case the major concern is to make sure the tasks are performed within a given time frame. It is obvious that having an upper bound on resource availability impedes the realization of such an objective. Therefore instead of scheduling resources into unit-periods first and then scheduling tasks while satisfying resource constraints, tasks are scheduled first then resources are assigned in sufficient quantities to meet the lower bound.

Again an inherent assumption in this case is that resource quantities are large enough to accommodate the requirements of each unit-period. The optimization problem is then to optimize resource usage since scheduling tasks first would satisfy the requirements on completion times, through-put and minimum delay etc.

Case-3 : Constraints on both Resources and Completion Times

If resources available for a unit-period do not meet the task resource requirements and at the same time tasks have time constraints on them two possibilities arise:

1. If it is possible to have additional resources at additional cost and a break-even point can be reached between the penalty cost of completing tasks late and the
cost of acquiring additional resources.

2. If no additional resources can be committed then case-1 holds since resource utilization will be optimum and the secondary objective will minimize the penalty costs. Resorting to case-2 would involve the risk of not being able to complete all the tasks.

These three cases are representative of most systems development environments, most common of which is the third case where both time and resource constraints exist. By scheduling resources or tasks for unit periods the resource allocation problem is reduced to the optimization of time-schedule or of resource use respectively. Now an allocation algorithm need be developed that can be used once the resource scheduled is reduced to a heuristic optimization problem.

The allocation algorithm that is needed must allocate tasks rather than resources. This is necessary since the tasks have standard resource requirements. Once a task has been scheduled the resources needed to execute that task has also been assigned. If resources are fixed then as case-1 suggests the problem reduces to one of optimizing task schedules. If the tasks are constrained in terms of time, then again task allocation determines the resource usage. In this case a secondary scheduling may be undertaken within a unit-period to economize on
resource usage. Nevertheless, scheduling the tasks leave little room for manipulating resource assignment to the point that a substantial benefit may be realized.

To complete the introduction of scheduling procedures, the priority rules for scheduling tasks will now be established. However, it is essential to bear in mind that these rules reflect the preferences and strategy of the PM in contrast to the scheduling rules that have been discussed so far. Those rules are dictated by the environment and are beyond the control of the PM. Several rules and their justifications are listed in the following paragraphs.

1. Schedule tasks on basis of expense required: PM may decide to schedule tasks with the largest expense as early as possible. This may be advantageous when the cost of the task is an indication of the ability of the task to facilitate decisions, its contribution to the completion of the project or its potential to generate documents and information. In all these cases the major portion of the work is completed producing information and increasing insight on the project development. Use of such a criterion is confined to cases where resources constraints are not prohibitive.

In another context, the PM may choose to schedule tasks that have the least expense. When resource
constraints are tight and projects must proceed to satisfy deadlines, work must continue on all projects under development. Also under circumstances where the commitment of the management is not firm or is in the process of reconsideration delaying large expenditures is obviously a better idea.

2. Scheduling on basis of task output: PM can choose to schedule tasks that produce or lead to the production of a system document. This is especially important when control over quality is vital. If the user and project management are not certain about the system specifications and are in fact willing to consider modifications to the proposed system.

3. Scheduling on basis of resource scarcity: If PM has knowledge of resource availability for the unit-periods then he may choose to schedule tasks that use resources which may not be available later on as soon as possible. By the same token tasks that need resources that may be available in the future can be postponed.

4. Scheduling on basis of precedents: Whenever makespan or throughput time is important, tasks that are followed by the largest number of tasks can be assigned as early as possible.

Once more it is important to note that all these scheduling procedures are applied after the priority
rules for the projects are executed and all schedules must satisfy time and resource constraints and precedence requirements.

To summarize the scheduling procedures, a review of what has been developed so far is given below:

First a set of data has been produced from the standard development cycle information and the proposed system projects. This constitutes the schedulable task data-file that can be represented as a project network. From here decision rules were outlined for the user and the PM to assign priorities to the various projects under development so that all the information could be organized into a single network and kept in a standard format. The concept of unit-period was introduced and the aggregate network was time-phased. Then resource availability was determined by the PM and user, who also specify a certain scheduling criterion through the decision rules provided. With the exception of PM and the user input, the process is comprised of information organization and scheduling.

It is essential to emphasize that no attempts have been made to computerize decisions, provide standard strategies or confine the PM to a specific direction. What has been done is the formulation of the process by describing the circumstances and objectives of the
system design and to extrapolate the system design effort by using different sets of assumptions and strategies.

The final product is an aid to "peek into the future" by extrapolating carefully screened organized information. Each priority rule the PM selects to use is in fact a "what if ....? " type inquiry. The responses to these inquiries are a series of schedules that are essentially scenarios feasibility of which must be assessed by the PM and the user.

It is conceivable that a total of \( (n \times m) \) scenarios can be generated if there are \( n \) priority rules and \( m \) scheduling rules. Remembering that each scheduling problem is in fact a task scheduling process regardless of the projects the tasks belong reveal the amount of processing that may be required of PMCS. In effect using PMCS to generate all possible scenarios amounts to simulating the entire systems development process under different assumptions. Before going into a detailed explanation of how PMCS operates a summary of its features and functional characteristics will be offered.

3.3.1.5 Input and Output Review of PMCS:

The main emphasis of PMCS is based on three essential characteristics:

1. A standard task file that contains standard cost and time data for over 80 tasks that span the entire
development process

2. A set of decision rules that assist the PM and the user to choose their priority and scheduling criteria. This is in fact a menu of possibilities that should be considered in a general systems design project.

3. The capability to generate schedules that cover the entire project life for each combination of decision rules.

The standard task file under ideal circumstances should reflect an organization's past performance on previous system design projects. It can also be developed by polling the industry record on various types of projects and sampling data from a large number of organizations. The initial data in this file is not as important as the necessity to update the file as internal data is developed through undertaken and completed projects. Over a period of time an organization that begins to use such a file will be able to reflect its capability and potential fairly accurately.

The decision rules reflect the possibilities that can possibly exist whenever a systems project is under development. They aim to force PM and user to make a systematic review of potential outcomes. Their premise is to raise questions and indicate general directions in search of responses to these questions. However these
rules do follow a logical pattern to exclude some "risky" or less likely possibilities. An argument in favor of this is the fact that if the number of decision rules (or possible strategies) are limited the expertise of the systems personnel will improve over time since the similarity between various projects would be extremely helpful.

Finally, taking advantage of computer technology PMCS can generate all possible schedules and hence provide insight to the PM and to the user.

It is important to note that resource allocations and scheduling modules do not employ highly sophisticated mathematical tools. Therefore there are few assumptions and idealizations inherent in such techniques. The only assumptions or idealizations come from the input from the PM and the user. Since this increases user involvement and because user preferences are reflected in these inputs the end result should not produce major disappointments.

As a final note to the summary it is possible to use the PMCS as a computerized game and use it as an iterative process of designing hypothetical systems. The insight gained from such an experience can be a distinct asset for an organization. Such employment of PMCS would enable a division manager for instance, to input his view
of an information system and to receive several scenarios on achieving his goals and the projected cost. As he moves through the development cycle with different scenarios, his understanding of cause and effect relationships is likely to improve.

The output of PMCS would be a number of reports corresponding to each schedule generated. The information regarding deviations from standard data for each task, aggregate demand for each resource type for each unit period. Total hours spent in analysis for a certain unit-period or a number of unit-periods would be available. This is a direct consequence of the way initial data have been organized and processed to produce the schedules. As has been mentioned in the previous chapters the essential feature of a project management system in MIS is the ability to account for tasks, since so little room exists for reliable estimates in an MIS environment. On this premise, PMCS has organized and processed information and as a result has rendered possible the generation of a multitude of reports. These reports have the unique feature of exposing task-resource-time trade-offs and cause-effect relationships arising from the constraints.

It is important to bear in mind that both the constraints and trade-offs are reflections of PM's and
user's understanding of their environment and therefore the reports and schedules are logical extrapolations of this environment. The point is to keep the final output of a system such as PMCS from distorting the real situation PM faces by processing the input data without a general consistent framework. PMCS is designed to achieve just that through its processing and scheduling modules while avoiding to impose decisions or strategies on the PM and the user.

Now some suggestions on the report content and format are in order. First several terms need be defined.

1. Resource accounting: monitoring the demand for and use of each type of resource over a period of time. This calls for identifying the use of each resource expended with a specific task and reporting its usage comparatively with respect to the standard data.

2. Unit-period accounting: analysis of a given unit-period in terms of the resources committed and the tasks performed. For every unit-period there would be a report generated, itemizing every task executed and resource committed.

3. Task accounting: a profile of each standard task in terms of resources used and deviations from the standard data.
4. Exception reporting: at every decision point, the PM is expected to establish thresholds of performance and when such limits are exceeded reports will be generated.

With exception reporting, the aforementioned report generators provide a procedure to monitor task performance, productivity of the project team and quality and pace of the work.
CHAPTER FOUR

PMCS OPERATION AND CONCLUSIONS

This chapter is intended to describe the operation of PMCS and should serve as a manual for its installment and operation. Also in this chapter, programming and maintenance aspects of PMCS will be addressed. Finally a general review of information systems project management and some suggestions for further research will be offered.

4.1 PMCS Initiation

First, a summary of what PMCS does will be given step by step.

1. PM and user face a set of tasks that must be executed between two decision points. PMCS generates a report that gives a summary of these tasks giving costs and duration of each task.

2. In scheduling and allocating resources to these tasks, PM and the user must take several factors into account. PMCS provides a series of objectives that point out these factors. PM and the user review these and the cases that justify such objectives.

3. PM and the user’s response to the questions mentioned in step-2 are inputs into PMCS. These questions reflect the environment and the capabilities the project team has to operate with. PMCS using these responses produces one or two possible objectives. These objectives indicate the strategy for scheduling and resource allocation with the highest probability of success.

4. Associated with each objective, is a priority rule for assigning resources to tasks.
that may be scheduled concurrently. Using each of these objectives and their associated priority rules, a set of schedules and allocation schemes will be generated.

5. Each of these schedules constitutes a plan for the stage at hand. The PM and the user have now "scenarios" that project the performance depending on user definition of the problem. The PM and the user are to study each one of these schedules choose one that looks most promising and initiate the work on the present stage.

6. At the end of the stage actual performance is reported and PM and the user now will have the chance to check their assumptions and choice of schedules against the results. Then to proceed to the next stage the PM and the user return to step-1.

The entire process is illustrated in figure 4.1.

As PM and the user go through the steps for each stage they have the option of modifying both standards file and/or their choice of objectives.(which result in schedules)

The justifications for the various objectives and priority rules were presented in chapters two and three. Also a questionnaire is given at the end of chapter two to monitor the qualitative aspects of the work done at the end of each stage.

At this point, the PM and the user have three main scenarios. These are based on the following cases.

1. Resources available are fixed and no time constraints exist.
2. Resources are available to complete the tasks but the total duration of the tasks are longer than the time allowed.
3. Both the time allowed and resources are
Fig. 4.1 Decision Process for each Stage
inadequate.

The arguments that elaborate on these cases have been presented in section 3.3.1.2. The emphasis, at this juncture, is to formulate these cases and priorities expressed by the user into a scheduling model.

In the first case, the objective is to keep resource utilization high and to control the costs.

The PM and the user can determine the priority rules by examining the circumstances they face and using the guidelines of section 3.3.1.2.

For the second case the objective is to keep makespan and idle time to a minimum. In each one of the first two cases the optimization problem consists of selecting the sequence of resources and tasks.

To schedule resources with minimum idle-time a unit-quantity resource is selected and assigned to a time slot in the schedule. This unit quantity is equal to the maximum resource quantity that the immediately schedulable tasks need. Then for the next schedulable task set maximum resource requirement of those tasks will be scheduled.

Consider the network in figure 4.2 for purposes of illustration.

After scheduling task-1 the immediately schedulable tasks are \{T2, T3\}. If resource quantities for these tasks are
Fig. 4.2 Resource Scheduling Example

Fig. 4.3 Time-phasing tasks in a network
and if \(Rs\) is the maximum amount of resource that can be assigned then the assigned resource quantity will be

\[ R = \max\{R_2, R_3, Rs\} \]

and since no time constraints are imposed, the same amount of resources can be reassigned until all the work on T2 and T3 is completed. Then the next stage consisting of T4 and T5 can be initiated.

For the second case a similar arrangement would be made for the tasks. However, this time completion times would be the variables in consideration. If \(S\) is the set of schedulable tasks then,

\[ S = \{T_2, T_3\} \text{ and } t_1 = \max\{T_2, T_3\} \]
\[ t_2 = \max\{T_4, T_5\} \]

where \(t_1\) and \(t_2\) would be assigned time-periods corresponding to the two stages.

Since the tasks in the set of immediately schedulable tasks can be executed concurrently, the tasks that have smaller resource requirements and completion times do not impede the minimization of idle-time and makespan respectively.

From a scheduling viewpoint the third case is the one that is interesting and one that exemplifies the practicality of heuristics.

In this case, there is a deadline by which all of
these tasks must be completed and there are limits on all
types of resources that will have to be used. Before
going any further some definitions have to be made.

Consider the network of figure 4.3. The tasks are
organized into four groups. None of the tasks in group
(k) can be scheduled before all the tasks of the
preceding group are scheduled. Each group constitutes a
set of tasks and when all tasks of group (k) has been
scheduled, set of tasks of group (k+1) is called the set
of immediately schedulable tasks.

- group I : \{1\}
- group II : \{2,3\}
- group III : \{4,5\}
- group IV : \{6\}

If there are n different types of resources then
\( R(j), \ (j = 1, \ldots, n) \) is the quantity of resource type j
available for the stage under consideration.

Finally, \( T(i,j,k) \) is the quantity of resource j
needed to execute task i which is an element of the set
of tasks of group k.

The primary goal of the scheduling algorithm for the
third case is to make sure only the tasks that can be
accommodated by the available resources be scheduled and
whenever a resource quantity becomes available through
the completion of any of the tasks, that particular
resource be re-assigned thus minimizing idle-time.

The steps of the algorithm are as follows:

step 1: Initialize

time = 0 (makespan of the network)
k = 1 (group or set no)
i = 1 (task no)
j = 1 (resource no)
S(i,j) = 0 for all i,j where S(i,j) is the point in the time frame where units of resource type j assigned to task i will become available for reassignment.

set of all schedulable tasks G1 = \{t(i,j,k)\} (all tasks in group k)

step 2: S(i,j) = min. \{T(i,j,k), R(j)\} for all j

where j refers to the resource type
R(j) = R(j) - S(i,j)
if R(j) not = 0 i = i+1 repeat step 2,
if R(j) = 0 j = j+1 repeat step 2.

In this way all available resources of type j will be assigned to tasks that require resource type j. Also, through this step resources that are committed are certain not to exceed the quantity needed hence preventing idle resources.

step 3: update the elements in the set k

t = min. \{S(i,j)\} for all i,j

this operation indicates a progress of t time units along the time frame of the stage. Now, the remaining work to be done for each job must be updated.

T(i,j,k) = T(i,j,k) - t for all i,j
S(i,j) = S(i,j) - t for all i,j
when S(i,j) = 0 for a given i,j the task i has no resources of the type assigned to it. Also an amount of t units of resource j has become available. Therefore:

if S(i,j) = 0 then R(j) = R(j) + t.
step.4 : At least one S(i,j) will be equal to zero after step.3 is carried out. Unless a priority rule has been established, the same units can be assigned to the same tasks.

\[ S_I = \min \{S(i,j)\} \text{ for all } i,j \]
if \( S_I = S(i,j) \) then \( S(i,j) = R(j) \)
repeat for all \( j \) such that \( R(j) > 0 \).

step.5 : update the set \( \{S(i,j)\} \)

following step.4 a new non-zero set of \( S(i,j) \) will be available.
If \( T(i,j,k) = 0 \) for all \( i,j \) stop, otherwise return to step.3.

Therefore, the resource allocation algorithm is basically a process of reducing the set of schedulable tasks until all available resources are assigned.

The following is a numerical example to illustrate the use of the scheduling algorithm. Consider the network of figure 4.3 and specifically, phase II of the network. Assuming there are two resources required for the two tasks, the algorithm is carried out as follows:

step.1 : initialize sets and indices.

\[ t = 0 \]
\[ i = 2 \]
\[ j = 1 \]
\[ k = 2 \]
\[ R(1) = 8 \text{ available units of resource type 1} \]
\[ R(2) = 12 \text{ available units of resource type 2} \]
\[ T(2,1,2) = 5 \]
\[ T(2,2,2) = 10 \]
$T(3,1,2) = 6$
$T(3,2,2) = 7$

where $T(i,j,k)$ is the amount of resource $j$ required for the task $i$.

step 2:

a.) $S(2,1) = \min \{ T(2,1,2), R(1) \}$
$S(2,1) = \min \{ 5, 8 \} = 5$

checking resource availability for other tasks
$R(1) = R(1) - S(2,1) = 3 > 0$

hence, $i = i + 1$

b.) $i = i + 1 = 2 + 1$

$i = 3$

$j = 1$

$S(3,1) = \min \{ T(3,1,2), R(1) \}$
$S(3,1) = \min \{ 6, 3 \} = 3$

again checking for resource availability,
$R(1) = R(1) - S(3,1) = 3 - 3 = 0$

since $R(1) = 0$, $j = j + 1$

$j = 1 + 1 = 2$

C.) $i = 3$

$j = 2$

$S(2,2) = \min \{ T(2,2,2), R(2) \}$
$S(2,2) = \min \{ 10, 12 \} = 10$

$R(2) = R(2) - S(2,2) = 12 - 10$

$d.) i = 3$

$j = 2$

$S(3,2) = \min \{ T(3,2,2), R(2) \}$
$S(3,2) = \min \{ 7, 2 \} = 2$

$S(i,j)$ indicate the number of time units after which resource of type $j$ will be released and be available for reassignment. In other words 5 time units after the beginning of the stage, resources of type 1 initially assigned to task 2 will be available.

step 3: update the sets and variables.

$t = \min \{ S(2,1), S(3,1), S(2,2), S(3,2) \}$
$t = \min \{ 5, 3, 10, 2 \} = 2$

t indicates the time units that will elapse before resources are available for reassignment.

to update the work yet to be done:
$T(i,j,k) = T(i,j,k) - t$
T(3,1,2) = 5 - 2 = 3
T(3,1,2) = 6 - 2 = 4
T(2,2,2) = 10 - 2 = 8
T(3,2,2) = 7 - 2 = 5

to update the resources available:
R(j) = R(j) + S(i,j)
in this instance i = 3, j = 2
R(2) = R(2) + S(3,2)
R(2) = 0 + 2 = 2
R(1) = 0, no units have been released
now, to reassign the released resources only
the tasks that use that particular type of
resource need be considered.

step.4 : only resource 2 is available for reassignment.

now, the set \{S(i,j)\} must be updated to
reflect the progress along the time frame.
S(i,j) = S(i,j) - t
S(2,1) = 5 - 2 = 3
S(2,2) = 10 - 2 = 8
S(3,1) = 3 - 2 = 1
S(3,2) = 2 - 2 = 0
since S(3,2) = 0, no resources are currently
assigned to task 3. On the other hand S(2,2) = 8
and therefore task 2 is still in execution. Hence,
it makes sense to reassign the released
resources to task 2 to keep it in execution. So:
S1 = min. \{S(2,2), S(3,2) \}
S1 = S(3,2) then, S(3,2) = R(2) = 2

step.5 : update the S(i,j) and check for the completion
of the tasks.

S(1,j) = 3
S(1,j) = 8
S(1,j) = 1
S(1,j) = 2

to check if all tasks have been completed;
if \[ T(i,j,k) = 0 \] stop, otherwise return to
step.3.

step.3 :
t = \min \{3, 8, 1, 2\} = 1
this indicates that work progresses by one
unit and the point where a unit of resource type
1 is released from task 3.
T(2, 1, 2) = 3 - 1 = 2
T(2, 2, 2) = 8 - 1 = 7
T(3, 1, 2) = 4 - 1 = 3
T(3, 2, 2) = 5 - 1 = 4
R(1) = R(1) + S(3, 1) = 1
S(2, 1) = 3 - 1 = 2
S(2, 2) = 8 - 1 = 7
S(3, 1) = 0
S(3, 2) = 1

step 4:

since the units of resource 1 will be
reassigned only S(2, 1) and S(3, 1) need be
considered.
S1 = \min \{2, 0\} = 0
since S1 = S(3, 1) = R(1) = 1

step 5:

the new S(i, j) set is as follows:
S(2, 1) = 2
S(2, 2) = 7
S(3, 1) = 1
S(3, 2) = 1
since T(i, j, k) > 0 for some i, j return to
step 3

step 3:

t = \min \{2, 7, 1, 1\} = 1
T(2, 1, 2) = 2 - 1 = 1
T(2, 2, 2) = 7 - 1 = 6
T(3, 1, 2) = 3 - 1 = 2
T(3, 2, 2) = 4 - 1 = 3
R(1) = R(1) + S(3, 1) = 1
R(2) = R(2) + S(3, 2) = 1
S(2, 1) = 1
S(2, 2) = 6
S(3, 1) = 0
\[ S(3,2) = 0 \]

**step 4:**

\[
S_1 = \min \{ S(2,1), S(3,1) \} = 0 \\
S(3,1) = R(1) = 1 \\
S_2 = \min \{ S(2,2), S(3,2) \} = 0 \\
S(3,2) = R(2) = 1
\]

**step 5:**

\[
S(2,1) = 1 \\
S(2,2) = 6 \\
S(3,1) = 1 \\
S(3,2) = 1 \\
\text{since } T(i,j,k) > 0 \text{ for some } i,j \text{ return to step 3}
\]

Hence, by repeating these steps an entire stage can be covered with a large rate of resource utilization. The sum of all \( t \)-values will give the makespan for the stage.

The major feature of this algorithm is the requirement that resources be expressed in units of time. Means of representing resources other than personnel must be devised. The justification for this limitation is the fact that the most critical resource as well as the most expensive is the personnel time in information systems development. So higher priority is to control the utility of this resource.

4.2 Programming Considerations
From a programming standpoint, the generation of sets of schedulable tasks require a significant amount of bookkeeping. In addition there is the problem of merging the user specified priority rules into the scheduling model. This potentially gives rise to as many algorithms as there are priority rules and hence increase processing requirements.

To simplify programming and also to develop a potent tool for book-keeping, the generation of sets of schedulable tasks must be treated as a different problem that in fact sets up the stage for the application of the scheduling algorithm.

The reasoning behind this approach will be apparent in a while but first a note on a convenient tool to generate different schedules. In representing the tasks as a network, precedence relationships are observed and the entire network is divided into groups. The tasks of each group could be performed concurrently and had no precedence conditions among themselves. The priority rules are based on an attribute of each task which also includes precedence relationships.

Now assuming that it is possible to order tasks within a network according to the priority rules then without violating this order if the tasks could be arranged according to precedence relationships the end
result would be a series of sets based on priority status with each set preserving precedence relationships.

Then the scheduling algorithm can be applied to each subset in order of arrangement. In this way expanding the scheduling model into a more complex one that would have to deal with set generation can be avoided.

This amounts to defining a series of scheduling problems that correspond to the priority rules then scheduling for each problem. The results of each of these problems then correspond to the scenarios that will form the basis of PM's and user's evaluations.

4.3 Conclusion

The objective of PMCS is to develop a control and monitoring procedure for information systems development. The central theme of the system proposed here is the use of a standard development cycle that would be valid for all information systems projects.

Relatively little research has been done in the area of developing a comprehensive set of standard tasks with reliable data. However, a substantial potential exists for the exploitation of standards similar to the MTM and MOST systems that are in widespread use in methods engineering.

The diversity, competitive nature and rapid pace of
change in the data processing industry have all contributed to the difficulty of attaining this goal. An additional problem is the lack of consensus on the basic strategy of project management. There is, in fact no consensus whatsoever on what variables need be controlled and which activities should be part of the development process.

Certainly without a standard data system that is derived from extensive industry-wide research, a case for the degree of success and the utility of the kind of project management system proposed here cannot be made. However, the entire scheme itself can be justified on the basis of the information generated and the decisions that are supported.

With this in mind PMCS is based on the development process proposed by Long (24). Since the objective was to originate a documentation manual, the tasks described are comprehensive. The activities that lead to and result in the production of a document are taken to be extremely vital since they contribute to information on the system.

However, it should be emphasized that serious and extensive research has to be conducted to develop standards and must be substantiated by industry surveys. Since the number and investment value of MIS projects has been increasing rapidly an industry-wide standards
development is definitely a justified undertaking.

The remaining features of PMCS (decision-making, user involvement, resource management and scheduling) have been described in detail in the previous chapters. At this time it would be worthwhile to explore the potential research areas in light of the ideas furthered in this thesis.

The type of decision making suggested here (i.e. feeding assumptions and a description of the environment to an interactive model and generating plans on an iterative basis) is the focus of significant amount of research. "Decision Support Systems" (heretofore referred to as DSS) are being developed at universities and corporations as an extension to the MIS. Most DSS are being designed in areas other than information systems, financial, manufacturing, research and development activities getting most of the attention. A great number of these systems are based on extensive mathematical modelling and simulation. Computing facilities and enormous amounts of quantitative data are being utilized to plot and test corporate strategy, to devise new ones, to describe various phenomena and training professional personnel in many areas.

The success of DSS is attributable to their capability to deal with decisions under uncertainty and
generate alternatives around which a consensus could be built. Exposing the variables and complexities of a situation in an iterative manner and extrapolating available data under a variety of assumptions seems to be the most popular feature of such systems.

However, most of the DSS in use today have been geared to the treatment of data that can be handled in a statistical and mathematical manner.

Researchers have so far confined themselves to situations where both the uncertainties involved and the description of the environment could be expressed statistically. Even though DSS designed to accommodate qualitative decisions do exist, the enthusiasm and acceptance for such systems is not encouraging. In this context, information systems area remains to be an uncharted course for DSS applications. However, given increasing amount of expertise and sophistication on part of the user, venturing into DSS development may be justified. Hopefully PMCS will be a first step (however confined in scope it may be) in this direction. In addition, the success of quantitative DSS should motivate the MIS community to develop standard data systems since the existence of such standard data will facilitate the use of quantitative methods with increased reliability. As detailed data is developed and standardized and as
more comprehensive DSS are developed along the lines of PMCS, information systems planning and modelling can expand to cover information systems management and maintenance, in addition to project management.

It is inevitable that the role and influence of information systems in corporate planning will expand. Therefore it is essential that more sophisticated and reliable tools and methods used in planning and managing information systems be developed.

PMCS can serve as a starting point to this end by expanding the number of cases covered by the objectives and priority rules. Such an expansion should be based on empirical research and industry-wide surveys to determine major issues MIS managers face. Also the number of resources, PMCS monitors can be increased.

Once again, the resources that seem to be the cause of problems has to be determined from the practical experience accumulated in the industry. An additional area of research is a classification of projects according to their functional characteristics. For instance information systems that support a manufacturing function have to be evaluated by different criteria than the one that support financial functions. The respective utility and qualitative assessment of these systems depend on totally different factors that arise from the
nature of the function they are designed to serve. A procedure to incorporate these functional characteristics into the decision tables that PMCS uses to determine objectives and priority rules would have to be established.

Finally, the general framework described for PMCS in this thesis is well-suited for exploiting information processing software technology that has exploded over the past decade. The information PMCS utilizes and generates can be organized into a CODASYL type data-base. The decision tables, strategies and objectives corresponding priority rules and output schedules with standard data can constitute a data-base that could accommodate information on functional areas such as manufacturing and accounting.

Interaction with various users and user-initiated queries about potential projects could be possible on a larger scale. Programming for various applications and using a variety of languages including query languages would increase the utilization of MIS as a resource and serve to justify the investments made in the area.

It would be possible for instance to make long range plans for the manufacturing department with the knowledge of the resulting demand on information services these plans would produce. Then MIS department can plan to
meet this demand well before it is imposed on it. Conversely, if reliable and convenient information is made available to manufacturing department regarding the potential of MIS, an additional incentive to plan for expansion is created. A data-base schema that aims to integrate the data PMCS utilizes is presented in figure 4.3.

It is important to emphasize once more that the main advantage of a software such as PMCS is the capability to use standard data to provide an iterative decision-making procedure.

The procedures outlined here are simple do not involve large-scale complex models and are designed to show that similar systems can be developed within every organization by analyzing and organizing the data that already exists in a manner similar to the one suggested by PMCS. The standard development cycle adopted, is simple however, comprehensive and representative of the project cycle of most concerns.

Until further standardization is undertaken by the industry, this development cycle is recommended and is sure to be a convenient starting point for establishing standards within the user's organization.

Potential of PMCS type software is most promising to organizations that are involved in consulting in the MIS
field. Since such organizations are exposed to a variety of user environments, they happen to be in a unique position to develop standard data as well as decision models of practical value.

It is the hope of the author that steps will be taken that will lead to the proliferation of decision models for the data processing industry and that this thesis will serve as a catalyst in the process.
REFERENCES


2. Baker, Kenneth 'Introduction to Sequencing and Scheduling' John Wiley and Sons, 1974


9. DeMarco, Tom 'Structured Analysis and System


18. Johnson, Robert L. and Derman, Irwin H. 'How
Intelligent is Your MIS*, Business Horizons, February, Vol.13, No.1, 1970


121


32. Murdick, Robert G. and Ross, Joel E. 'Information Systems for Modern Management' 2nd Ed. Prentice-Hall 1975


35. Smith Jr., William A. and Wechsler, Ben


123
APPENDIX

Work Flow Diagrams for the Development Cycle
WORK FLOW DIAGRAMS

PHASE III - EVALUATION AND ONGOING OPERATION

1. REVISE COST REPORT
2. POST IMPLEMENTATION SYSTEM EVALUATION
3. POST IMPLEMENTATION REPORT TO ISPC
4. SYSTEM REVIEW SCHEDULE
5. SC APPOINTS REVIEW COORDINATOR(RC)
6. AC REVIEWS SYSTEM DOCUMENTATION
7. AC Conducts User and DP INTERVIEWS

A

8. Compile Summary of Periodic Review
9. SC & UH REVIEW
   NO
   ACTION
   REVISION RECOMMENDED
10. FILE SUMMARY OF PERIODIC REVIEW

11. SCOPE OF REVISION
    MAJOR
    BEGIN PHASE I OF SYSTEM DEVELOPMENT
    END OF PERIODIC REVIEW

12. BEGIN PHASE I OF SYSTEM DEVELOPMENT

13. SC AND UH REVISION PERSONNEL AND TIME SCHEDULE

14. MAKE REVISIONS TO SYSTEM

15. UPDATE DOCUMENTATION

END OF PERIODIC REVIEW

THE FOLLOWING SEQUENCE OF ACTIVITIES IS CONDUCTED PERIODICALLY ACCORDING TO THE SYSTEM REVIEW SCHEDULE, 111910.

END OF PERIODIC REVIEW

131
VITA

The author was born in Ankara, Turkey on July, 20
1956. He completed his elementary and secondary education
in Ankara, graduating from Ankara College High School in
1974. He attended Middle East Technical University in
Ankara until March 1977 and transferred to Lehigh
University where he received the bachelor of science
degree in industrial engineering in 1979. He has been
enrolled in the Graduate School at Lehigh University ever
since and has been working as a research and teaching
assistant.