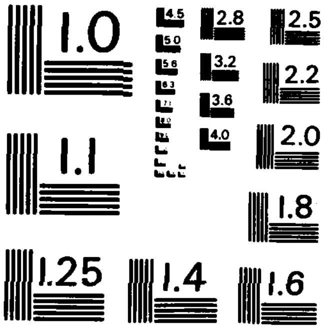
AD-A161 690 DECISION SUPPORT SYSTEM FOR ASD (AERONATICAL SYSTEMS DIVISION) PROGRAM MANAGERS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF SYST.. 1/3

UNCLASSIFIED T W BROTHERTON SEP 85 AFIT-GSM/LSY/85S-5 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

DECISION SUPPORT SYSTEM
FOR
ASD PROGRAM MANAGERS

THESIS

Terrence W. Brotherton
Captain, USAF

AFIT/GSM/LSY/85S-5

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSM/LSY/85

DECISION SUPPORT SYSTEM
FOR
ASD PROGRAM MANAGERS

THESIS

Terrence W. Brotherton
Captain, USAF

AFIT/GSM/LSY/85S-5

Approved for public release; distribution unlimited

The contents of the document are technically accurate, and no sensitive items, detrimental ideas, or deleterious information are contained therein. Furthermore, the views expressed in the document are those of the author(s) and do not necessarily reflect the views of the School of Systems and Logistics, the Air University, the United States Air Force, or the Department of Defense.

A-1

AFIT/GSM/LSY/85S-5

DECISION SUPPORT SYSTEM
FOR
ASD PROGRAM MANAGERS

THESIS

Presented to the Faculty of the School of Systems and Logistics
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Systems Management

Terrence W. Brotherton, B.S.
Captain, USAF

September 1985

Approved for public release; distribution unlimited

Acknowledgments

The Program Manager's Decision Support System was designed to allow the program managers in ASD to more effectively use their Zenith Z-100. These are powerful computers which are almost solely being used as "dumb" terminals or word processors. After conducting a literature review, I sought out existing software and a generic ASD program manager to be the prototype user. In both these areas I received extensive assistance from personnel assigned to Wright-Patterson AFB.

Lieutenant Robert Carringer of the Integrated Computer Automated Manufacturing branch of the Materials Laboratory supplied me with a data tape containing over 30 floppy disks full of automated Operation Research techniques. With this large number of techniques I was better able to pick the proper techniques to be included in the DSS. Without his assistance, I am doubtful I could have produced the same quality of a system.

Major Mary Camblin was the prototype user for the DSS development. Her guidance during the design phase and redirection on each iteration enabled the DSS to be acceptable to a large number of prospective users. It has subsequently been successfully demonstrated to the two letter SPO chiefs of RW and TA. Throughout the DSS design, Major Camblin assisted with whatever was needed. Besides being the prime prototype user, she provided a Z-100 computer to conduct development work on and made the necessary arrangements for the high level briefings. Due to her assistance this thesis effort will be used by the SPOs of RW and TA.

Lastly, but definitely not least, I wish to thank my wife Stephanie and our three girls. During this arduous undertaking, many times Stephanie had to carry-on as a single parent. Her support, understanding and cheerful acceptance of this lot enabled me to devote the attention I deemed necessary to my work. Our girls carried both of us thru the AFIT experience. The cheerful and playful pursuits of Shaye and Jessica pierced the doldrums of academic life like a breath of fresh air. Finally, I wish to thank my youngest daughter Terrell, whose nighly attentive ear while being rocked to sleep, gave me a place to vent my daily frustrations. My family's support enabled me to view this academic experience with the proper perspective.

Table of Contents

List of Figures

Page

List of Tables

AFIT/GSM/LSY/85S-5

Abstract

The Program Manager's Decision Support System was developed to enable program managers to use their Zenith Z-100 computers on program management problems. This thesis effort identifies the program management tasks most amenable to computerization, researches existing implementation of the identified tasks, and incorporated selected implementations with a user friendly interface.

The thesis is a combination of reviewed literature and the demonstration of the prototype concept. The literature review concentrated on the program management environment, the application of a Decision Support System (DSS) to that environment, Information System design factors related to development of a DSS and the evaluation of Information systems. A prototyping effort ensued to insure that the system would meet the requirements of the prototype user.

The DSS prototype was demonstrated to two sub-groups of generic program managers at ASD and AFIT. Using a developed evaluation instument, they evaluated eleven qualities of the DSS. The evaluation was composed of the three subcategories of system worth, system quality, and user propensity to use the system. The DSS was favorably received by both groups of prospective users.

DECISION SUPPORT SYSTEM FOR ASD PROGRAM MANAGERS

I. INTRODUCTION

Background

The Defense weapon system acquisition process is a multifacet, multi-dimension process requiring interaction and decision making with different functional areas. The Air Force Program Manager must, while keeping his eye on objectives, balance the requirements among "technical, cost and schedule parameters" (10:viii). The program management environment is inherently uncertain, requiring many unstructured and non-routine decisions. The Program Manager is constantly balancing the goals of controlling the cost of the program, insuring it is on schedule, will perform to the operational specifications needed and can be easily and efficiently maintained. To complicate the job even further, decisions about these trade-offs recieve mixed reaction from the many "factions" of the Program Manager world. The Program Manager must maintain a good working relationship with: Air Force Headquarters, Air Force Logistics Command, the using command, the contractor and the contracting agency. Since these pressures are interrelated the Program Manager is forced to make many varied and diverse decisions. Many of these decisions are complex and require extensive information processing. The quality of the decision depends on the depth of the program manager's analysis.

Recently, management has attempted to use the support of others to make better decisions. "Operations Research (OR) is an approach to this problem that resulted from the experiences of the Allies during WWII" (55:44). It is comprised of many analytical techniques which assist the decision maker to obtain the optimum objective. Many of these techniques have been coded into computer usable form by other military organizations.

To aid the Program Management Offices of the Aeronatical Systems Division (ASD) make the complex decisions required of them, ASD has implemented the Automated Management System (AMS). The AMS system is a massive collection of mini-computers and micro-computers. In FY84 there were 503 Zenith Z-100s procured to assist the AMS system, and another 500 systems are planned for FY85. These systems were justified "to be utilized for matrix management, provisioning, packaging, handling and transportation tracking" (2:1). Unfortuately, there is no software currently available on the micro-computers to accomplish these tasks. They are currently planned to "down load some of the work off the AMS computers" (59:1). The Z-100s are primarily planned to be used as simple word-processors. With the mass distribution of the Z-100s to System Program Offices (SPOs), the Program Manager has at his disposal a capable computer resource which could support the decision making process.

Statement of Problem

The purpose of this research is to develop, demonstrate and assess a Prototype Decision Support System that will help Program Managers at ASD use their Z-100 computer systems as decision aiding tools. The research identifies information and decision processes performed by the program manager which lend themselves to computer support.

Investigative Objectives

The specific research objective of producing this decision support system can be broken down into five subobjectives:

1. Identify those tasks that the Program Manager currently accomplishes which could be assisted by applying decision tools. The selected decision tools would be prioritized according to the criterion of being the most useful to the Program Manager.
2. Research implementations of the most usable decision tools. Concentrate on methodology, assumptions, and ease of use.
3. Design a user friendly Driver/Interface to bridge the gap between a ASD Program Manager and the selected tools.
4. Implement the selected software version of the selected decision analytical technique within the decision support system on the Zenith Z-100.

5. Obtain feedback from ASD Program Managers on the quality and suitability of the designed system.

Scope and Limitations

The research interviews supporting investigative objective 1 will be limited to personnel located at Wright Patterson AFB. Interviews will be conducted with ASD Program Managers, AFIT instructors, and AFIT students with Program management experience.

The research for computer usable decision tools will be limited to software available without charge. This includes software authored by other government agencies and public domain software.

The final version of the decision support system will be limited to operatior on the Z-100 system under the MS-DOS operating system. Since this is the Zenith implementation of microsoft's operating system MS-DOS, the system should be usable on most MS-DOS computers. The configuration of the target system is as follows: 192k RAM, two dual density floppy disc drives, a monochrome monitor, and printer. The printer is connected to the system using the standard MS-DOS parallel interface port. The Z-100 needs the printer connected via the J-3 port in the rear.

Assumptions

1. Decision tools can in-fact help Program Managers do their work more efficiently and effectively.

2. There is a variety of different implementations of the decision tools needed by Program Managers available to the researcher.
3. The Program Manager has a rudimentary knowledge of system analysis and management techniques.
4. Program Managers will have access to the developed decision support system.

Definitions

Decision Support System. The organization of usable analytical models and data bases in a fashion that enables the Decision Maker to apply his/her judgement to Semi-Structured problems for effective decision making. For the purposes of this research, computer based systems will be the only decision support systems addressed.

Unstructured Decisions. "Unstructured decisions are those that are either not capable of being structured or that have yet not been examined in depth and so appear to the organization as unstructured" (AF:a-13). An Air Force example of a program management unstructured decision occurred recently with the B-1 bomber. The Environmental Protection Agency attempted to close the Palmdale B-1 painting facility due to excessive paint emissions. The Program Manager was faced with the prospect of a government induced stoppage of work on all succeeding B-1s. This type of problem is definitely unstructured. An automated system might be able to assist the

Program Manager with pieces of the solution, but the Program Managers insight, creativity, political influence and the ability to reseach EPA regulations were the factors that enabled him to implement the proper solution (from the AF vantage point).

Semi-Structured Decisions. Managerial judgement alone is not enough to solve semi-structured problems. Large computational processes must be accomplished to support the manager. The manager needs to guide these processes and interpret them. Semi-structured decisions are those that the decision support system can be most useful on. The computation ability of the computer complements the managers insight to solve the problem at hand (27). ASD Program Managers regularly face these types of problems. Program Managers are expected to know the required amount of time needed to field their system. The programs at the ASD program office of RW are composed of approximately 50 distinct activities. The time duration for each of these activities have a wide variance. For example, the completion of the Program Management Plan could occur in 20 days or take as long as 60 days. A DSS could assist the Program Manager by doing the schedule calculations. The Program Manager remains in the 'driver seat' although, since he/she inputs estimates concerning the activity duration and the activity sequencing.

Structured Decisions. Structured decisions are straight forward. Once the structure is known the manager can delegate these to either a subordinate or to an automated system to carry out. There are not many examples of structured decisions in the program management world. One of the few occurs during the contract writing activity. Specific clauses must appear in contracts that are over specified dollar threasholds. When the program costs reach these limits the personnel in program control inform contracting and the clauses are added.

II. LITERATURE REVIEW

Program Manager

The Program Manager is the single individual responsible for the process of successfully acquiring defense systems. This process requires interface and decision making with different functional areas and disciplines. The Air Force Program Manager must balance the requirements of "technical, cost and schedule parameters"(10:viii) while insuring the objectives of the program are met. This requires many unstructured and non-routine decisions. Parameters need to be traded off against each other. The reliability of the system can be improved, but it may impact the schedule or the cost of the program. The program manager must balance the sub-goals of controlling the cost of the program, insuring it is on schedule, will perform to the required degree and can be easily and efficiently maintained. As if this job is not challanging enough, the program manager exists in a dynamic environment. While conducting his balancing act, he must also contend with the pressures of this setting. He needs to be sensitive to external political and economic conditions. Since the threat estimate for his system is constantly being 'refined', he must also refine the system needs. Lastly, he needs to manage the internal organization for which he is responsible. This brings with it a whole range of new problems to be solved. All of these pressures force him to make many varied and diverse decisions. Many of these are complex and require extensive information processing. The quality of these decisions depends on the depth of the Progr a Managers analysis. A decision is more apt to be correct if the depth of analysis is increased (21:a-8).

> The depth of analysis which can be achieved with a decision support system is very much greater than that normally considered possible when no such system is in use (21:a-8)

Baumgartner states that program management is "one of the most complex, demanding, and rewarding tasks in government" (10:6). He further states that the Program Manager must "develop plans and controls that provide adequate visiblity" (10:6) into his program. There are numerous analytical tools available which could assist the Program Manager to better visualize the progress being made on his program. "What he needs to know is whether, particularly during development, he is getting adequate progress or value for the money spent" (10:6).

The Program Manager is expected to guide his program to attain the desired goals. His project is characterized by:

- o Stringent time, cost, and technical performance requirements exist.
- o The undertaking is of greater complexity or scope than normal.
- o Significant contribution is required by two or more functional organizations.
- o The rewards of success or penalties for failure are particularly high. (10:4)

The success of a project depends on the ability of its manager to plan, monitor and track the needed steps. The manager needs to exercise these skills in order to achieve the project results given the time, and resource constraints placed upon him. Project planning includes identifying the needed steps and the process of sequencing these steps in the proper order. The amount of resources (everything from TDY budget to required military and civilians to the raw materials and cost of the system) need to be estimated, as well as scheduled.

Monitoring is concerned with the present implemention of the plan. The manager needs to react to variances in the plan and revise the plan to meet the specific goal.

Project tracking is historical in nature. The variances of actual performances are compared to the earlier plan to determine how efficiently the project is proceeding (17:24).

Air Force project management encompasses many tasks to be performed by the Project Manager. They need to make tradeoff decisions on Engineering Change Proposals, evaluate multiple contract proposals for source selection, generate technical and schedule risk estimates, be an advocate of the program to Air Staff and in general 'keep on top of things'.

Program Management Tools.

While attempting to find literature specifically related to project management analytical tools, it was discovered that most were networking or networking based (e.g. PERT, CPM)(57:46). Many of the other analytical tools which could be used during project management, have not been specifically addressed to this implementation. Project management encompasses many tasks to be performed by the Project Manager. Networking tools will help with many of these problems, but other Management Science (MS)/ Operations Research (OR) tools can also assist.

> Only within the past few years have techniques been developed for giving the Project Manager this vital information. Tools are now available whereby he can determine, with considerable accuracy and to as low a level as he needs, cost status, trends, and the cost impact of problem areas (10:7)

The Research and Development management community is beginning to accept and use Management Science techniques (32:971). They are using: "GANTT charts principally for project control" (32:971). "PERT/CPM for scheduling and control and decision analysis for project evaluation" (32:971). Literatore and Titus (32) found inexperienced corporate managers tended to compensate for their lack of experience by using Management Science analytic techniques more than their experienced counterparts. "Thus with increasing management experience, the typical R&D manager tends to rely more on interpersonal relationships and the knowledge of his staff's capabilities than on formalized scheduling and control techniques" (32:968).

There was a wide range on the types of tools used by the R&D community. Liberatore and Titus found that "nearly all of the[ir] respondents use a few of the standard measures of financial analysis" (32:970). These managers used discounting techniques to screen and choose R&D projects. "Discounted cash flow analysis is often used selectively for those projects where cost and rewards can be estimated with some certainty" (32:970).

Scheduling techniques are available to assist the program manager determine the task interrelationships. The Program Evaluation and Review Technique (PERT) and the Critical Path Method (CPM) are among the first developed scheduling tools. They were developed in the late 50's to assist with Department of Defense (DoD) acquisition programs. "By 1962, both within the Department of Defense and industry, there existed many volumes of PERT directives, procedures, and accounts of use" (19:74).

The DoD fully embraced these techniques and used them successfully on several major system acquisition programs. The most widely publicized PERT success was its use on the Navy Polaris program. It was also used on the C141 program. The C141 program director praised its use:

> Without PERT, one could envision numerous program delays of serious nature facing the SPO. At this time - there are a great many plans that have been revised as a result of PERT to become compatible with the overall program (19:74)

The director points to one of the benefits of using PERT or CPM. These techniques force a user to think in terms of the whole program instead of individual activities. "Developing a network forces thinking through the entire project from beginning to end" (19:77). The user describes the activity dependency relationships while using the technique. If an event requires another to be complete before it can start, the manager may want to increase attention to the prior activity.

Although "R&D managers see PERT as the best or one of the best project planning tools available" (19:77), many are hesitant to use it themselves. Several of the R&D managers surveyed by Liberatore and Titus were "not completely satisfied with the available techniques for project monitoring, scheduling and control" (32:971). They were interested in a user-friendly system which would contain "up-to-date data and the ability to obtain information concerning project costs and milestone progress with a modicum of computer-related experience and effort" (32:971). The PERT era of the early 60's came to an end when the DoD switched official scheduling systems. One of the reasons for the demise of PERT is the difficulty in entering and updating data of network nodes. "The necessary tasks of data gathering and modifying a detailed network plan are time consuming and cumbersome. Day-to-day progress review can be accomplished more efficiently" (19:77).

Several commercial micro-processor based network software packages are beginning to appear on the market and in the literature. These are aimed at the project manager.

> Once confined to university business classes of defense contracting using mainframe computers, computerized project management techniques are gaining acceptance and manual systems are being replaced by PC's and other personal computers (17:241).

Dauphinais and Darnell recommend some attributes that program managers should consider when acquiring a project management tool. They state that the tool will need to assist the manager plan, monitor and track projects. To plan, the manager needs to identify tasks and their relationships. This includes "milestones and deadlines, and estimating what resources" (17:241) are required. The system should be adaptable to assist with the monitoring function. The manager needs the ability to react to change and modify the schedule in real-time.

The planning portion of the project management package should be very capable. "Planning is probably the most essential part of project management" (17:242). For planning, Dauphinasis and Darnell recommend that a package: be tied to a calender, have the ability to define required task resources, show subordinate levels of detail, allow partial tasks to be prerequisites, show multiple task resources used by an activity, and portray the results with graphics. To enable the manager to monitor the program, they recommend the tool have the ability to update and modify the appropriate data base.

Operations Research.

Management's job includes using the talents of those under their control to make better decisions. Operations Research (OR) found its birth during World War II to help management quantify large problem areas. "Management Science (MS) is a later profession much like OR" (55:44). Both are used to help the decision maker. OR techniques tend to be more analytical while MS is more concerned with the soft sciences of organizational behavior and interaction. Although these disciplines have been developed to aid decision makers, it is well recognized in MS/OR literature that as a whole, it is difficult to implement their results(55). The prime complaint with existing MS/OR systems is they are not designed to ease user involvement. Dialog between the MS/OR specialists and the user is strained since the specialists do not understand the user's environment. Operations Research is applicable for well structured decision problems.

> It comprises the techniques of modeling; statistical analysis; computer simulation; resource allocation; optimization and [mathmatical programming be they] linear, nonlinear and integer programming (55:44).

The program manager requires a diverse library of operation research techniques to enable him/her to choose the tool suited to the problem at hand. Many questions are answerable by the use of statistics or a data base management system. Others require interaction with "complex algorithms (e.g. linear regression, forecasting techniques, optimization)" (20:73). Lockett conducted a survey of managers which had taught themselves MS/OR techniques to use in their work. By and large these managers were from environments very similar to that of the military program manager. The projects on which OR projects were used were contained within a single department. The manager was in total control of the project and therefore had more flexibility concerning the methods used. All the self starters also had easy access to the required OR computer facilities. The results indicated that the MS/OR self-starters viewed the available analytical techniques as useful and not frightening or threating. With "the increasing availability of cheap computerized systems, they see OR as something that should be part of their tool kit ... For example, linear programming, simulation and critical path analysis" (34:61) are tools which can benefit the manager to better perform the decision making tasks of his job. The program manager can use the DSS to assist with many of these tasks. Data relevant to proposed Engineering Change Proposals (ECP) can be organized, dissected and traded off against other ECPs to evaluate which is more beneficial to the program. The DSS is ideally suited to assist the Program Manager keep tabs on the program schedule. When the manager identifies that an activity will not be completed on schedule, this data can be input to a DSS to identify the impact of the slip on the entire program. Subsequent penalty costs or withholding of payment for the activity slippage can be justified with this data. Other DSS applications range from data base manipulation to exercising operations research models.

Since the computer tools are distinct and separate from each other, the ultimate power within the applications are not being used. The Data Base Management System (DBMS) and analytical tools can be thought of as building blocks. With the proper 'glue', the blocks can be combined to satisfy unforeseen requests from the Program Manager. This 'glue' is the Decision Support System.

> The integration of traditionally separate tasks (such as spread sheet analysis, data management, and program modeling) releases a previously untapped dimension of micro-computers power (20:65).

In Liberatore's survey, respondents from "Fortune 500" companies were asked about their use of quantitative techniques for project management. The respondents indicated they relied heavily on financial methods for project selection, but did not effectively use the other available operation research techniques. These were not used more extensively because of user dissatisfaction with the implementation of the techniques. They indicated that they were interested in using the tools but required a user friendly system to use them (32).

In Wynn's review of the Decision Support System literature he encountered a 1982 article by Vazsonyi. Vazsonyi gives the motivation for Management Science/ Operations Research specialists to move toward to the application of decision support sciences:

- The methodology of DSS is the application of the scientific method of decision making
- DSS provides specific guidance on how to integrate electronic models into the decision making process.
- DSS definitely fills a need and there is a market for DSS. (58:55)

Decision Support Systems

Decision support systems are designed to combine the theory of decision assistance with the reality of problem solving. They are specifically tailored to solve complex and semistructured problems often faced by but not limited to upper management. The goals of the DSS are:

- To assist managers in the decision making process when dealing with semi-structured problems;
- To support,rather than replace managerial judgement;
- To improve the effectiveness of decision-making rather than the efficiency (57:3)

The four elements of a decision support system are the decision maker, the DSS interface/driver, the bank of analytical techniques and the user data base. The bank of models enables the user to implement the correct technique for the problem at hand. The models are executed by the driver portion of the system in accordance with the desired result for the program manager. The data base includes information pertaining to the application area and the usability of the models and external data. The Data Base Management System (DBMS) extracts the data needed by the selected model from the user's data bases and adds the result of the model execution to the appropriate field in the data base (AF:). A decision support system is just that; a decision support system. It is not designed to alleviate the decision maker from his responsibility. It is there to aid him. Since the decision maker exists in a fluid, changing environment, the system must also be flexible (6).

Decision Support Systems are characterized by their users, orientation, focus and emphasis. These systems are used by persons within an organizations line of control and management. These users are interested in making the right decision concerning their programs or the effectiveness of the organization. The decision support system incorporates analytical models with past data and managerial insights to help the manager structure the problem. With the increase in problem structure the DSS can assist the manager to generate options or alternatives. Lastly, given these options, other analytical models can aid the manager in making his choice and can predict future outcomes. The system focuses on the future of the firm. Since the DSS is concerned with a changing environment, it must be flexible. It needs to use the 'building block' models in varied and diverse ways (6).

DSS Disciplines. Among the many disciplines that need to be included in the Program Manager's Decision Support System are interactive decision processes and OR analytical models implemented through computer science data base and model management systems.

> Being user orientated, decision support systems require almost instantaneous response times, interactive entry and display devices, and appropriate operations research routines (forecasting, statistics, simulation, ect) (57:3)

The interactive decision processes are aids to the manager. These methods help the decision maker structure his decision making process so all relevant criteria will be used. The decision maker can structure multiple diverse criteria into a framework to help him make his decision. After a decision has been proposed by the system, he can conduct sensitivity analysis to determine how sensitive the decision is to varying the importance of different criteria (51).

Operations Research studies have developed many analytical methods which the decision maker can use to aid his decision. Pritsker and Associates (1) recently completed an indepth analysis and software gathering effort for the Integrated Computer Assisted Manufacturing (ICAM) office of ASD. The analytic techniques they discovered and their applications appears in Table I.

TABLE I

DECISION SUPPORT SYSTEM APPLICATION TECHNIQUES (1:3)

BABALB	- An analysis which determines an optimal grouping of the operations of a production line.
CANQ	- Analytically solves a closed network of queues for long run average system performance.
CEAP	- Analyzes a capital expenditure, taking into account investment amount and timing, useful life, depreciation & salvage
EOQ	- Calculates economic batch (lot) sizes for manufactured parts. Assumes the demand rate is constant and continuous.
GERTE	- Analyzes networks which involve probabilistic branching & stochastic activity duration.
INV	- Calculates economic order quantities and reorder points for an inventory system.
JOBBAL	- A heuristic analysis which determines a good assignment jobs to resources.
LEARN	- Calculates learning curve factors.
MIP	- An optimal solution to an integer program problem
NETFLOW	- Solves the problem of minimizing the cost of flow thru a network.
NETSOL	- Analytically solves a network of queues for long run average system performance.
NPVROR	- Calculates the rate of return and net present value for an investment.
PERTCP	- Project planning using either Critical Path Method or the Program Evaluation and Review Technique.
RESALL	- A heuristic analysis which allocates scarce resouces to a Critical Path type of project.
SCHED	- A heuristic which generates a schedule for a multiple job, multiple machine job shop.
STAT	- A family of statistical analysis techniques.
XMP	- Finds the optimal solution to a linear program.

With these methods the decision maker is better able to conceptualize the meaning of the data from the real world. Statistics, including trend and regression analysis help the Program Manager determine where his program is going and what the driving factors are. Mathematical programming models, including linear, non-linear and goal programming, enable the Program Manager to realize the most efficient method to allocate resources in attaining his goal (51:5). DSS have refocused research and attention away from the study of models to the problems those models are designed to solve (56:83).

Computer Science is the gel which ties together the analytical decision techniques into a Decision Support System. The techniques are programmed for use on the computer. Data bases are organized to facilitate rapid combination of data into information usable by the Program Manager.

> Data management refers to the organization, creation, maintenance, retrieval, integrity, and security of the data (20:69).

The results of manipulation of this data by the analytical techniques is displayed in easily understandable graphic form for the user (51:5). The DSS helps the manager manipulate MS/OR tools to fit his specific problem (25:118). Simon states that "in decision support systems, models are only some of the components that may be utilized" (56:83). Computer science organizes the tools, and data bases so the resultant system is more powerful than single model implementations.

DSS Environment. According to Wynne (58:55) DSS are advantageous to MS/OR and Management Information Systems because it is more a process than a strict structure. He states "DSS are, when properly done, a combination of quantitative, behavioral and information sciences" .

> The typical DSS thus aims to provide the decision maker specific information in response to specific requests bearing on specific decisions. (54:47)

Wynne believes that the implementation of Decision Support Systems have narrowed the gap between the decision maker and the developed MS/OR discipline. The DSS allows the user to easily access the needed data and manipulate it with the desired tool. His definition of DSS includes the "DSS' impact...on the decisions where sufficient structure exists for analytical aids to be of value but where management judgement is essential" (58:52).

The Program Manager's decision problems can be described using the Keen and Scott Morton Needs Analysis. This approach classifies problems according to their Management Activity and the Structure of the decision environment. The structure dimension has the three levels of Structured, Semi-structured, and Unstructured. Structured decisions are straight forward. Once the structure is depicted a subordinate or a computer can carry out the decision. Semi-structured decisions require the intuitive judgement of the manager and his decision framework to make the decision.

The other dimension concerns the degree of management activity. The activity levels are operational control, management control and strategic planning. In the literature, strategic planning analysis was the concern of large entrepreneurs. The need for operational control and management control is seen as a more conducive environment for the implementation of a DSS (21:I-10).

DSS Characteristics. In Geoffrin's outgoing speech as president of The Institude of Management Science (TIMS), he chastised the membership to be more responsive to DSS because of its qualities. Since a DSS can be used on ill-structured problems, it can be used when traditional MS/OR techniques can not. Even more important is that the DSS "puts the user first, and the underlying technology second" (58:54).

A Decision Support System should be flexible and adaptive to the manager it serves. The models can be combined to meet the changing needs of the manager and his environment. The ideal system is one that the manager can adapt to mesh his own decision making and judgement process (25). The Program Manager may need to do exploratory queries of large data bases. He may wish to distill only a selected piece of information from the data base. For example, he may want to know which contractors are behind in submitting their Cost Scheduling Control System Criteria Cost Performance Reports (C/SCSC CPRs). This involves unanticipated inquiries to the

DSS. The DSS should be able to look at the appropriate data base and retrieve the needed information for the Program Manager.

Information Management. Blanning (11) is particularly optimistic about the potential of coupling Data Base Management Systems and DSS model management. He feels these two sciences together will produce a "comprehensive framework for information management in DSS" (11:72). While explaining the current research areas in decision science, Blanning identifies the following three as important to information management:

(1) The construction of knowledge-based interactive systems.
(2) The development of frameworks for model management systems similar to those for data base management systems (DBMS).
(3) The integration of data management and model management to produce an emerging science of information management (11:71).

The decision maker taps several different types of knowledge to make a decision. Dr. Holsapple of the Purdue Management Information Research Center classifies the needed knowledge into the categories of: basic empirical knowledge, formula knowledge, and procedural knowledge. Basic empirical knowledge consists of specifics about the realm of the Program Manager. Knowledge about the relationship between the contractor and the military is an example. Formula knowledge tells the Program Manager how the existing data is combined to form new knowledge. Procedural knowledge goes a step beyond formula knowledge. It indicates what algorithms should be used to derive the new information (6:65). DSS are used to depict to the user the procedural knowledge about the pertinent MS/OR tools and environment.

> Effective integration of the various types of knowledge is accomplished when an assortment of piecemeal tools is replaced by a single integrated DSS (6:65).

In the discussion by Bonczek, Holsapple and Whinston about knowledge based systems, they state that much of the power of a DSS is derived because of its "knowledgeability about the problem domain" (13:70). The knowledge includes vast assortment of facts about the decision maker's industry. He could not possibly keep all the required data on the top of his head (13). The knowledge is organized in a systematic fashion so the decision maker may easily draw upon it.

Many DSSs manage the decision models themselves. These systems are reacting to the fact that decision models are important to the organization and should be managed. The models are described within the DSS so the relevant model may be used when it is appropriate. The advances in data base management are applied to the models. The model characteristics, dependencies, and limitations are retained within the DSS model base (11:72).

Sprague also views information management as the direction that DSS development should take. He has proposed that the DSS must accomplish the key task of DSS model generation. The goal of DSS generation is to define a system that provides a great deal of flexibility to the users in developing applications to solve problems. This flexibility is required due to the wide range of application problems that the decision maker will require the DSS to assist him with (52). Huber, who has been described by Wynne an an "organization scientist with a strong interest in the DSS field" (58:54) states that a DSS designer should strive for a system that is "flexible, friendly and provides a variety of options" (22:567).

Micro-computer Implementation

The micro-computer places an unprecedented amount of computing power directly into the hands of the decision maker. Decision support systems lend themselves to this type of implementation.

> DSS, by its very nature implies one person or a small group with a common goal interacting with a computer system dedicated to facilitate goal achievement (20:64)

This is the very reason that so many micro-computers have been procured for the ASD staff, and more are planned.

> With this buy [Zenith Z-100s], we will get the capability to create AE-wide products with automated integration and summary of information (7:1)

The availability of micro-computers which are able to support a DSS has opened the door to the building of a system specifically suited for a certain individual. Whereas before the individual had to conform to the computer system to get a usable computer product; now the system can be tailored to the user.

> Micro-computers will rapidly facilitate the personalization of data systems to individual users, allowing people to structure data to suit their information needs (12:24)

Geoffrion views personal computers as:

> a problem solving environment that can be (and is) used directly by a manager and can be tailored to the manager's personal needs (58:53)

Many sophisticated software tools are also appearing on micro-computers. These enable the manager to execute programs which required a large main frame computer. Data base management systems and high order language compilers are included in this class. Without these system level routines, micro-computers would still be just the play things of computer hobbyists. The compilers allow routines which were authored for large computer systems to be transported to micros. In Holsapple's study which included micro-computers, he states:

> In surveying existing micro software, we can see the first primitive signs of general problem processors in file management systems such as Condor and dBase which allow data management and ad hoc inquiry to be integrated with procedure execution (20:68).

Information System Design.

Design of information systems has traditionally followed a sequential flow of events from feasibility studies, system analysis, design specifications, actual computer programming, testing and implementation (49:7). "The life cycle is intended to ensure the translation of system objectives into operational systems within constraints of schedule and budget" (49:6). This type of development design is appropriate for information systems that are highly structured and have a high degree of "user task comprehension and developer task proficiency" (18:570). The development of a large business accounting system is an ideal application for this development design. The development of a decision support system is less certain and requires a different approach (18).

The amount of uncertainty in an information system development effort can be gauged by evaluating four characteristics of the effort: project size, degree of decision structureness the system is to support, user task comprehension, and developer task proficiency (18). The characteristic of project size has a direct correlation with the uncertainty of the development effort. As the project gets larger, the amount of uncertainty increases. "Large project size increases the difficulty of assuring that requirements are met because of the number of persons involved" (18:565).

The structuredness of the decision process itself has a modifying effect on the uncertainty of the system development effort. The less structured the decisions that the information system is to support, the higher the uncertainty of the development (18). Decision support systems are specifically suited for semistructured decisions. As a result, there is a relatively high uncertainty inherent with decision support system developments.

The characteristic of user task comprehension has to do with how well the user of the system understands the tasks which the decision support system is to support. If the users do not agree on the tasks which the system is to support or do not understand them, than the design uncertainty is increased (18).

Developer task proficiency is the last characteristic which mediates the level of development uncertainty. A task proficient information system developer can do his job well. This is a measure of "directly applicable experience" (18:565) of the developer.

By combining these four characteristics, the development uncertainty can be assessed. Gordon and Olson present a model which uses these development characteristics to assess the overall uncertainty of the information system development. Once the uncertainty level has been assessed a development strategy can be pursued (18).

Gordon and Olson present four candidate development strategies to use during the development of the information system (18:566). The selection of the proper strategy is dependent upon the level of development uncertainty. The acceptance assurance strategy should be selected when uncertainty is very low. Under this approach, the developer is given a set of requirements which are " mplete, correct and firm" (18:566).

With a moderate level of uncertainty, the linear (or traditional software life cycle) and the iterative assurance strategies are appropriate. Both of these require "sign offs" by the user when phases are complete. The difference between them is that with an iterative assurance strategy "whenever requirements are found to be wrong or inadequate during the development, the requirements are revised by a return to the requirement development process with the user" (18:566). With the linear assurance strategy this is not done.

The last strategy is the experimental assurance strategy or as it is more commonly known, the "prototyping" (18:567) approach. Prototyping is recommended when the development effort is highly uncertain. "The prototype design method reduces uncertainty by producing successive approximations" (18:567) of the system that the user wants. Prototyping is based on the idea that a user can specify the qualities they dislike in an information system better than the qualities they want in an elusive proposal of a system. The prototyping development process moves thru four distinct phases (18). These are: 1) Identify the users initial requirements; 2) Develop an initial system to satisfy the bare bones requirements; 3) Let the user use the prototyped system; 4) Change the system to reflect the revised user requirements. The process is continued by cycling through steps three and four until the user is satisfied with the evolved system (18,24).

Bally, Brittan and Wagner state that the "greatest advantage of the prototype strategy [is] the generation of user confidence" (9:25). They further this assertion by stating that "any information processing system must achieve both "technical" and "psychological" success" (9:25) to truly be a success. By technical success they are referring to the ability of the system to do what it was designed to accomplish. "Psychological success is the degree to which the end user has confidence in the final system" (9:25). With the prototype approach, the user learns early in the development what the system can do, and has the ability to modify the system to accomplish those tasks he truly desires. Since the final system is based on the user's actual experience, "the user is far more likely to have confidence in the final product" (9:25).

Alavi (5) conducted an analysis of the effectiveness of the prototype approach and the attitudes of both the users and the designers. He was interested in highlighting the "opportunities, problems, benefits, and shortcomings of prototyping" (5:556-7). Five advantages of prototyping were identified. First, the presence of the prototype system allows users to give more meaningfull feedback on their specific needs and requirements. It is easier to criticize the actions of a real system than to identify what they want of a mythical one. Using a prototype system gives both the developer and the user a common reference point to communicate from. User enthusiasm is captured by using this development strategy. The system is visible to the user, hence real. With an up-and-running information system, the user feels that the developer is actually supporting their requirements and interested in their needs. "They felt they had some real influence in the design process" (5:557). This in turn, helps establish better relationships between the user and the developer.

In Alavi (5) analysis, four problems of the prototype development approach are presented. These are: unrealistic user expectations, project management difficulties, inappropriate strategy for large information systems, and the difficulty of maintaining the early enthusiasm of users (5:558).

Initial prototype information systems are by definition very limited systems. If the prototype is oversold to the user, and the user places multiple requirements on the initial system, it may become overly complex and unrealistic. When the development of an unrealistic system is not accomplished on schedule, the user may become disappointed and lose confidence in the development. Further development suffers due to the users loss of confidence.

The eventual information system which is derived from a prototyping effort is unknown at the outset of the prototyping development. Milestones and the exact nature of final deliverable products required from the effort are also unknown at the start of the project. This difference from normal project management has caused some difficulties with traditional "planning, budgeting, managing and control systems" (5:558).

The last two shortcomings that Alavi points out are that prototyping may not be appropriate for large systems, and the users enthusiasm may sway. He states that large information systems are difficult to prototype because "it is not clear how a large system should be divided" (5:558). This argument conflicts with Gordon and Olson's proposition (18) that the development of a large information system has increased uncertainty and so prototyping may be more appropriate. The difficulty of maintaining user enthusiasm may cause the user to release the developer from a prototype system before it has reached it's development objective. "After high priority user requirements were satisfied by the prototype ... users wanted the team to move on to a new project"(5:558).

User Friendly Design. Many technically superior computer programs have failed to be well received because they fail in their interface with the user.

> While the technical computer literature descibes algorithms and systems that are technically effective, computer specialists have developed an informal, more private folklore of systems that were underused or abandoned because they were ineffective person/machine systems (29:41)

The conditions which should be avoided and lead to user UNfriendly systems are: designing of systems that are not understood by those they serve, systems that require excessive precision or attention, systems which are hard to modify, and systems that provide the wrong answers (29:25). By using a prototype strategy, many of these problems are discovered early in the development.

Morlan (41) points out the effect that obsolete terminal interface design is having on users of modern systems.

> Users of state-of-the-art hardware are often disappointed to find that their productivity is significantly reduced by cumbersome data entry procedures, obscure error messages, intolerant error handling, inconsistent procedures and confusing sequences of cluttered screens (41:484).

He points out some methods that can be implemented to improve the man-machine interaction.

Morlan's (41) foremost recommendation is to do the analysis of the prospective user interface early in the system design. He points out that designing an effective interface to the machine cannot be a task that is done as an afterthought. He has several specific recommendations that can be used to better the communication between the man and the machine. The most important is simplicity. To reduce the occurrence of user error, a simple screen layout lets the user know what is occurring in the information system during the interaction.

He attributes the problem of interface complexity to two sources. First, the programmer is fascinated with the intricate capabilities of the system. This factor can be thought of as a programmer 'showing-off' his technical ability.

Morlan recommends that attention and intrinsic reward be offered to those who show a "visible concern for simple, direct and easy-to-use systems" (41:487). The second source of complexity is the distribution of the machine interface task to multiple programmers. Whenever possible one person should be responsible for the user interface. When the system is so large that many people have to work on the interface, they should have the guidance of the same interface design guidelines.

Morlan (41) offers numerous suggestions to help the developer of screen orientated information systems develop a better interface. The idea of simplicity is key to a good screen design. The system will actually execute faster and the user will become more confident of its process when less is placed on individual screens. One of the methods of simplification is to eliminate meaningless phrases and words. Clarity is improved by getting rid of such social amenities as "please" and "if you wish" (41:487).

Clarity is also preserved by being consistent with screen titles. "A one-to-one correspondence between menu items and the associated subsequent screen title enables the user to easily perceive the logic of multiscreen functions" (41:487). An example of this technique follows: The user of a DSS is presented with a screen of analytical techniques. His selection is "PERT or CPM". The next screen has the selection item "PERT or CPM" as its title.

Another technique that adds to the power of the interface is using multiple colors on the terminal screen. Highlighting can be used for:

1) Linking logically related data;
2) Differentiating between required and optional data;
3) Highlighting errors;
4) and Separating prompts (41:488) [from other data]

An example of effective highlighting is to use reverse video (light letters on dark background) to let the user identify the menu item he has selected.

Another powerful feature is the use of function keys to accomplish certain actions (41:489). The use of function keys can simplify the job of both the designer and the user. If the user must activate specific actions with function keys, error detection becomes easier to accomplish. Error detection and correction is an important activity for an information system.

Error Handling. It is possible to develop information systems that "minimize both the occurrence of errors and their effect" (44:254). Norman's (44) research has concentrated on design tools for the development of interactive computer systems that reduce the likelyhood of errors. This is important for two reasons: errors can lead to serious results, and they can deter beginner users from using an information system. His main point is that people invariably will make errors, so the information system should be designed to be tolerant of errors. Norman suggests some flexible design guidelines to follow while designing information system. These are: Feedback; Similarity of response sequences; Actions should be reversible; and consistency of the system (44:257).

By using Norman's (44) design points in the development of the information system, the user benefits with a system that will be friendly to his needs and gives him a feeling of control over the system. The feedback guideline specifies that the state of the system should always be apparent to the user. When it is in an editing mode,for example, the user should be aware of this state. The similarity of response sequence guideline means that different types of actions should appear different on the terminal. The editing of data screens should look different than the selection of analytical techniques screens.

Norman's (44) principle that actions should be reversible has two components. The user should be able to reverse an action. If this is impossible or difficult to implement, for instance in the case of deleting a file, the irreversible action should be difficult to activate.

The last attribute of error tolerant information systems is that the system design should be consistent. Consistency of the system allows users to become more rapidly proficient in the system. An example is to always use the HOME key to return to a previous menu or the HELP key to give assistance to the user. Inconsistency will breed user frustration, and may deter a user from using a system.

Information System Evaluation.

Davis and Olson (18) identify the last phase of the development life cycle as a post audit. An evaluation is made using the development groups pre-development objectives and the expected cost/benefit of the project against the actual performance and the cost of development. "The results of the post audit are intended to assist in improving cost justification and management of future projects" (18:613) as well as improving the current project. One of the measures which can be used to evaluate the system is "system value" (18:613). The ideal evaluation of the value of the system would be to determine the systems affect on decision making. Since this is difficult to measure, surrogate measures are often sought. An evaluator can query the user to indicate how satisfied they are with the system. By using this surrogate, the assumption is made that a satisfied user is using the system effectively.

Another technique that can be used to assess the system is to develop a prior assessment and compare the results to the post audit evaluation. This technique reduces the problem of comparing the development effort to unreasonable expectations. This method, however is not the norm, "most evaluations of I/S [information systems] are provided only in efficiency-orientated terms on a post hoc basis by system users" (28:43). These evaluations are concerned with how well the information system does a job, not whether the system is doing the right job. Doing the right job, or the effectiveness of the system needs to be assessed. The effectiveness oriented evaluation of an information system can be accomplished by assessing the system throughout its development. "These assessments are made in terms of attitude, value perceptions, information usage and decision performance" (28:43).

King and Rodriguez (28) document an instrument developed by Schultz and Sevin to evaluate the value perception of the information system. "Value perceptions are ... more direct assessments related to specific MIS. For instance, an answer to a question such as "how good is the system?" is a value perception" (28:45). Prior to the King and Rodriguez study, much research had been conducted to quantify user satisfaction (18:614, 23:785-793).

Bailey and Pearson establish an instrument to measure information system user satisfaction. They reviewed the literature in the field and by "adapting the semantic differential scaling technique a questionnaire for measuring satisfaction was then created" (8:530). In their evaluation of user satisfaction they derived five factors that were important for information systems. These factors are: "accuracy, reliability, timeliness, relevance and confidence in the system" (8:537).

To evaluate the degree which the five factors of quality were contained in an information system, a questionnaire was constructed that used the semantic differential technique. "The semantic differential technique was developed by Osgood, Suci and Tannenbaum to measure the "meaning" of things" (8:533). In general the respondee is asked to pair an adverbial modifier with one extreme of a continum of opposing adjectives to describe their feelings about a concept. For instance, the respondees feelings concerning the adequacy of DSS training is assessed by having him pair one of the adverbs (extremely,quite,slightly) with one of the adjectives on a continuum ranging from complete to incomplete. The respondee's result would be "extremely complete" if he thought this phrase was descriptive of the training.

Using the semantic differential technique, Bailey and Pearson (8) constructed an evaluation instrument containing 39 specific user dimensions of the five important factors. These dimensions were evaluated using "four bipolar adjective pairs" (8:533). The researchers added two additional scales to those assigned to each dimension. "The first scale was the adjective pair, satisfactory-unsatisfactory. This was done to test the internal consistency of the other four pairs and thus the internal validity of the instrument" (8:534). The second was the adjective pair important-unimportant. This factor was used to gauge the relative weight of importance of the factor.

The evaluation instrument was tested for validity and reliability. A reliable instrument is consistent. It will measure an identical attribute the same on independent occasions. A valid instrument is correct in its evaluation of the attribute. The difference is illustrated by an example of a scale. A scale that always weighs a 200 lb person at 150 lbs is reliable but not very valid. The reliability coefficients of the Bailey and Pearson instrument were assessed with an average of .93. This high coefficient means very little of the variance found in the results of their survey is due to measurement error (8:536).

The validity of the instrument was also measured. The validity attribute consists of the three subcategories of construct validity, external validity and content validity (8). Construct validity occcurs when the factors being measured are the same as the factors being evaluated. The Bailey and Pearson questionnaire measures user satisfaction with the information system. "If those factors which are important to perceived satisfaction are important in the measurement questionnaire" (8:536) then construct validity is established. External validity is the measurement of a factor with an external assessment. The fifth adjective pair of satisfactory verses unsatisfactory was added to help measure the external validity. The result of this pair was compared against the results of the other four pairs. The instrument demonstrated a very high average external validity of .91 (8:536).

The last dimension of validity is content validity. Content validity occurs when all qualities of the item being measured are included in the instrument. While conducting their literature review, Bailey and Pearson identified 39 different attributes of user satisfaction. These were then paired with four adjective pairs to solicit the users perception on the quality. "The methodology used to develop the factor list and the result of the critical incident analysis suggests strong content validity" (8:535-536).

Ives, Olson and Baroudi conducted an in depth literature review and analysis of information systems evaluation techniques and instruments. They evaluated many earlier works and selected the Pearson instrument to build upon because it was the sole instrument with "adequate empirical support, which covers both the information systems product and general system services and provides multiple indicators" (23:788). Their research goals were to :

1. Replicate Pearson's findings concerning the validity of the instrument
2. Reinforce the validity of the instrument
3. Reduce the length of the overall measure
4. Develop a standard "short-form" (23:788)

The goals of the study were realized. The researchers lend support to Pearson's instrument by replicating its result. They next sought to shorten the instrument. Using statistical means as their selection criteria, the most closely predictive questions on the users satisfaction were identified and retained. Seventeen attributes were eliminated and the number of adjective pairs was reduced from four per factor to two. The researchers next validated the shortened version by extracting the original data used in the Pearson Questionnaire. The new instruments "correlation was .90 (significant at p=.001)" (23:791).

> These correlations provide substantial evidence that the short form questionnaire is a sound general measure of Pearson's original UIS [User Information Satisfaction] Concept (23:791)

The Air Force has also recognized the importance of evaluating Air Force computer systems. The Air Force Operational Test and Evaluation Center (AFOTEC) has established policies, procedures, guidance and questionnaires to evaluate the adequacy of computer systems. A Deputy for Software Evaluation is assigned to evaluate the software of new Air Force computer systems. His main duty is to test the adequacy of the computer hardware, software and user interface and issue a report on them prior to Air Force acceptance. The direct system user is an important person in this evaluation of the adequacy of the system.

AFOTEC has written a 95 item questionnaire (4) to evaluate the quality of the operator-machine interface. The desired attributes addressed in this instrument are grouped into the six categories of: assurance, controllability, workload reasonability, descriptiveness, consistency and simplicity (4).

The quality of assurability ensures that the software assists the operator in "validating data, avoiding errors and correcting errors once made" (4:2). A system with controllability allows the operator to direct the operations of the computer. An important category for DSSs is the quality of workload reasonability. This quality ensures the users abilities are not overtaxed by using the system. It is the

> design of a system which involves an operator and a computerized machine taking advantage of the best capabilities of both: the machine to perform repetitive tasks rapidly and the operator to make command decisions (4:2).

A system has descriptiveness to the extent that the operator has adequate explanation for tasks he needs to perform. Consistency is the characteristic of a system that operates as documented. The last category, simplicity ensures that the operation of the system can be accomplished without overly complex instructions (4:2).

III. METHODOLOGY

This chapter focuses on the methods used to construct a decision support system for the Program Managers at ASD. The purpose of this research is to develop, demonstrate and assess a Prototype Decision Support System that will help Program Managers at ASD use their Z-100 computer systems as decision aiding tools. Prior to actual construction of a computerized system, information about the user and his available computer resources need to be specified.

The research objectives identified in chapter I subdivided the large task of creating a DSS into five smaller ones. Each of these will be completed using its own methodology. The five sub-objectives are:

1. Identify those tasks that the Program Manager currently accomplishes which could be assisted by applying decision tools. The selected decision tools would be prioritized according to the criterion of being the most useful to the Program Manager.
2. Research implementations of the most usable decision tools. Concentrate on methodology, assumptions, and ease of use.
3. Design a user friendly Driver/Interface to bridge the gap between a generic ASD Program Manager and the selected tools.

4. Implement the selected software version of the selected decision analytical technique within the decision support system on the Zenith Z-100.

5. Obtain feedback from ASD Program Managers on the quality and suitability of the designed system.

Decision Support System Environment

Specification of the decision support system intended environment encompasses the identification of the specific user, computer system, and the decision task(s). The user is an ASD Program Manager. There are many System Program Offices at ASD, so this definition needs to be further refined. The prototyping methodology requires interaction with a specific person or group to specify the strengths and purpose of the DSS. Koble (30) researched the decision tasks that AFSC program managers feel should be included in a DSS. His research indicates that Program Managers think scheduling orientated processes should be among the first implemented in a DSS. Using this guidance, an ASD Program Manager interested in prototyping a scheduling system was identified. The ASD program office of RW commenced a study in early 1985 to define their program tasks and task relationships. They require an automated system to portray the information, allow for sensitivity ("what if") analysis, and, of great importance to them, to be a program managers assistant and training aid.

Program Managers have been distributed large numbers of

Z-100 computers. These systems are capable of supporting the decision making processes faced by the System Program Offices (SPOs). The decision support system is created for a generic MS-DOS computer. In those cases in which incompatabilities exist between the Z-100 and a generic MS-DOS system, the Z-100 solution will be implemented in the DSS.

Software Selection

Investigative objective 2 encompasses the functions of identifying and selecting specific software to be integrated into the decision support system. The most appropriate decision tools for program management problems will be further researched to determine the 'best' implementation of that decision tool. Since a given problem can be solved in many different ways, the specific software variant of the chosen decision tools will be selected from the vantage point of the Program Manager.

Software Gathering. The research for computer usable decision tools is limited to software available without charge. This includes software authored by other government agencies and public domain software.

Many AFIT theses have included coding various O/R techniques into computer usable form. These theses become an excellent source for quality software variants of different mathematical techniques. The Air Force has many study organizations which have adapted some O/R techniques for their particular uses. Their software is also available, without charge, to be included in the decision support system. Another source of computer source code is the Design Center located at Gunter AFB. They develop, catalog and distribute existing Air Force domain software. The Design Center is also a good source of other contacts.

The Air Force Zenith Z-100 procurement included some powerful software packages. These packages are proprietary since they are bought for specific systems. A word processor bought for system 'A' can not be legally executed on system 'B'. The Z-100 buy for ASD includes some of these packages. In FY84 ASD specifically purchased:

TABLE II

ASD Procured Software Tools (59:ATCH 1)

503	16-BIT OPERATING SYS Z-DOS
413	BASIC COMPILER 16-BIT
56	FORTRAN-86 16-BIT COMPILER
139	CONDOR 16-BIT DBMS
62	GRAFTALK 16-BIT GRAPHICS
64	LOTUS 1-2-3
35	dBASE II (16 BIT)

These packages can be applied to the program management world. The proprietary problem is one reason that prohibts the program manager from adopting one of these purchased packages. Program Managers are frequently traveling between the facilities of his program's interested factions. Ideally, he/she should be able to take the DSS with them to use. Since the procured tools can only be executed on their specific target system, it can not travel with them. Not enough of the powerful tools are available for all the program managers to have one. Systems that could assist the Program Manager are : Lotus 1-2-3, dBase II and Condor. These packages are spread throughout ASD. They are installed on machines that are unaccessable to the majority of program managers. Lastly, the system program offices are concerned about the lack of experience of new program managers. By making a procured system become the program managers prime DSS, these new inexperienced managers must learn it prior to becoming productive as program managers. This becomes just one more training task for the program office to conduct in order to train a new program manager. The procured software is excluded from the candidates of potential DSS integrated software.

Candidate Software. Koble's (30) research identified the specific techniques that program managers felt they needed in a DSS. The top techquies are : 1. GANTT/MILESTONE, 2. NETWORKING and 3. FINANCIAL METHODS (30:100). A GANTT program was acquired from the ICAM program office of the Materials Laboratory at WPAFB. This program was limited in its ability to represent data in different ways for the user, so a GANTT program was created during this research. There were six programs acquired that could assist with networking. A summary of the six appear in table III.

TABLE III

NETWORKING ANALYTIC TECHNIQUES

PERTCP	- Conducts a Critical Path Method or Project Evaluation and Review Technique analysis.
CANQ	- Assume a closed network. Performs a steady-state average network performance. Specific for manufacturing problems.
GERTE	- Conducts an analysis of a stochastic network.
NETFLO	- Optimizes the flow through a network.
NETSOL	- Conducts an analysis of a network of queues.
CSNAS	- Conducts a Critical Path Method analysis.

PERTCP and CSNAS are the only two of the above programs that are specifically suited for the critical path analysis that is required by the RW program office. PERTCP will conduct either a stochastic PERT or a deterministic CPM. CSNAS only conducts the CPM. The difference between these two is that PERT uses the node's most likely time, pessimistic duration and optimistic duration in its calculation of the network duration. Using this data one of the outputs from PERT is a duration probability estimate. The user is informed, for example that in 99% of occurences, the network will be complete in a certain number of days. This is the pessimistic estimate of the network duration. The Program Manager is given an estimate for the: 99%, 90%, 75%, 66%, 50% and 25% probability durations of the network.

Software Metrics. Multiple software tools have been identified which can accomplish the same analytical technique. A selection process to determine the 'best' software implementation of each technique is necessary. Software quality has been defined by Robert Poston as:

> The totality of features and characteristics of a software product that bears an ability to satisfy a given need (46:356).

James McCall defines a set of attributes in his discussion about software metrics which can be used to demonstrate the quality of a software package (35:133). Quality software will contain most of these attributes. These attributes are used as a rule to determine which software package is included in the DSS.

TABLE IV

Attributes of Quality Software (7:129)

Correctness	Extent to which a program satisfies its specifications and fulfills the user's objectives
Reliability	Extent to which a program can be expected to perform its intended function with required precision.
Efficiency	The amount of computing resources and code required by a program to perform a function.
Integrity	Extent to which access to software or data by unauthorized persons can be controlled.
Usability	Effort required to learn, operate, prepare input, and interpret output of a program.
Maintainability	Effort required to locate and fix an error in an operational program.
Testability	Effort required to test a program to insure it performs its intended function.
Portability	Effort required to transfer a program from one hardware configuration and/or software system environment to another.
Reusability	Extent to which a program can be used in other applications-related to the packaging and scope of the functions that programs perform.
Interoperability	Effort required to couple one system with another.

The selected software will need to be converted into Z-100 usable form. Most of the software is written in

FORTRAN. Since FORTRAN has been around for so long, there are numerous variants of this 'standard' language. Programs that need to be converted will be converted to FORTRAN 77 since it is the most transportable version of FORTRAN.

User Friendly Driver/Interface

Research objective 3 identifies the task of creating a user friendly driver. The selected programs need to be interlaced with each other. The strength of a Decision Support System is its ability to translate a users need into the execution of the appropriate program or set of appropriate programs. A data base will need to be constructed so the decision support system will 'know' the attributes of each of its programs. This data base needs to be accessible by the system. A data base management system (DBMS) needs to be employed to keep track of the user's data and the models' attributes.

The DBMS and DSS models need to be controlled by some top level driver. This driver will need to take advantage of Z-DOS system level routines. A computer language which can address the system will be used. The ZBASIC language is an example of this type of computer language. Since the target computer systems do not all have the ZBASIC interpreter, the ZBASIC source code is compiled and the machine usable code is distributed to the ASD program managers. This has the added advantage of alleviating the proprietary problem. The DSS can legally be executed on any Z-100.

Selected Software Implementation

Research objective 4 identifies the task of implementing the selected software. The PERTCP (1) program was chosen as the basis for the scheduling program. This routine needed to be modified to be placed within the DSS. In its acquired state (3): It was too large to be compiled on the Z-100; The program referenced nodes by number instead of by text labels; and it did not compute the network probability estimates. These modification were applied to the source code. PERTCP outputs a listing of activity start and finish times.

A graphic presentation of the PERTCP output would allow the program manager to absorb the data and its meaning faster. The GANTT program was written to present the PERTCP data in a more meaningful way. The activities are presented using psuedo-graphics. Psuedo-graphics uses the normal characters on the keyboard to simulate graphic symbols. GANTT displays activities that are on the network critical path as '*'. The event durations are displayed as '+' and any slack time is displayed as a '-'. The critical path of the network is the sequence of activities with no slack time. The result of delaying one of these nodes is that the entire network will be delayed. Slack time is shown after the event duration. The slack time of a node is the amount of time a node can be delayed without impact on the critical path of the network.

While viewing the network GANTT chart, the program manager is given the opportunity to view the details surrounding any particular activity. With this visibility function the program manager highlights the activity of interest, and views the activity worksheet. The worksheets contain information about the activity. It specifically contains: a description of the activity, the Office of Primary Responsibility (OPR), estimates of the activity duration, a listing of related regulations, and previous lessons learned about the activity. The program manager is allowed and encouraged to edit the worksheets as modifications to the program occur.

Decision Support System Evaluation

Obtaining feedback on the Decision Support System ability to assist program managers was identified as research objective 5. The prototype of the Decision Support System was demonstrated to the following three categories of users: actual program managers from the program office of RW, AFIT students in the program management curriculum of GSM, and the top management for the program offices of RW and TA. A literature review was conducted to identify an appropriate instrument to evaluate the DSS. King and Rodriguez (28) developed an instrument that evaluates the value perception of the DSS. This evaluation category can be accessed after only having a DSS demonstrated to a user. Value perception questions were selected from this instrument.

Pearson (8) developed a 39 question questionnaire to evalute decision support systems. His instrument exhibited very high reliability and validity. Olson (23) validated his questionnaire and shortened its overall length. The instrument as modified by Olson was still considered too long, and for most questions not appropriate for users that have not extensively used a DSS. Evaluation questions from her modified instrument were selected to be placed in the feedback instrument.

There were no evaluation instruments found in the literature to evaluate a DSS by those who had not extensiviely used the system. Questions concerning the apparent user friendliness of the DSS were derived and added to the selected literature review questions.

IV. SYSTEM DESIGN

Development Philosophy

The Decision Support System should be flexible, adaptable, modular and user friendly. These tenants were paramount in the development phase of the Program Manager's Decision Support System. The system is flexible and adaptable to future program manager requirements. The Decision Support System can be expanded without any change to the software source code. Flexibility is obtained by orienting the system to user files.

The user files are organized in a tiered arrangement, that is; a root analytic selection file points to the individual analytic technique edit and model selection files. Each of these, in turn points to its own help file. The user is offered as much assistance as he may require. An experienced user may need no assistance, whereas a novice can recieve extensive help.

The tiered arrangement of the help files is illustrative of the organization of the entire Decision Support System. The software is also arranged in this fashion. There are four different levels of software in the DSS. The Controller/User Interface is the top level. This is a compiled ZBASIC program. lt directly interacts with the user and his selected option. An interaction may require the system to access many types of files. At the systems disposal, hence also the user, are help files, menu files, edit field description files and the actual user input streams. The controller outputs a single line BATCH file to direct further system actions. A BATCH file, the second level of DSS software, is a sequence of operating system commands. The controller generated BATCH file is interpreter by the MS-DOS operating system, the third level of software, and executes another (user selected) BATCH file. The user selected BATCH file is passed a series of arguments to enable it to execute the proper FORTRAN or compiled ZBASIC program. These high level computer language programs compose the fourth level of DSS software. The programs are implementations of the selected analytic technique. They use a batch type of input stream to derive their results. A batch type of input is not the same as a BATCH file. Microsoft, the author of the MS-DOS operating system, adopted the term "BATCH", meaning a series of operating system commands from the traditional meaning of the word, that is a batch type of input stream. A batch input stream incorporates all of the data that a program will need into a file. There is no direct interface with the user. All of the FORTRAN programs execute using designated files for their input. The user data is copied onto this file to enable the FORTRAN routine to execute with the proper data.

Controller/User Interface

The user selects and edits his data by using the controller/user interface. By standardizing the method of accomplishing selection and editing, a user-friendly interface was designed. The user is only allowed to operate six keys in the selection mode. Any keystroke other than the six is considered an error, causing the highlighted menu item to blink. In the edit mode; operation selection, menu movement, and actual text editing is permitted. During both modes, help is only a keystroke away. While editing, the help messages inform the user of the meaning of the highlighted data field. In the selection mode, help messages inform the Program Manager the effect of executing the highlighted menu item.

The system's extensive help facility is designed to reduce training time and instruct the Program Manager about program management activities. While in the edit mode, the user is encouraged to view and update the activity worksheets. These are single screen descriptions of all the program manager's activities. Worksheets, if kept up-to-date, can be used to inform the program manager of the details behind schedule slippages.

When the user updates a worksheet or changes an input stream a backup file is created on the user disk. These are identified by the extension of ".BAK" as the last four characters in the data file name. Since the software linkages are through files, the backup file are a safety precaution.

File Oriented

The system is tied together with the extensive use of files. Approximately 200 files are required to activate all of the features of the DSS. These files are split between two disks: the PMDSS-SYS (system disk) and the PMDSS-USR (user disk). The analytical techniques, controller, menu files, help files and the input stream data field description files reside on the system disk. A Program Manager will not normally need to change this data. The user disk contains user specific data. Stored on the user disk are: the input streams indexes, the actual input streams, backup files and the program management activities worksheets. By orienting the system to files, a natural modularity and flexibility is obtained. The DSS uses ASCII files. An ASCII file can be viewed or modified with a standard editor, such as WORDSTAR.

Flexibility. The expandibility of the DSS has been briefly mentioned. Oulining the steps and files required to add a new analytic technique to the DSS will illustrate its file oriented flexibility. Table V lists the files that are required to be added to the system disk for the new application. The "ATSLCT.MNU" file is the system disk pointer to other analytic technique files. To add a technique, this ASCII file should be viewed and appended with a new file name and text description. For this example, the file name of "NEW" will be used. A help file, menu file, edit file with its associated help and the edit sub-options help files need to be added to the system disk. Respectively, the added files will be named: NEW.HLP, NEW.MNU, EDTNEWIN.MNU, EDTNEWIN.HLP. The edit sub-option help files explain the edit menu item. The six edit menu items are: modify an input stream, create a new file by modifying an existing input file, input a data stream from the keyboard, merge from an existing file, and delete a data file. Respectively, the sub-option help files are named: NEWMODRC.HLP, NEWMODCR.HLP, NEWINPUT.HLP, NEWMERGE.HLP and NEWDELET.HLP.

TABLE V

System Disk Files Required to Add an Application

ATSLCT.MNU	- The AT pointer file. Add a line for the "NEW" application.
NEW.MNU	- The NEW menu file. Points to the edit and input stream indexes and help files.
NEW.HLP	- NEW help file. Should describe the purpose of the NEW application.
EDTNEWIN.MNU	- NEW edit pointer file. Points to the edit help files. Used for menu selection
EDTNEWIN.HLP	- NEW edit help file. Should describe the edit alternatives.
NEWMODRC.HLP	- Describes the file change process.
NEWMODCR.HLP	- Describes the action of changing a file to create another.
NEWINPUT.HLP	-Describes the input from the keyboard alternative.
NEWMERGE.HLP	-Describes the merge option. With the merge option, a file is created from parts of two other input streams.
NEWDELET.HLP	- Describes the process of file deletion
N.FLD	- A description of the individual data fields used by "NEW". Describes the field start column, its width, its maximum and minimum values.
NFLD.HLP	- Contains a four line help message for each data field in N.FLD.
NEW.BAT	- The MS-DOS BATCH file to execute NEW
NEW.EXE	- The machine usable code for NEW

The input stream required by the program "NEW" has a specific format. That structure needs to be captured in the "N.FLD" file. For each field in the file, a four line help message can be generated with the ZBASIC program BLDHLP. The multiple field help messages are saved in the "NFLD.HLP" file. The above files are required to insure a user friendly interaction. The actual execution files for the program "NEW" are: a MS-DOS BATCH file and the compiled machine usable code. All these files reside on the system disk. The user disk requires an index file named "CHGNEWIN.MNU" to point to the specific input streams. Each of the input streams on the user disk start with the three letters NEW and ends with the extension of ".INP". The middle five characters are user selected. An example is the name of the file "NEWRELRW.INP". This is the REaL data for RW.

User Friendly Design

Several schemes are incorporated into the DSS to enable a friendly person-machine interaction. The controller is the primary user interaction program. The controller uses a hierarchy of screens to enable the user to select the execution of an analytical technique and/or edit the input stream. Many options are available, yet the user could execute a program with only four keystrokes or leave the DSS with a single keystroke. The screens have a simple, standard layout. The top portion displays the menu alternatives while the lower section reminds the user of the activated keys.

To ease workload reasonability, only a few options are allowed from any particular screen. An option can be selected or explained with the use of a limited number of activated keys. Six keys are activated during the selection process and twelve function keys are available for editing. A different, yet consistent screen layout is presented to the user in the edit mode to make it obvious to the user that the interaction mode has changed. The current data field is highlighted. The data can be modified while in the edit mode. For numeric fields, the new data is tested against the field's upper and lower limit. Unreasonable data is rejected and the field is reset to the field's minimum.

Extensive use of highlighting is used by the controller and visibility programs. Highlighting directs the users attention to items of importance. In cases where the user may become confused about his options or the current menu item, highlighting is used. When a keystroke is expected of the user, a highlighted message informs him of the alternatives. Each screen has a title. This is highlighted to differentiate it from the menu options. The title of subsequent screens also has a one-to-one correspondence with the selection menu item.

The last user friendly scheme implemented in the system is the amount of assistance offered to the user. Help is only a keystroke away. By depressing the "HELP" key on the Z-100 keyboard, explanatory messages are displayed.

V. Results and Recommendations

Results

Two results have emerged from the research. The obvious one is that a decision support system prototype has been tailored to the needs of the program managers at the System Program Office (SPO) of RW. Using an iterative process the system was developed to satisfy the prime prototype user. An evaluation of the prototype system by its prospective users is the second result of the research. The ease of system use and the goal of using the system as a SPO training aid were stressed to the developer. An eleven question evaluation questionnaire was created to examine how well the prototype satisfied these and other DSS requirements.

The Decision Support System was demonstrated to two groups of prospective users. The system was developed using a program manager at the system program office of RW as the prototype user. Her reactions to the various versions of the system were captured and used to modify subsequent versions. The first demonstration of the system was to her peers in RW. Ten RW practitioners were assembled for the system demonstration. This group had a wide demographic composition. It consisted of civilian program managers with years of experience to new Air Force second lieutenants. They were shown the method used to enter the system, create a network, modify the created network, and the meaning of the system outputs (both help messages and analytic techniques). The second group that the system was demonstrated to was the 13 members of the Systems Management curriculum (GSM) for AFIT class 86S. Their curriculum is intended as a graduate education for program managers. Whereas the people from the RW SPO are current practitioners, this group consists of past and future practitioners of program management. This group's demographic make-up was more similar than the RW program managers. It consisted of senior lieutenants, junior captains and two foreign military officers. With both demonstrations, a volunteer from the group was selected to execute the system. They were given minimum guidance and directed to the system's help facility to resolve questions. The observations from both groups of generic program managers were captured and quantified in a developed questionnaire. The sample size is very small, thus statistically significant conclusions can not be drawn. The following analysis demonstrates the system user satisfaction trends.

Questionnaire. The eleven questions in the designed evaluation instrument are intended to measure: an observer's perception of the system value, their propensity to use the system, and the evaluation of the quality of the system results. The value perception questions originated with the research of King and Rodriguez (28). The system quality questions were derived from the Information system User Satisfaction work of Ives and Olson (23) and Pearson (8). The remaining questions reguarding the users propensity to interact with the DSS were self generated. The established evaluation instruments (28, 8, 23) demonstrated high validity and reliability. They were designed to evaluate systems by experienced users. The 23 generic program managers who evaluated the PMDSS only had the opportunity to view a demonstration of the DSS. Many of the established questionnaire items are inappropriate for this group.

The established instruments' high reliability and validity can not be claimed by the created questionnaire. All the questions in the established instruments have undergone an intensive screening process. This process has improved the quality of the questions selected from these instruments, hence it has also contributed to the validity and reliability of the created questionnaire.

The semantic differential method of evaluation was coupled with a seven point Likert scale to quantify the respondees evaluation. The seven values are obtained by pairing a descriptive adverb with one of the opposing extremes of an adjective continuum. The adverbs are: Extremely, Quite, Slightly, or Neither adjective applies. Descriptive statistics are applied to the results.

The mean is the group's average reply. The standard deviation is a measure of the average difference from the

the responses spans more than four adjacent question values. The minimum standard deviation of the "low" agreement category is 1.5. The label of "moderate" is given to the level of agreement between the "high" and "low" ranges. Figure 1 pictorically demonstrates these ranges of agreement for a question with a mean of "4". If all the responses fell within the indicated "high" or "moderate" brackets then the standard deviation would also fall within the ranges for these categories. These ranges are: 0 to 1, 1 to 1.5 and above 1.5.

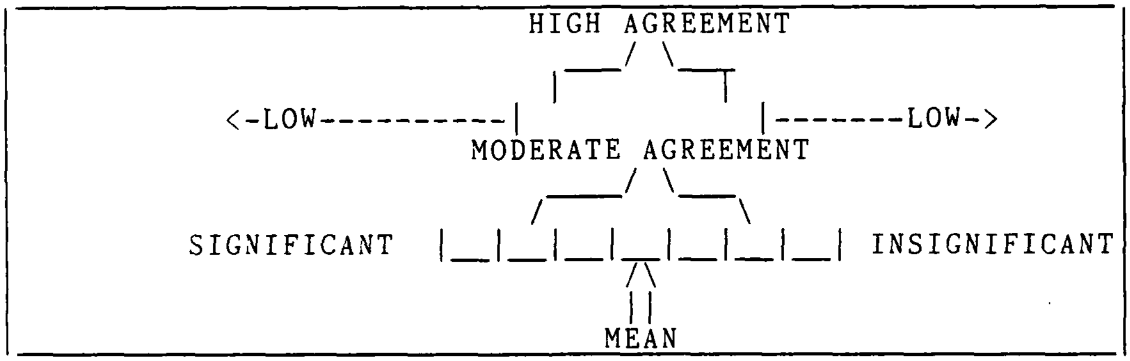


Fig 1. Standard Deviation Ranges

The questions, descriptive statistics and a histogram showing the actual replies are shown. The histogram symbols of "S" and "*" are used. They respectively represent GSM students and RW program managers.

<u>Value Perception</u>. The first four questions of the evaluation instrument measure the respondents value perception of the DSS. They indicate the respondees estimation of the worth of the system.

```
IV.  Probability that you would use the system.

    SAMPLE MEAN= 2.57, STANDARD DEVIATION = 1.4
RW Program Managers MEAN = 3.0, STANDARD DEVIATION= 1.8
 AFIT GSM Students MEAN = 2.2, STANDARD DEVIATION=  .93

             S
       SS SS SS
       S* SS *S
       ** S* ** S  ** *
HIGH  |_6|_6|_7|_1|_2|_1|__| LOW
```

Fig 2. Probability of System Use

Figure 2 displays the results of the first question. The mean value is 2.57. This can be interpreted as the group has a "QUITE HIGH" to "SLIGHTLY HIGH" probability of using the system. The standard deviation is 1.4. It is close to the "low" confidence threshold. This means that the group was not in agreement on the specific amount of use of the DSS. Their opinions were spread about the mean. All except three indicated a variant of "high" probable use, so the spread is due to the degree of high probable use that each desires with the system. The three extreme evaluators are from the RW SPO. The sub-group from RW has very "low" agreement about their group's "slightly high" probability of using the DSS. This is contrasted with the "high" agreement that the GSM students have concerning their "quite high" probable use of the system.

The second value perception question appears in figure 3.

```
 2V.  Probability that other managers will use the system.

     SAMPLE MEAN= 2.22, STANDARD DEVIATION = 1.1
 RW Program Managers MEAN = 2.5, STANDARD DEVIATION= 1.5
  AFIT GSM Students MEAN = 2.0, STANDARD DEVIATION=  .71

                          S
                          SS
                          SS
                          SS
                       S  SS
                       S* S*
                       ** ** ** S  **
                HIGH  |_5|13|_2|_1|_2|__|__| LOW
```

Fig 3. Probable System Use by Others

They believe there is a "QUITE HIGH" probability that others will use the DSS. The standard deviation falls into the moderate range. The respondees are somewhat in agreement. The sub-groups difference is similar to that of the previous question. The RW program managers have a "high" level of disagreement about their sub-group average evaluation of there being a "QUITE HIGH" probable use of the system by others. The GSM students have an extremely "high" level of agreement between themselves about there being a "QUITE HIGH" probable use of the system by others.

The third value perception question appears in figure 4.

```
3V.  Probability that the system will be a success.

  SAMPLE MEAN= 2.35, STANDARD DEVIATION = .88
RW Program Managers MEAN = 2.6, STANDARD DEVIATION= .97
 AFIT GSM Students MEAN = 2.2, STANDARD DEVIATION=  .80

                         SS
                         SS
                         SS
                         SS  S
                      S  **  *S  *
                      S* **  **  *S
             HIGH   |_3|12|_5|_3|__|__|__| LOW
```

Fig 4. Probable System Success

The respondees believe the system will be a "QUITE HIGH" success. The standard deviation indicates that there is high agreement between the group on the degree of expected system success. Both subgroups support this opi..ion, although the GSM students rating of "QUITE GOOD" is higher than that of the RW program managers average rating of "SLIGHLY GOOD". They are also in higher agreement as a subgroup than the RW SPO.

The fourth question is displayed in figure 5.

```
4V.  Managers evaluation of the worth of the system.

  SAMPLE MEAN= 2.35, STANDARD DEVIATION =1.15
RW Program Managers MEAN = 2.8, STANDARD DEVIATION= 1.3
 AFIT GSM Students MEAN = 2.0, STANDARD DEVIATION=  .91

                         SS
                      SS SS  S
                      SS SS  S*  S
                      ** **  **  **  *
             GOOD   |_6|_8|_5|_3|_1|__|__| BAD
```

Fig 5. System Worth

The last value perception question directly addresses the respondees evaluation of the worth of the system. Their evaluation of the system's worth is that it is "QUITE GOOD". The respondees are in moderate agreement. Since only one respondee evaluated the system as a degree of BAD, the group's disagreement concerns the degree of the system goodness.The prospective users from the subgroups varied on their evaluation of the system worth. The program managers from RW have a larger range of responses than the GSM students. The GSM students think the system is worth more than the RW SPO program managers, although both rate it as a degree of "high" worth.

The respondees answers to the four value perception questions indicate that they believe the system is "QUITE" valuable. There is a moderate amount of agreement between the respondees. Of the 92 individual replies, only six indicate a low percieved value of the system. These are interpreted as outliers. The amount of disagreement has more to do with the degree of goodness than whether the system is good or bad. The RW program managers responses vary widely on each question.

System Quality. There are three questions derived from Pearsons (8) instrument to indicate the users satisfaction with the DSS. The qualities of expected training, relevancy and the users confidence in the DSS are measures of the system quality.

```
6V. The degree of training you would need before you
could use the system.

  SAMPLE MEAN= 4.0, STANDARD DEVIATION =1.6
RW Program Managers MEAN = 3.9, STANDARD DEVIATION= 1.9
 AFIT GSM Students MEAN = 4.1, STANDARD DEVIATION=  1.4

                                 S
                                 SS
                           S     SS
                        SS S*    SS
                     *  S* ** *  ** *S *
             LITTLE |_1|_4|_5|_1|_9|_2|_1| MUCH
```

Fig 6. Expected Required Training

The sixth question in the evaluation instrument is the first system quality measure. The question and the respondents results are shown in figure 6. The respondees average reply was "NEITHER LITTLE nor MUCH" training will be required by potential users. There is high disagreement between the respondees on the needed level of training. The responses range from "EXTREMELY LITTLE" training is required to "EXTREMELY MUCH". The above histogram shows that there are two groups with diametrically opposed opinions concerning the amount of required training. It appears as if the GSM students are more polarized. The RW program managers spread their responses from both extremes of required training. They have very "low" agreement within their group.

The second system quality question concerns the confidence that the respondees have in the DSS results. It evaluates their confidence in the PMDSS and how important they feel it is to have confidence in any DSS. Figure 7 displays the respondents results. The respondees indicate they have "SLIGHTLY HIGH" confidence in the PMDSS. The low standard deviation shows there is high agreement between the group. The single outlier accounts for the majority of the standard deviation. The evaluators also highly agree that it is "QUITE IMPORTANT" to have confidence in a DSS.

```
  8E. Confidence in the system.  The extent of your
 assurance or confidence in the system results.

    SAMPLE MEAN= 2.7 , STANDARD DEVIATION = .93
 RW Program Managers MEAN = 3.0, STANDARD DEVIATION= 1.25
  AFIT GSM Students MEAN = 2.4, STANDARD DEVIATION=  .52

                        S
                        SS SS
                        SS SS
                        SS SS
                        ** **
                        ** ** *    *
               HIGH |__|11|10|_1|__|_1|__| LOW

    SAMPLE MEAN= 2.0 , STANDARD DEVIATION = .90
 RW Program Managers MEAN = 2.2, STANDARD DEVIATION= 1.0
  AFIT GSM Students MEAN =1.85, STANDARD DEVIATION=  .80

                     SS SS
                     SS SS SS
                     S* S* S*
                     ** ** ** *
  item is IMPORTANT |_8|_8|_6|_1|__|__|__| UNIMPORTANT
```

Fig 7. Confidence in DSS

The single outlier is an RW program manager. The variability of the RW SPO sub-group jumps from being very little (high agreement of .707 std Dev) without this person to only a moderate amount of agreement with him included.

The students exhibit very high level of agreement that they are "QUITE HIGH"ly coniident in the system.

The last system quality question evaluates whether the system provides the assistance that the program manager thinks is needed. The demonstrated system relevance and the importance of this qualitv for any DSS is evaluated. The question and the groups response are shown in figure 8.

```
9E. Relevancy. The degree of congruence between what
you want from the system and what is provided.

   SAMPLE MEAN= 2.26, STANDARD DEVIATION =1.0
RW Program Managers MEAN = 2.3, STANDARD DEVIATION= 1.3
 AFIT GSM Students MEAN = 2.2, STANDARD DEVIATION=  .83

                  SS
                  SS
               S  SS SS
               S* S* S*
               ** ** ** S  *
        USEFUL |_5|10|_6,_1|_1|__|__| USELESS

    SAMPLE MEAN= 1.95, STANDARD DEVIATION = .98
RW Program Managers MEAN = 2.1, STANDARD DEVIATION= 1.1
 AFIT GSM Students MEAN =1.85, STANDARD DEVIATION=  .90

               S
               SS SS
               SS SS
               ** SS S*
               ** ** ** *S
 item is IMPORTANT |_9|_8|_4|_2|__|__|__| UNIMPORTANT
```

Fig 8. System Relevence

The PMDSS is evaluated as "QUITE USEFUL". The group views the PMDSS as answering a program management need. The agreement on the usefulness of the PMDSS is high. The group also highly agrees that it is "QUITE IMPORTANT" for any DSS to be relevant. The averages of the sub-groups is similar, but the variability differs greatly. The GSM students exhibit a "high" level of agreement, whereas the RW program managers border on a "low" level.

A DSS should be relevant to the users domain and the user should be confident in the system. The PMDSS is evaluated as being quite relevant and the evaluators have quite high confidence in it. The evaluators are confused as a group about the amount of required training it will take to use the PMDSS. The majority of the sample's variability is due to the RW program managers.

User Interaction Propensity. The remaining four questions in the evaluation instrument relate to the propensity of the user to interact with the PMDSS. These have to do with the effect using that system will have on the Program Manager's job. It is assumed that a Program Manager will want to use tools that assist him. The first interaction propensity question is displayed in figure 9.

```
5V.  The level of timesaving you expect from the system.

         SAMPLE MEAN= 2.3, STANDARD DEVIATION =1.15
  RW Program Managers MEAN = 2.6, STANDARD DEVIATION= 1.6
   AFIT GSM Students MEAN = 2.1, STANDARD DEVIATION=  .64

                 S
                 SS
                 SS
              S  SS
              S* S* SS
              ** ** S* *  **
  SIGNIFICANT |_5|11|_4|_1|_2|__|__| INSIGNIFICANT
```

Fig 9. Expected Time Savings'

Using the PMDSS is expected to make a "QUITE SIGNIFICANT" time savings impact on the program managers job. There is a moderate amount of agreement. The two outliers account for a large portion of the variance. There would be high agreement if these two were excluded. The two outliers are program managers from RW. They radically affect the amount of agreement that the sub-group of RW shares on their average evaluation of the system being "QUITE" to "SIGHTLY" "SIGNIFICANT" as a timesaving tool. Ther GSM students have high agreement as a sub-group that the system is a "QUITE" "SIGNIFICANT" as a time saver. Besides being a time saving tool, the PMDSS needs to support the program manager. Question 7 in figure 10 demonstrates whether the PMDSS will provide the needed support.

```
  7V The extent the system will support your sche-
  duling decisions.

        SAMPLE MEAN= 2.48, STANDARD DEVIATION =1.08
 RW Program Managers MEAN = 2.8, STANDARD DEVIATION= 1.5
  AFIT GSM Students MEAN = 2.2, STANDARD DEVIATION=  .60

                  S
                  SS
                  SS
                  SS SS
               S  S* SS
               ** ** ** *  **
        HIGH  |_3|11|_6|_1|_2|__|__| LOW
```

Fig 10. Expected Scheduling Support

The group expects the system to offer "QUITE HIGH" support to their scheduling decisions. This is important since scheduling is the most visible of the four aspects of the program managers job. There is moderate agreement between the evaluators that the system will offer 'QUITE HIGH" scheduling support. The disagreement of the sample is attributed to the lack of aggreement of the program managers from RW. Again they evaluated the quality with both the highest and lowest values. The effect of the PMDSS on all aspects of the program manager's job is directly accessed with the question in figure 11. The importance of a DSS having a significant effect on the program manager's job is shown in figure 12.

```
 10E. Job Effects. The changes in job freedom and job
 performance that are ascertained by you by using the
 PMDSS.
      SAMPLE MEAN= 2.83,  STANDARD DEVIATION =1.19
RW Program Managers MEAN = 2.8, STANDARD DEVIATION= 1.6
 AFIT GSM Students MEAN = 2.8, STANDARD DEVIATION=  .92

                       SS SS
                       SS SS
                       SS *S S
                    ** ** ** ** S  *
       SIGNIFICANT |_2|_8|_8|_3|_1|_1|__| INSIGNIFICANT
```

Fig 11. Job Effect due to System Use

```
  SAMPLE MEAN= 2.48,  STANDARD DEVIATION =1.16
RW Program Managers MEAN = 2.7, STANDARD DEVIATION= 1.5
 AFIT GSM Students MEAN = 2.3, STANDARD DEVIATION=  .85


                       S
                       SS S
                       SS SS
                    SS S* *S
                    ** ** ** S*    *
 item is IMPORTANT |_4|_9|_7|_2|__|_1|__| UNIMPORTANT
```

Fig 12. Importance of Job Effect

Respondents believe the PMDSS will have a "SLIGHTLY SIGNIFICANT" effect on their job. They are in moderate to low agreement about the changes in job freedom resulting from using the PMDSS. Since only two responses fall into the "INSIGNIFICANT" effect category, the disagreement seems to be about the degree of significant job effect caused by using the PMDSS. The evaluators feel it is "QUITE IMPORTANT" for any DSS to offer a significant effect on the program management job.

The last question is related to the users propensity to use the PMDSS. By designing an easy to use system, the propensity of the program manager to use it should increase. The user friendly question has two parts. Figures 13 and 14 contain the two portions of this question. The second is redundant since only two respondents varied in their evaluation on the second part.

```
IIE.  User Friendly. The DSS allows novice users to
operate effectively .

 SAMPLE MEAN= 2.26, STANDARD DEVIATION = .92
RW Program Managers MEAN = 2.3, STANDARD DEVIATION= .82
 AFIT GSM Students MEAN = 2.2, STANDARD DEVIATION=  1.0

                         SS
                         SS
                         SS
                         **
                      SS ** SS S
                      S* ** ** *S
                 EASE |_4|12|_4|_3|__|__|__| DIFFICULT
```

Fig 13. PMDSS User Friendliness

The PMDSS is viewed as "QUITE EASY" to use. They are in high agreement about the systems user friendly quality. There is a quite high propensity to use the system by the respondents. It is percieved as a useful program management tool that is easy for individuals to execute.

```
      SAMPLE MEAN= 2.3 , STANDARD DEVIATION = .93
 RW Program Managers MEAN = 2.5, STANDARD DEVIATION= .71
  AFIT GSM Students MEAN =2.15, STANDARD DEVIATION=  1.1

                        S
                        SS
                        SS
                        ** S
                     SS ** S* S
                     SS ** ** *S
      SIMPLE TO USE |_4|11|_5|_3|__|__|__| HARD TO USE
```

Fig 14. PMDSS User Friendliness

The 23 respondents generally had a very favorable impression of the PMDSS. The program managers from the SPO of RW had excessive disagreement within their sub-group. The demographic variation of this group may account for the consistently "low" level of agreement which they shared. They were responsible for the most dis-satisfied response for every question. The DSS is designed to be a tutorial tool for young, inexperienced military program managers. The students from AFIT and a majority of the sample from RW fit this description. The disagreement by the RW program managers can be traced to two individuals. The GSM students may have also been overly generous in their evaluation of the DSS. They are undergoing an intensive program management orientated curriculum designed to expose the student to useful analytical tools. Their propensity to embrace such tools may be higher than the "average" program manager in Air Force Systems Command.

Recommendations

There are three categories of recommendations to be made. Recomendations concern future evaluations of the current prototype, enhancements to be made to the present system and follow-on efforts. The evaluators only had an opportunity to view a demonstration of the system. Another evaluation should be conducted with a larger sample of experienced users. These users will have had a chance to "ring-out" the system and find any problem areas. They will be in a better position to realistically evaluate the system.

One of the results of Koble's (30) research is a prioritized listing of analytic techniques desired by program managers at ASD. Scheduling tools are present in the PMDSS. Other tools can be added to the PMDSS by creating and modifying ASCII text files. The software is capable of the addition, but the systems disk is completely filled.

There are a variety of approaches that can be used to enable the PMDSS system disk to use more techniques. These will be listed from the easiest to the most dramatic change.

The system disk contains help files, BATCH files and execution files. Currently the system is dependent on using only two disks. If a larger configuration can be obtained, only the reference to the location of these files will need to be changed in the current system. Since the beauty of the PMDSS is its ability to execute on the standard (small) 192k Z-100, this option is not very attractive.

The current system has help files for the six edit sub-options for each of the six analytic techniques. The 36 files can be pared down to six with minor adjustment to the BLDMNU ZBASIC routine. The freed space may be sufficient to allow another technique to be placed on the disk. This may be a viable short-term alternative, but sooner or later another technique will be desired and the search for more free disk space will resume.

The compiled ZBASIC programs are very large in comparison to simi'ar programs written in a more structured computer languages. These programs (BLDMNU, EVENT) could be rewritten into a more efficient language (PASCAL,"C"). The space saving may again be enough to allow another analytic technique to be placed on the disk. As with the above solution, this solution is short term at best.

The recommended solution is to partition the functions of the BLDMNU program into two programs. This routine currently conducts analytic technique selection, input model selection and model editing. All the analytic techniques' help files, BATCH files and execution routines must reside on the system disk due to the combination of functions. Most PMDSS interactions will concentrate on a specific analytic technique. The extra overhead of having all the files present is not needed and can be eliminated by transitioning to a multiple disk system.

The multiple disk variant of the PMDSS will have a root analytic selection routine on the root disk. This disk will contain the help files with broad explanations of each of the techniques and the technique options. The user would enter the system using this disk, describe the interaction goal and the system would instruct the user which subsequent disks to use. The next disk would contain specific help messages for the analytic technique, and the execution files. The interaction would then occur using this specific analytic technique exection disk and the data disk much like the current PMDSS. The system's expansion capability becomes virtually unlimited. The ZBASIC program BLDMNU can be easily modified to accommodate this transition. This effort would be a good canidate for a follow-on thesis.

Appendix A: Sample Terminal Session

The Program Manager's Decision Support System (PMDSS) assists a Program Manager with scheduling related problems. The following 47 screens demonstrate the process, screens and output that are presented to the Program Manager. Inputs, alternatives and the system's outputs will be briefly explained.

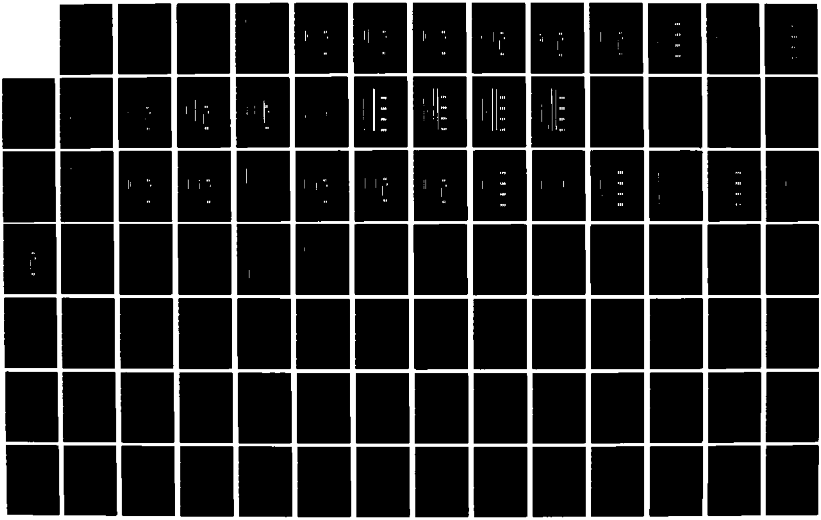
The required system configuration includes: a standard Zennith Z-100 computer with 192K RAM and two floppy disk drives. The user may recieve printed copies of the Z-100 display by depressing the SHIFT and the F12 keys simultaneously. The assumption is made that the computer is connected to the printer in the standard MS-DOS fashion. The parallel printer should be plugged into the Z-100 J3 port behind the computer.

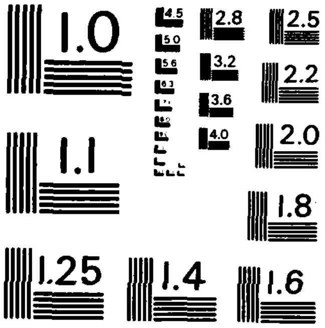
The system will self "boot" itself upon system power-up. To change the MS-DOS program load parameters the MS-DOS CONFIGUR program can be executed. The printer parameters, disk access speed and system disk drive are some of the items which can be modified. In the following sample session the only preparatory action not shown is the system power-up.

AD-A161 690 DECISION SUPPORT SYSTEM FOR ASD (AERONATICAL SYSTEMS DIVISION) PROGRAM MANAGERS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF SYST..

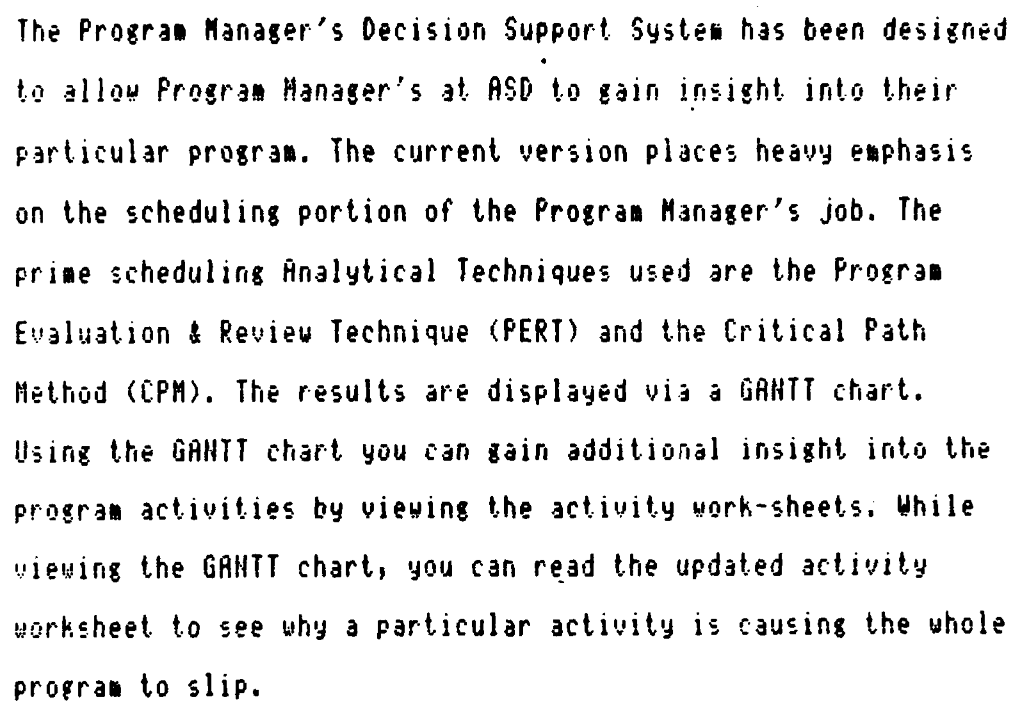
2/3

UNCLASSIFIED T W BROTHERTON SEP 85 AFIT-GSM/LSY/85S-5 F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



The Program Manager's Decision Support System has been designed to allow Program Manager's at ASD to gain insight into their particular program. The current version places heavy emphasis on the scheduling portion of the Program Manager's job. The prime scheduling Analytical Techniques used are the Program Evaluation & Review Technique (PERT) and the Critical Path Method (CPM). The results are displayed via a GANTT chart. Using the GANTT chart you can gain additional insight into the program activities by viewing the activity work-sheets. While viewing the GANTT chart, you can read the updated activity worksheet to see why a particular activity is causing the whole program to slip.

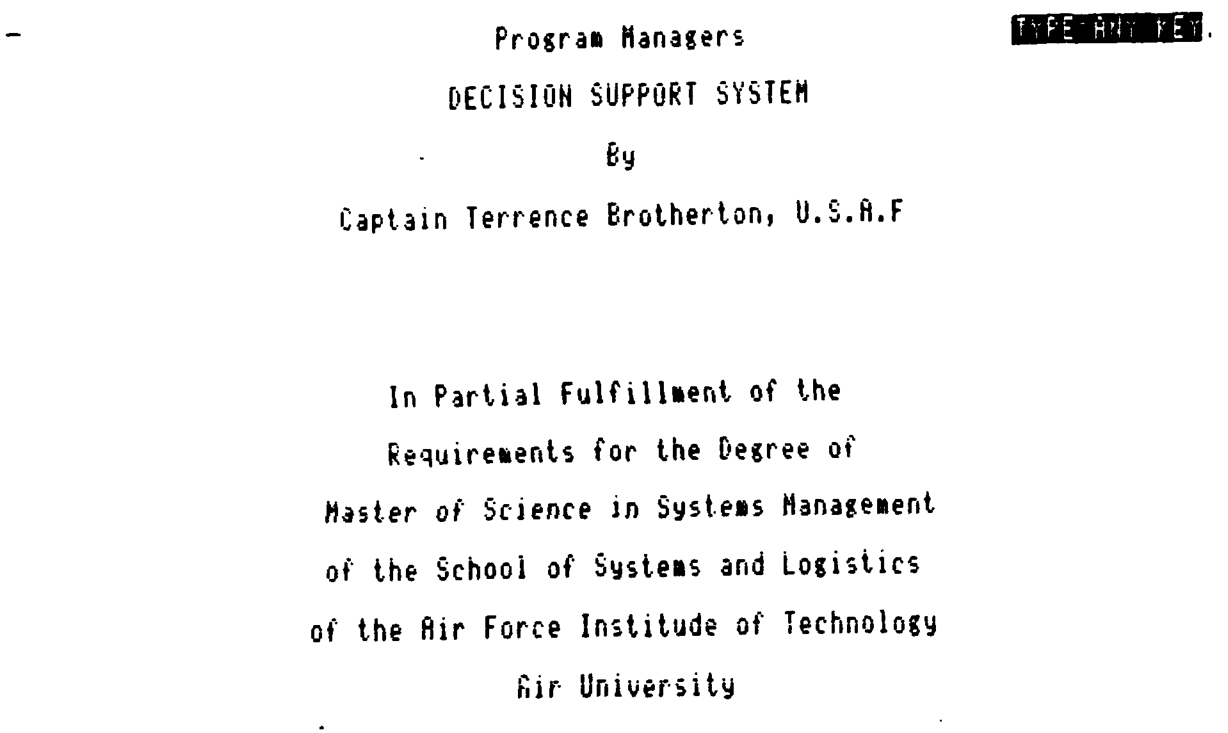
Fig 15. PMDSS Introduction Screen

The opening screen introduces the system to the program manager. The user could exit at this point by replying "y" to the question "DO YOU WANT TO EXIT THE DSS?" . By entering any other key the next slide is presented.

THE DSS MODEL AND OPTION ROUTINE IS BEING LOADED!

Fig 16. Option Routine Load Screen

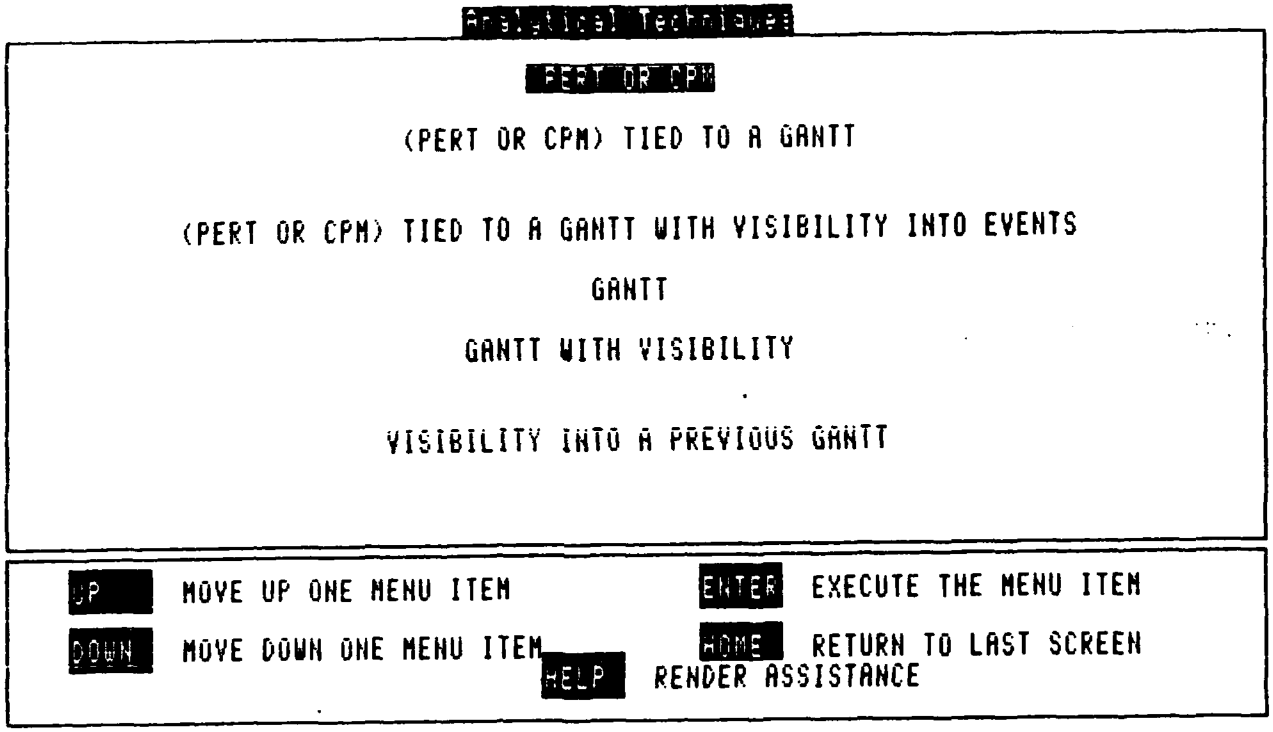
The DSS is composed of many different program. The above message is displayed while the option routine is being read for the disk.



```
-                      Program Managers                TYPE ANY KEY.
                   DECISION SUPPORT SYSTEM
                             By
            Captain Terrence Brotherton, U.S.A.F


               In Partial Fulfillment of the
               Requirements for the Degree of
          Master of Science in Systems Management
           of the School of Systems and Logistics
          of the Air Force Institude of Technology
                       Air University
```

Fig 17. Controller Introduction Screen

The DSS Controller/User Interface has been loaded and is ready to conduct the terminal session. The user depresses any key to continue.



Analytical Techniques

PERT OR CPM

(PERT OR CPM) TIED TO A GANTT

(PERT OR CPM) TIED TO A GANTT WITH VISIBILITY INTO EVENTS

GANTT

GANTT WITH VISIBILITY

VISIBILITY INTO A PREVIOUS GANTT

UP MOVE UP ONE MENU ITEM

DOWN MOVE DOWN ONE MENU ITEM

HELP RENDER ASSISTANCE

ENTER EXECUTE THE MENU ITEM

HOME RETURN TO LAST SCREEN

Fig 18. Initial Analytical Technique Screen

The six Analytical Techniques are presented. The user activated keys appear in the lower portion of the screen. The "UP" and "DOWN" refers to the arrow keys.

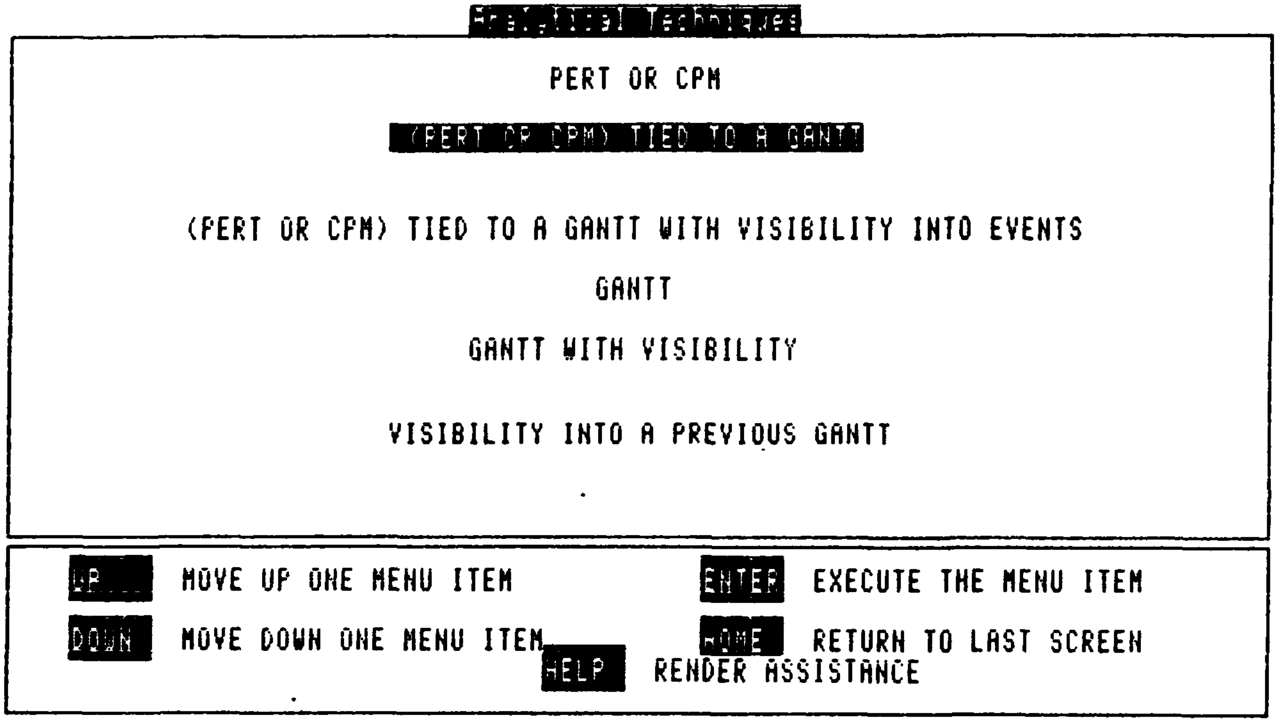
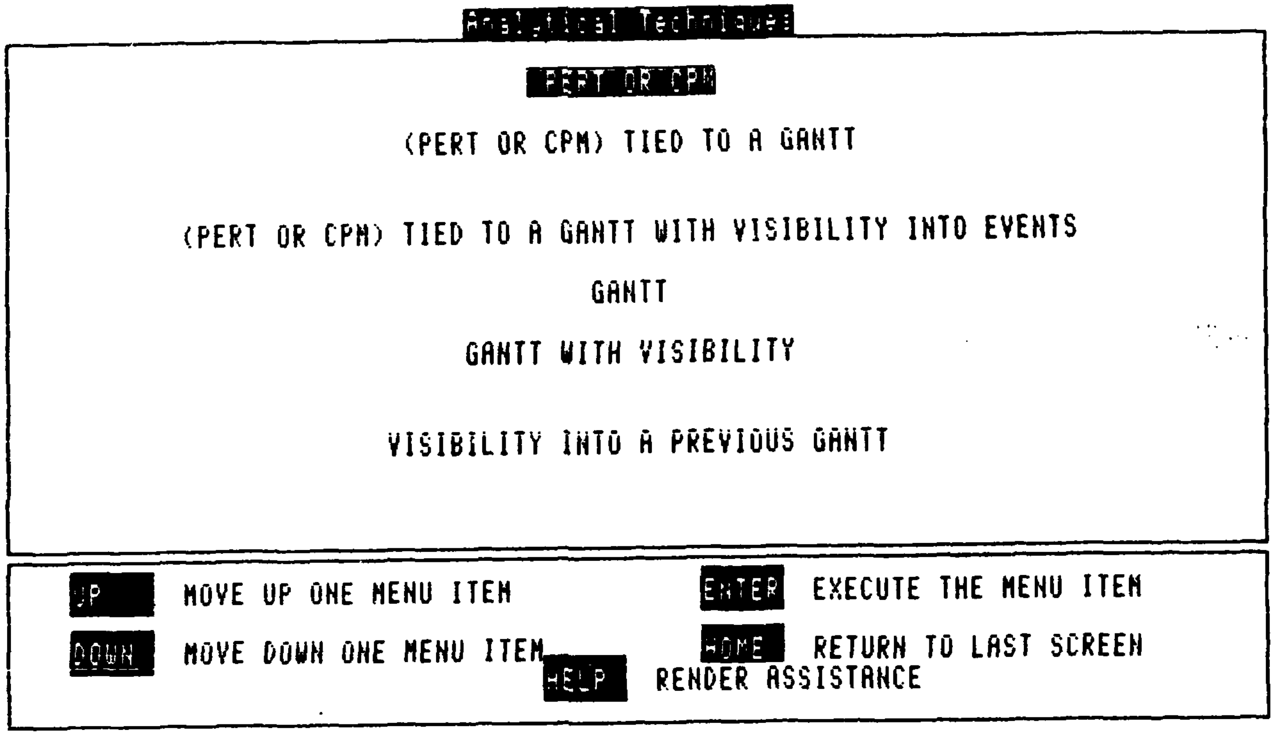


Fig 19. Second Analytical Technique Screen

Depressing the down arrow moves the highlighted menu item down one.



Analytical Techniques

PERT OR CPM

(PERT OR CPM) TIED TO A GANTT

(PERT OR CPM) TIED TO A GANTT WITH VISIBILITY INTO EVENTS

GANTT

GANTT WITH VISIBILITY

VISIBILITY INTO A PREVIOUS GANTT

UP MOVE UP ONE MENU ITEM

DOWN MOVE DOWN ONE MENU ITEM

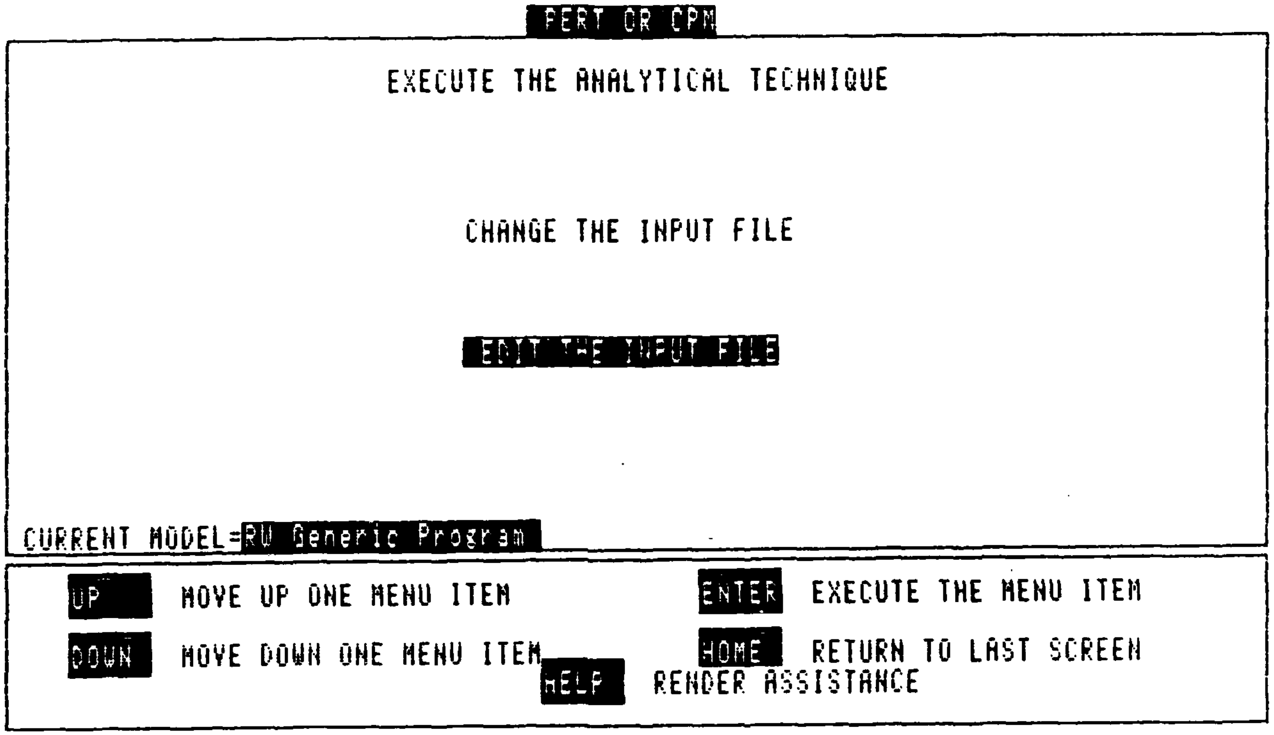
ENTER EXECUTE THE MENU ITEM

HOME RETURN TO LAST SCREEN

HELP RENDER ASSISTANCE

Fig 20. Third Analytical Technique Screen

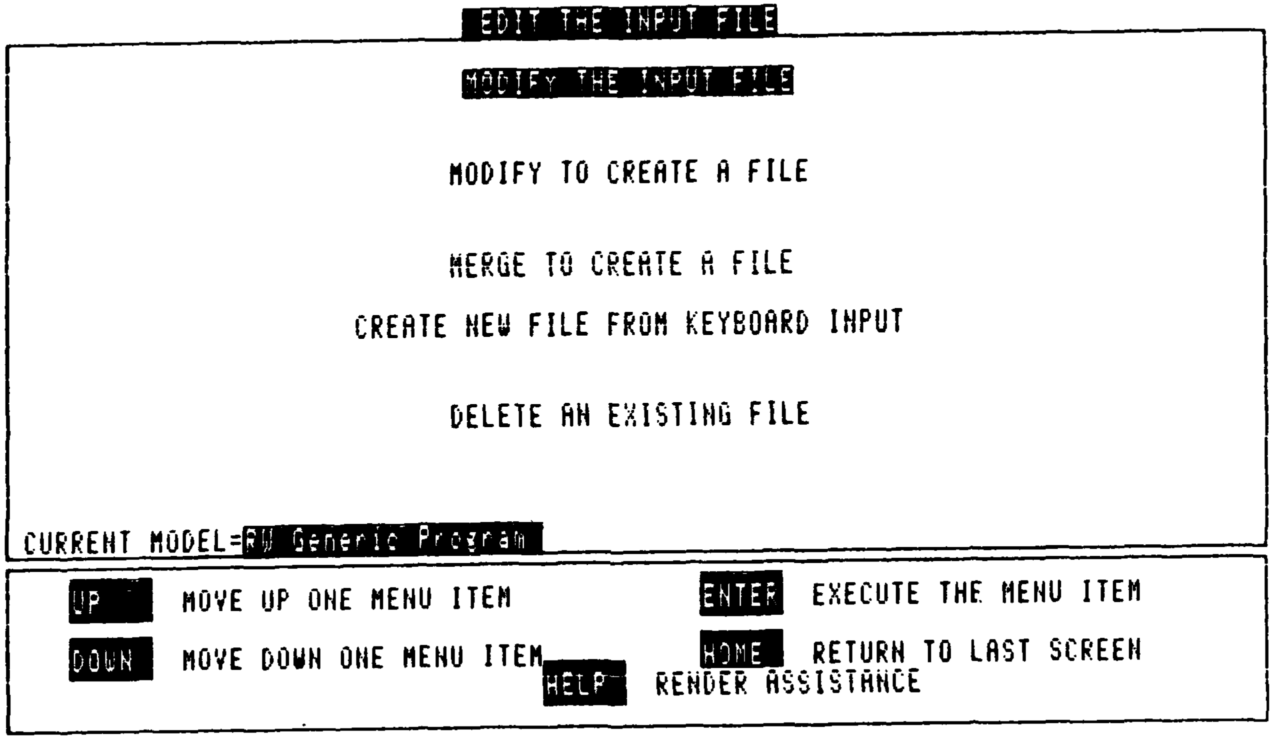
The up arrow returns the highlighter to the top menu item. Depressing the ENTER key indicates that the "PERT or CPM" analytic technique is desired.



CURRENT MODEL=RW Generic Program

Fig 21. Initial "PERT or CPM" screen

The current input model is highlighted in the lower left corner. It is the "RW Generic Program" by default. An input model is the input stream used during the AT execution. The cursor was moved to EDIT and the current model is editted.



EDIT THE INPUT FILE

MODIFY THE INPUT FILE

MODIFY TO CREATE A FILE

MERGE TO CREATE A FILE

CREATE NEW FILE FROM KEYBOARD INPUT

DELETE AN EXISTING FILE

CURRENT MODEL=RW Generic Program

UP MOVE UP ONE MENU ITEM

ENTER EXECUTE THE MENU ITEM

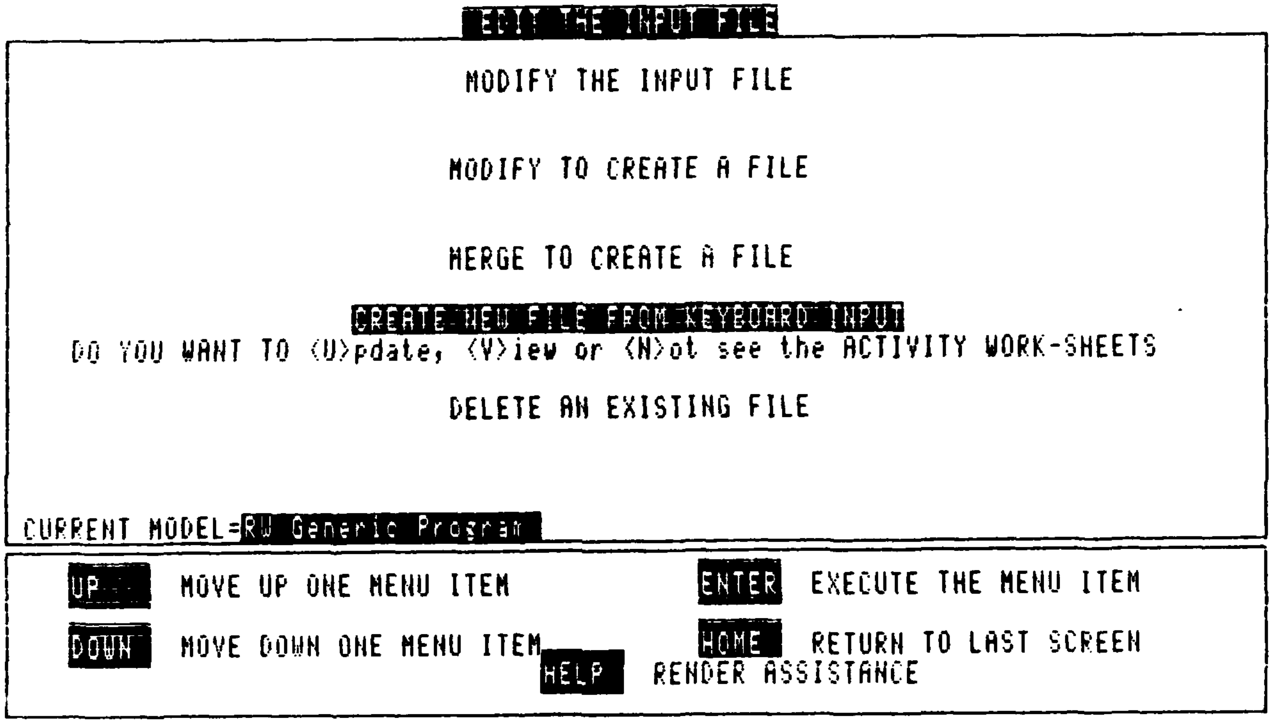
DOWN MOVE DOWN ONE MENU ITEM

HOME RETURN TO LAST SCREEN

HELP RENDER ASSISTANCE

Fig 22. Initial Edit Option Screen

The above five alternatives are available edit options.



EDIT THE INPUT FILE

MODIFY THE INPUT FILE

MODIFY TO CREATE A FILE

MERGE TO CREATE A FILE

CREATE NEW FILE FROM KEYBOARD INPUT

DO YOU WANT TO <U>pdate, <V>iew or <N>ot see the ACTIVITY WORK-SHEETS

DELETE AN EXISTING FILE

CURRENT MODEL=RW Generic Program

UP MOVE UP ONE MENU ITEM

ENTER EXECUTE THE MENU ITEM

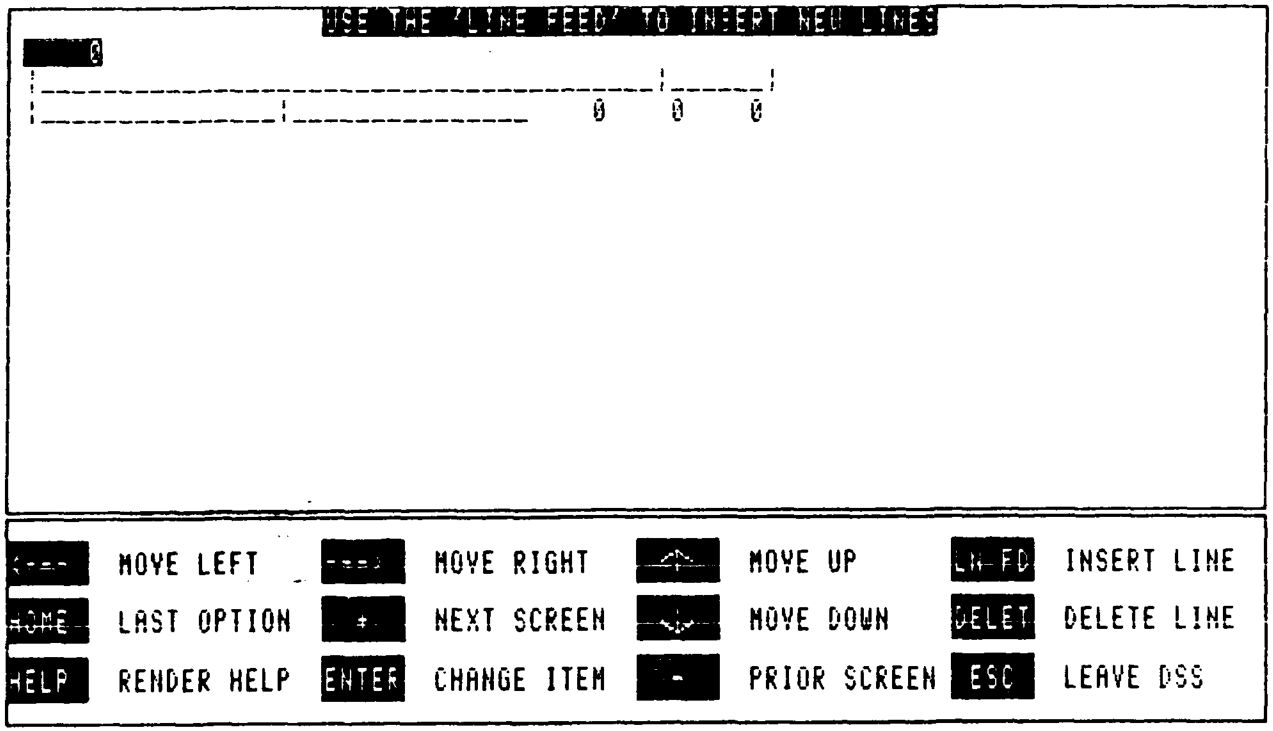
DOWN MOVE DOWN ONE MENU ITEM

HOME RETURN TO LAST SCREEN

HELP RENDER ASSISTANCE

Fig 23. Keyboard Entry Edit Selection Screen

The user wants to create a new file. The cursor is moved and the ENTER key depressed. While editing a file, the activity worksheets are available. These are single screen descriptions of the network nodes. The user selected "N".

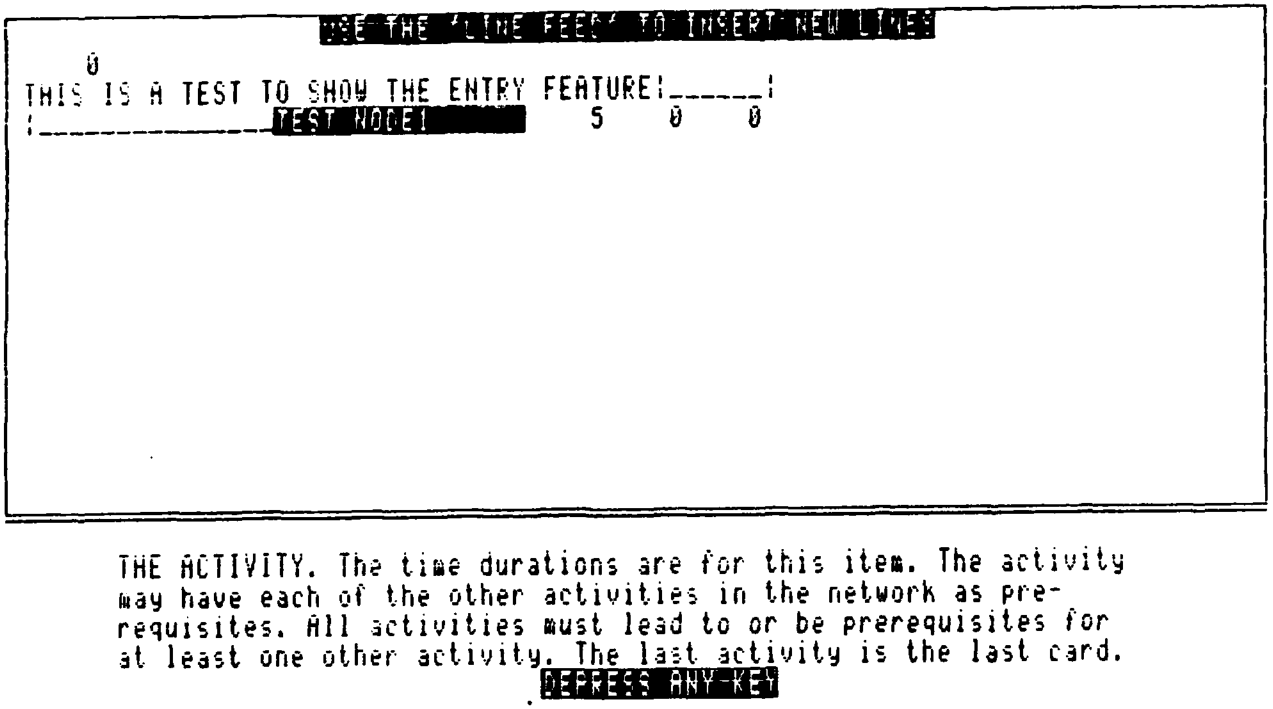


USE THE 'LINE FEED' TO INSERT NEW LINES

```
0
0  0  0
```

rig 24. Initial PERT or CPM Creation Screen

An empty file screen is initially presented. The cursor ("O") highlights the first data field in the file. The activated keys are presented on the lower portion of the screen.

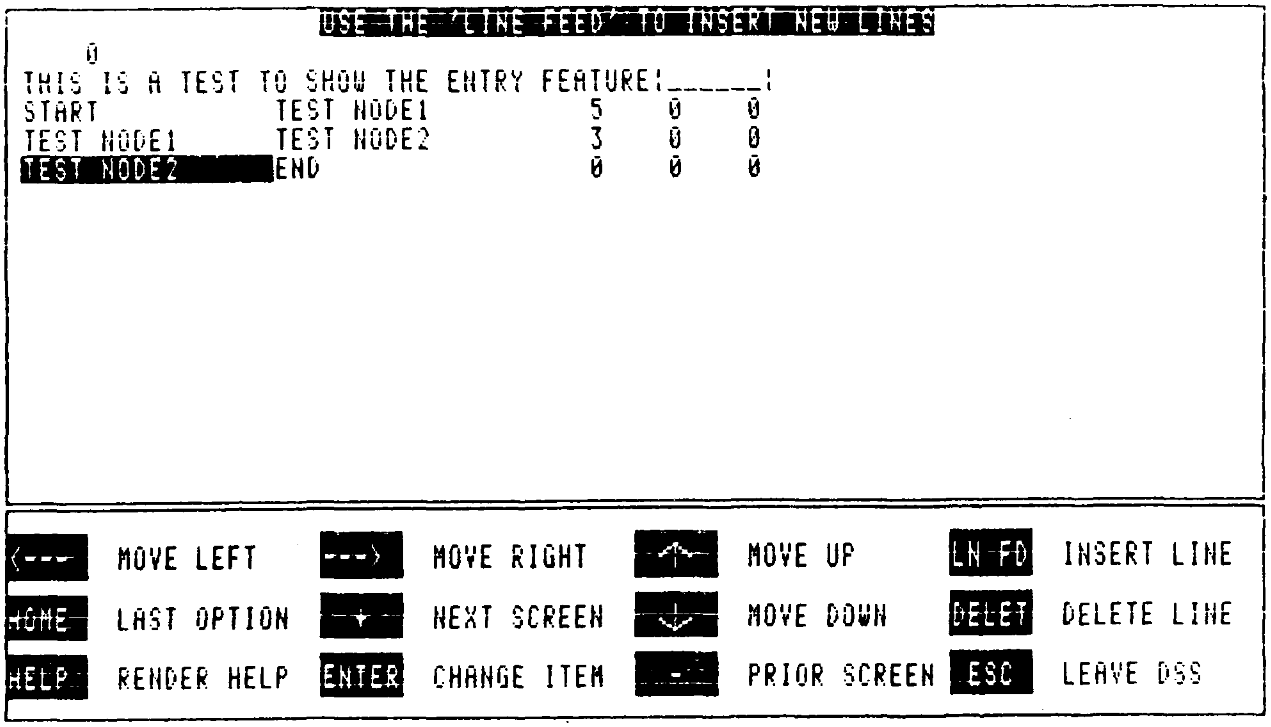


```
            USE THE 'LINE FEED' TO INSERT NEW LINES
    0
THIS IS A TEST TO SHOW THE ENTRY FEATURE!______!
!________________TEST NODE1              5    0    0
```

```
THE ACTIVITY. The time durations are for this item. The activity
may have each of the other activities in the network as pre-
requisites. All activities must lead to or be prerequisites for
at least one other activity. The last activity is the last card.
                            DEPRESS ANY-KEY
```

Fig 25. Edit Help Screen

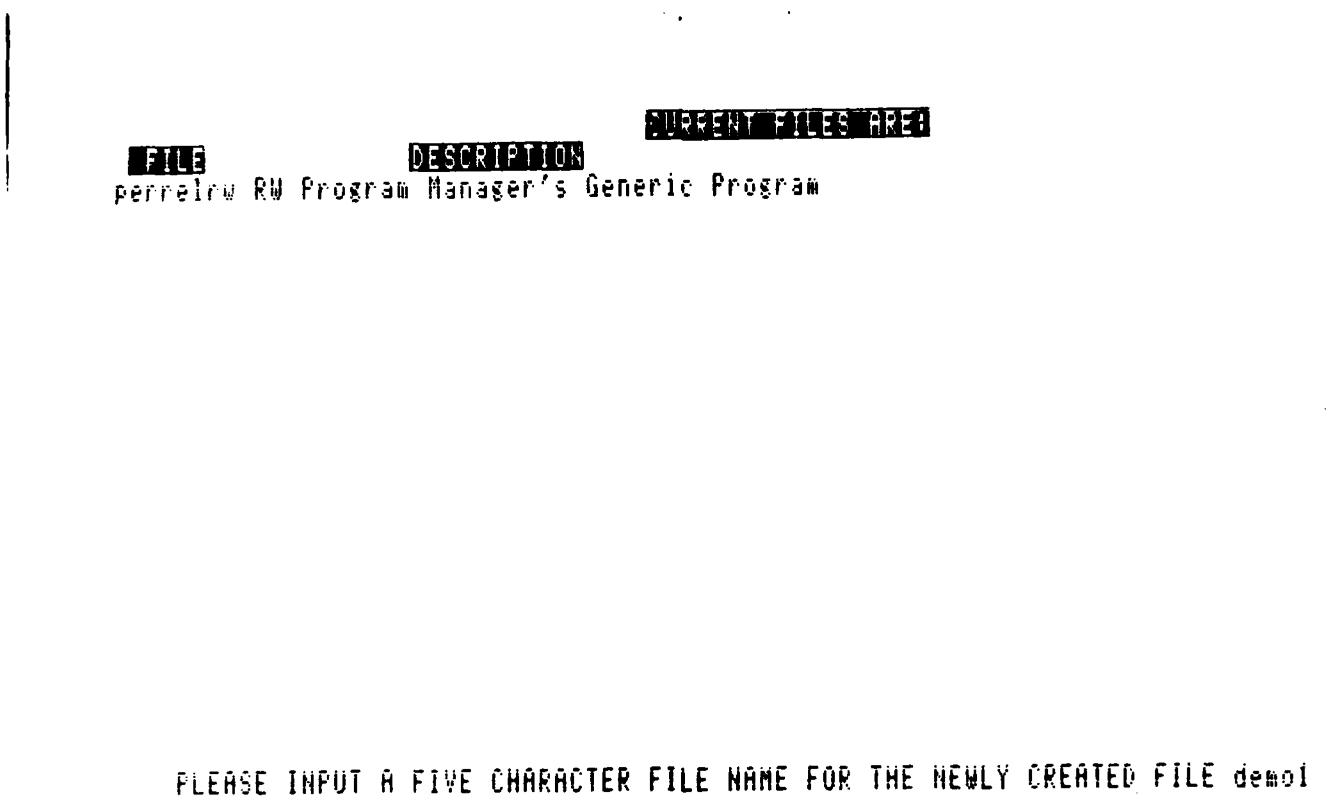
The HELP key has been depressed. The current highlighted data field is explained to the user. Every data field has a HELP message for all six ATs. The LINE FEED key is used to insert lines below the cursor line.



```
                    USE THE 'LINE FEED' TO INSERT NEW LINES
    0
THIS IS A TEST TO SHOW THE ENTRY FEATURE!______!
START           TEST NODE1            5    0    0
TEST NODE1      TEST NODE2            3    0    0
TEST NODE2      END                   0    0    0
```

Fig 26. Edit Insertion Screen

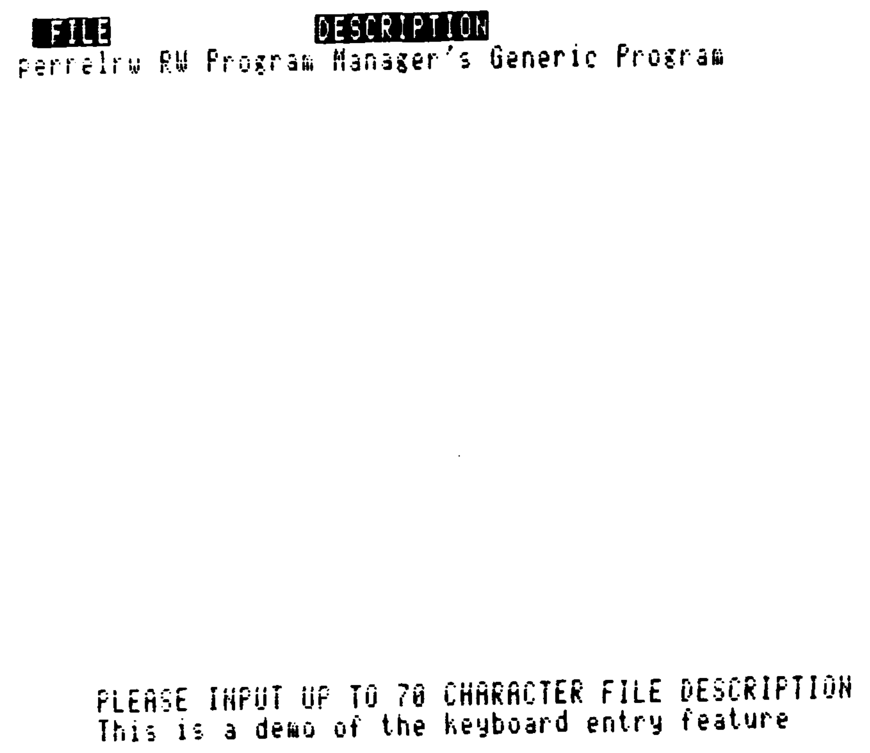
A third line has been added. The input stream is complete so the HOME key is depressed.



```
                                        CURRENT FILES ARE:
   FILE              DESCRIPTION
perrelrw RW Program Manager's Generic Program




   PLEASE INPUT A FIVE CHARACTER FILE NAME FOR THE NEWLY CREATED FILE demo1
```

Fig 27. New File Name Screen

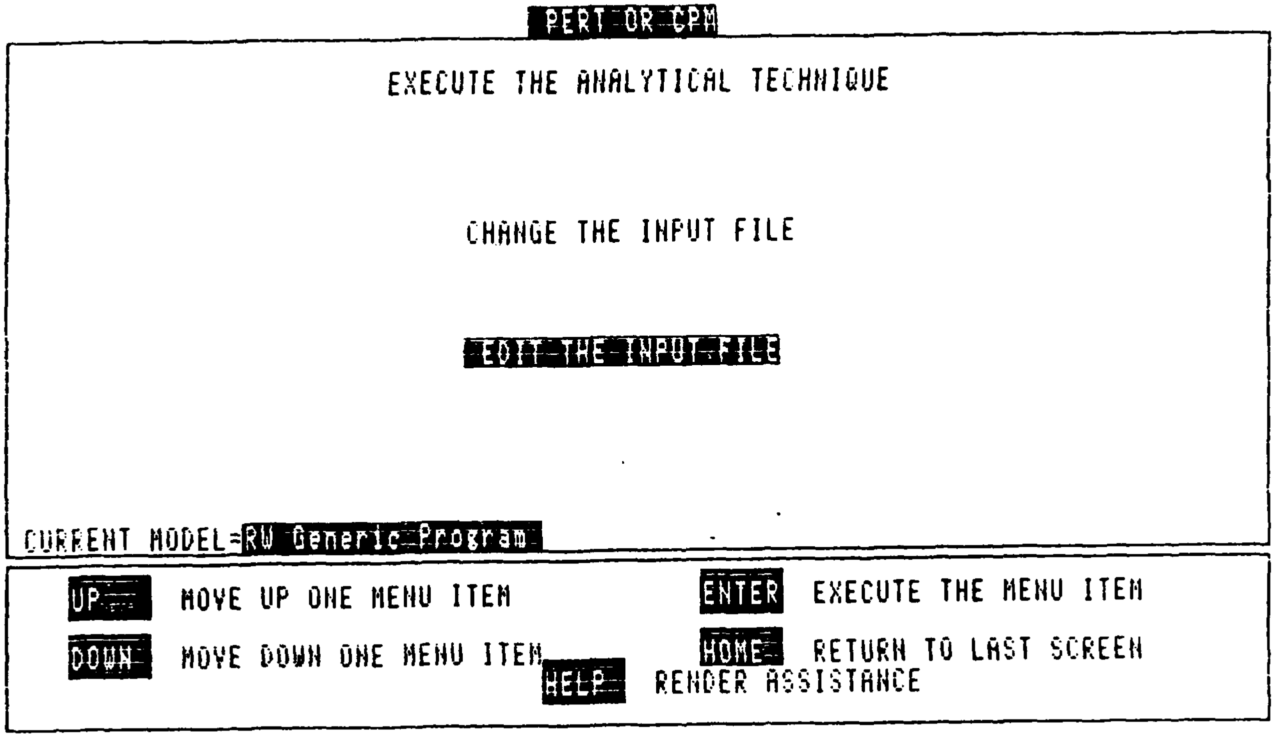
The file created for the keyboard needs a name for the system to recognize it. The current names are displayed to prevent the user from erasing one.



```
 FILE               DESCRIPTION
ferrelrw RW Program Manager's Generic Program



PLEASE INPUT UP TO 70 CHARACTER FILE DESCRIPTION
This is a demo of the keyboard entry feature
```

Fig 28. New File Label Screen

The new file also needs a label so the user can recognize it.

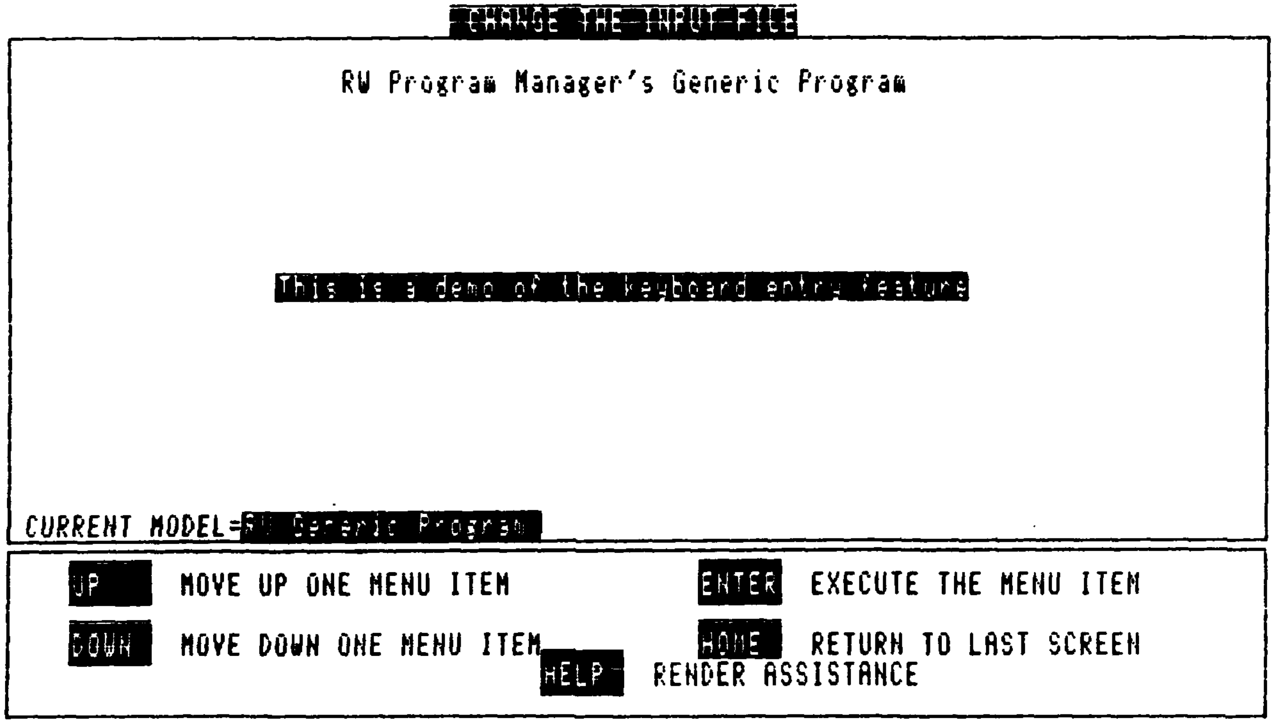


PERT OR CPM
EXECUTE THE ANALYTICAL TECHNIQUE
CHANGE THE INPUT FILE
EDIT THE INPUT FILE
UP MOVE UP ONE MENU ITEM
DOWN MOVE DOWN ONE MENU ITEM
ENTER EXECUTE THE MENU ITEM
HOME RETURN TO LAST SCREEN
HELP RENDER ASSISTANCE

CURRENT MODEL=RW Generic Program

Fig 29. Second Edit Option Screen

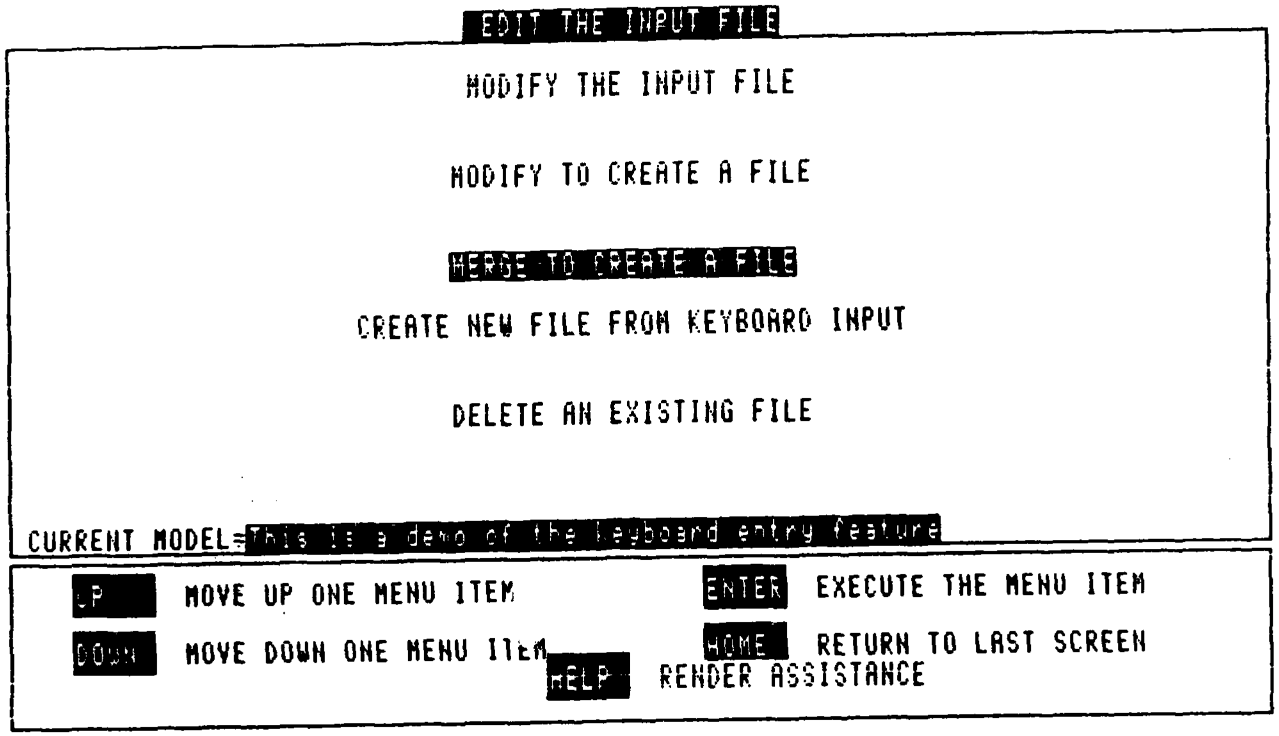
The file has been saved and the edit screen is presented. The user moves the cursor to the CHANGE INPUT FILE option to retrieve another input model.



```
CHANGE THE INPUT FILE

RW Program Manager's Generic Program

This is a demo of the keyboard entry feature

CURRENT MODEL=RW Generic Program

UP    MOVE UP ONE MENU ITEM          ENTER  EXECUTE THE MENU ITEM
DOWN  MOVE DOWN ONE MENU ITEM        HOME   RETURN TO LAST SCREEN
                              HELP  RENDER ASSISTANCE
```

Fig 30. Change Input File Screen

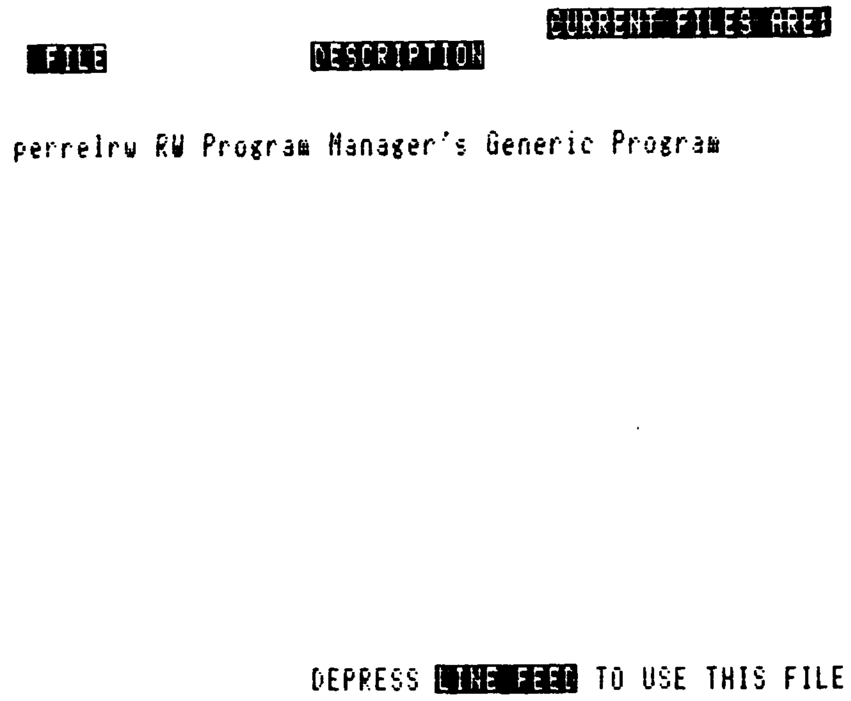
The existing models are displayed. The created model is selected to be used for further edit and execution options.



CURRENT MODEL=This is a demo of the keyboard entry feature

Fig 31. Third Edit Option Screen

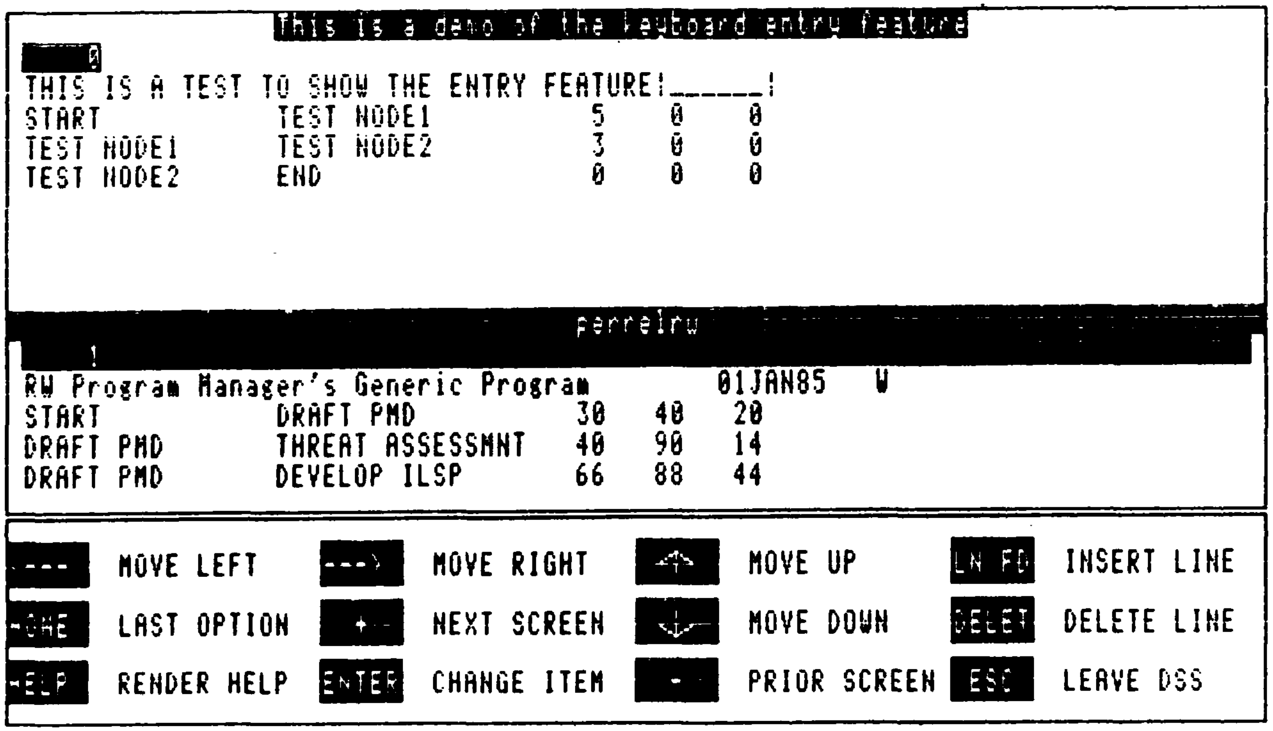
The CURRENT MODEL has changed to the selected input model. The user will merge lines from a merge file into the current model in order to create a third file.



```
                                    CURRENT FILES ARE:
 FILE           DESCRIPTION

perrelrw RW Program Manager's Generic Program




                DEPRESS LINE FEED TO USE THIS FILE
```

Fig 32. Merge File Selection Screen

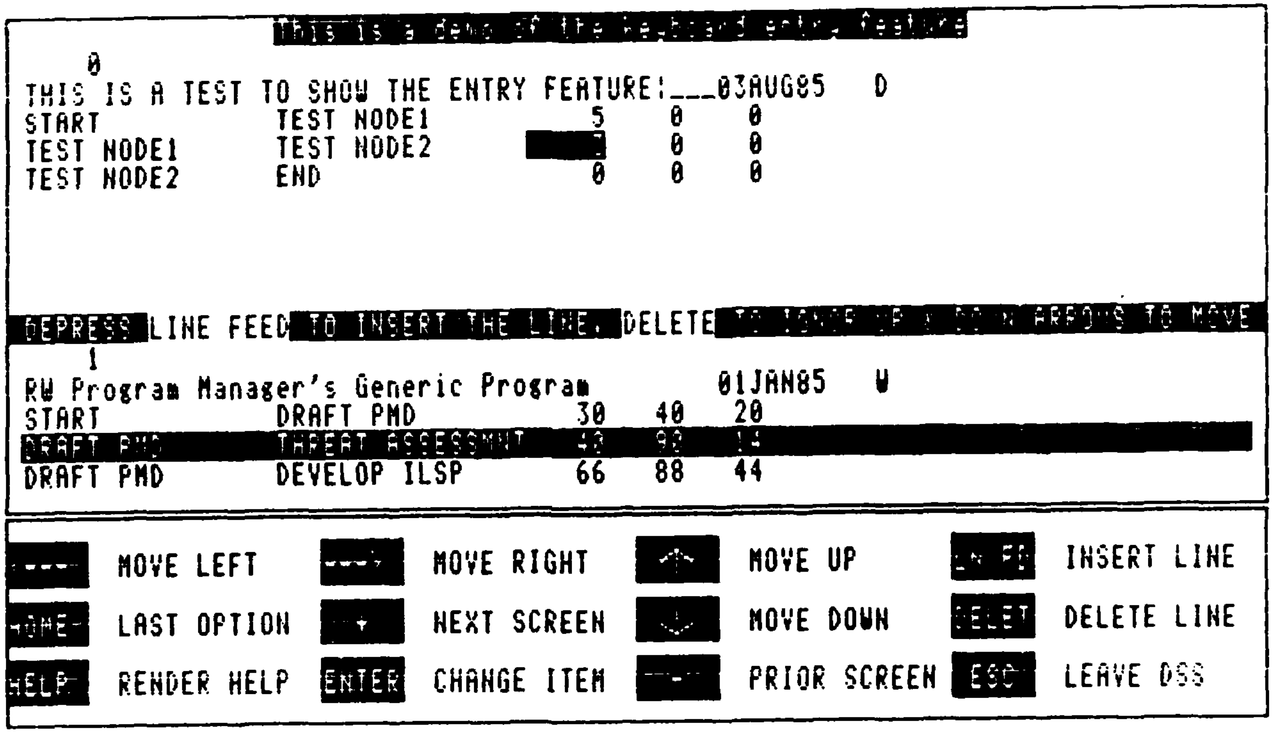
The user selects the file to merge from. The chosen file is indicated by depressing the LINE FEED key.



```
              This is a demo of the keyboard entry feature
      0
THIS IS A TEST TO SHOW THE ENTRY FEATURE!______!
START            TEST NODE1           5    0    0
TEST NODE1       TEST NODE2           3    0    0
TEST NODE2       END                  0    0    0

                              perrelru
      1
RW Program Manager's Generic Program         01JAN85   W
START            DRAFT PMD           30   40   20
DRAFT PMD        THREAT ASSESSMNT    40   90   14
DRAFT PMD        DEVELOP ILSP        66   88   44
```

Fig 33. Initial Merge Edit Screen

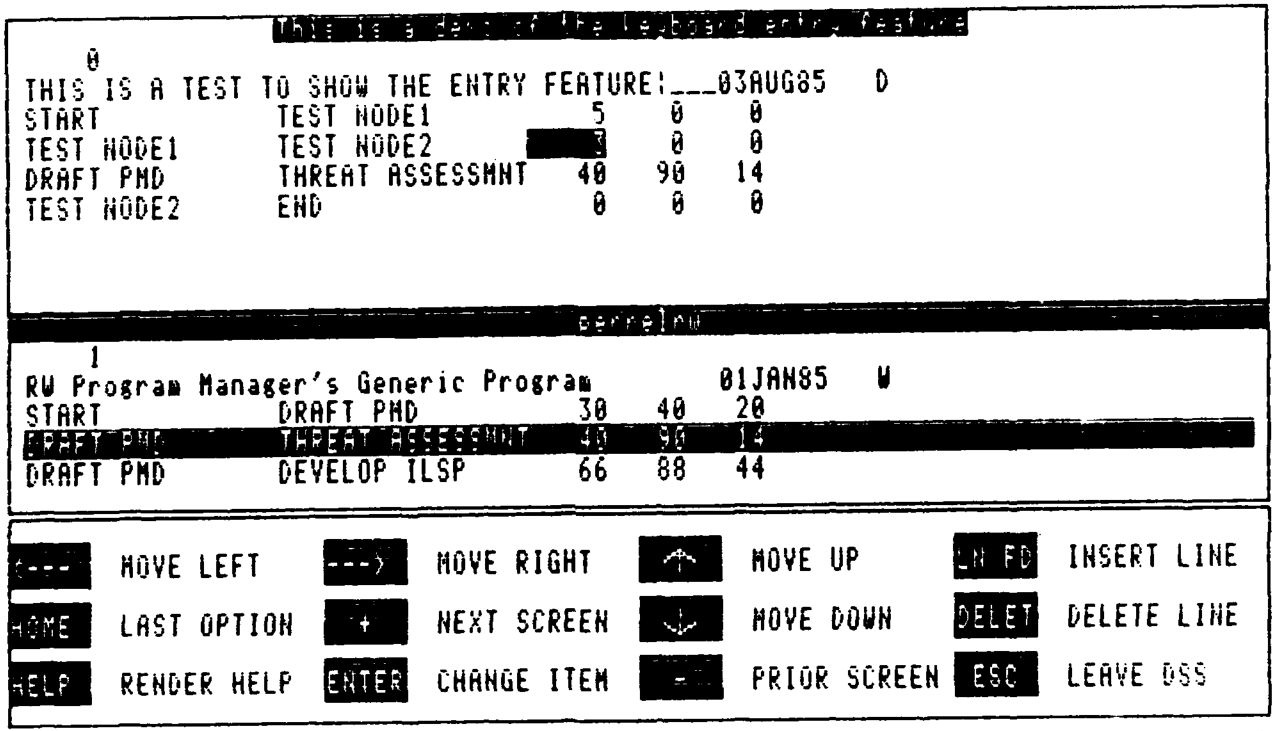
The merge file appears in the lower portion of the EDIT screen. The current model is in the top portion. The Merge cursor and Edit cursor are highlighted. The Edit cursor is activated. The arrow keys move the top cursor.



```
                    This is a demo of the keyboard entry feature
   0
THIS IS A TEST TO SHOW THE ENTRY FEATURE!___03AUG85   D
START            TEST NODE1           5    0    0
TEST NODE1       TEST NODE2           3    0    0
TEST NODE2       END                  0    0    0



DEPRESS LINE FEED TO INSERT THE LINE. DELETE TO IGNORE, UP & DOWN ARROWS TO MOVE
   1
RW Program Manager's Generic Program        01JAN85   W
START            DRAFT PMD           30   40   20
DRAFT PMD        THREAT ASSESSMNT    40   90   14
DRAFT PMD        DEVELOP ILSP        66   88   44
```

Fig 34. Merge Option, Line Insertion Screen

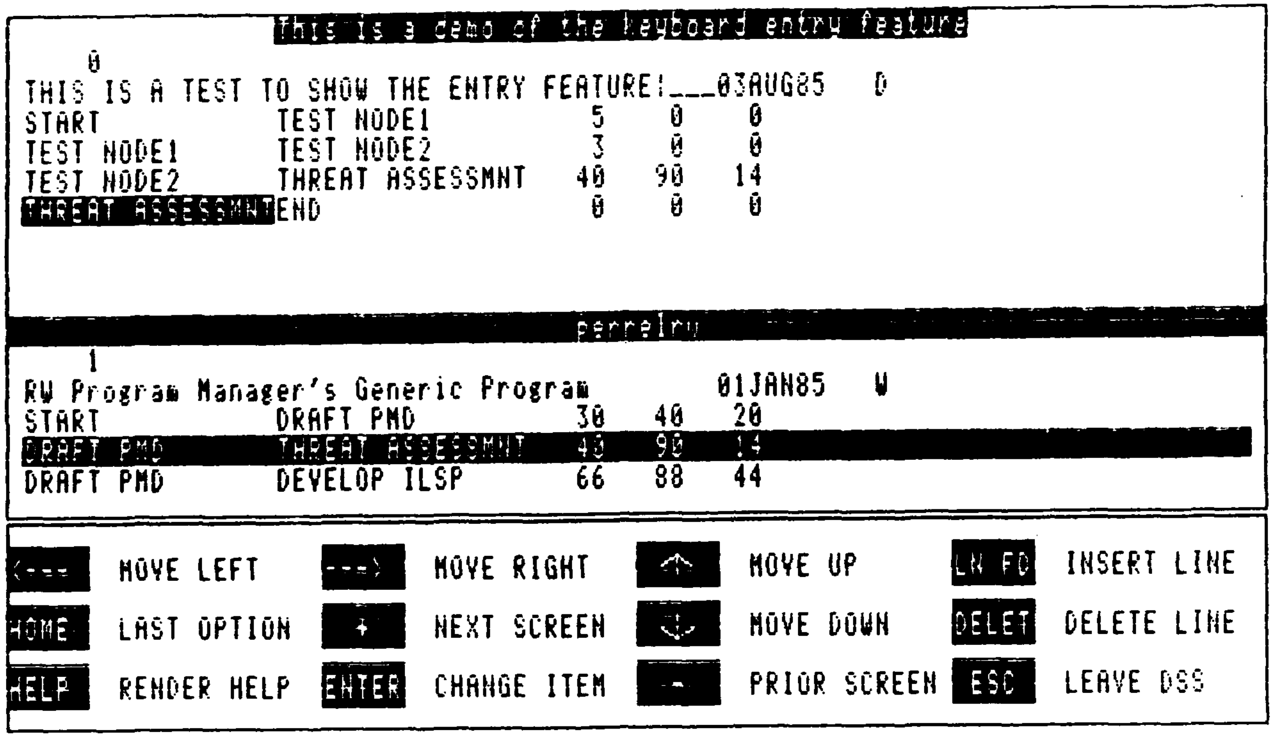
The activated Edit cursor is moved to the line above (highlighted "3") the destination for the insertion line. The LINE FEED key is depressed, and the Merge cursor is activated. The insertion line is selected by depressing LINE FEED.



```
            This is a demo of the keyboard entry feature
    0
THIS IS A TEST TO SHOW THE ENTRY FEATURE!___03AUG85   D
START            TEST NODE1               5     0     0
TEST NODE1       TEST NODE2               3     0     0
DRAFT PMD        THREAT ASSESSMNT        40    90    14
TEST NODE2       END                      0     0     0

                              genrelnw
    1
RW Program Manager's Generic Program           01JAN85   W
START            DRAFT PMD               30    40    20
DRAFT PMD        THREAT ASSESSMNT        40    90    14
DRAFT PMD        DEVELOP ILSP            66    88    44
```

Fig 35. Merge Option, Second Line Insertion Screen

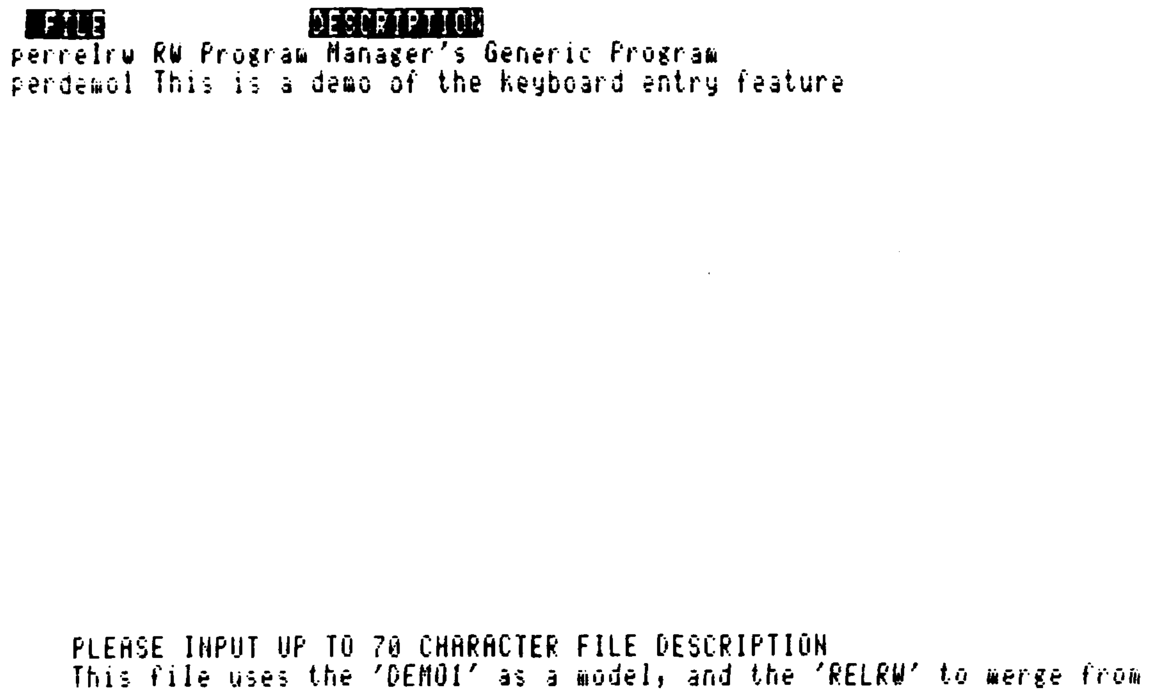
The merged line is copied to the current model file. The Edit cursor is reactivated.



```
                         This is a demo of the keyboard entry feature
   0
THIS IS A TEST TO SHOW THE ENTRY FEATURE!___03AUG85   D
START           TEST NODE1          5    0    0
TEST NODE1      TEST NODE2          3    0    0
TEST NODE2      THREAT ASSESSMNT   40   90   14
THREAT ASSESSMNTEND                 0    0    0

                                     perrelry
   1
RW Program Manager's Generic Program           01JAN85   W
START           DRAFT PMD          30   40   20
DRAFT PMD       THREAT ASSESSMNT   40   90   14
DRAFT PMD       DEVELOP ILSP       66   88   44
```

Fig 36. Merge Option, File Edit Screen

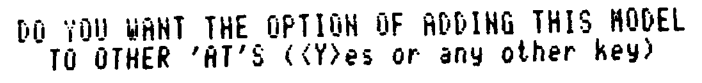
The prerequisite activity is changed to "THREAT ASSESSMNT". All activities must have a prerequisite and lead to another activity for the network to be complete. The AT will identify "loose ends". The file is finished, so HOME is depressed.



```
 FILE               DESCRIPTION
perrelrw RW Program Manager's Generic Program
perdemo1 This is a demo of the keyboard entry feature




PLEASE INPUT UP TO 70 CHARACTER FILE DESCRIPTION
This file uses the 'DEMO1' as a model, and the 'RELRW' to merge from
```

Fig 37. Second File Label Screen

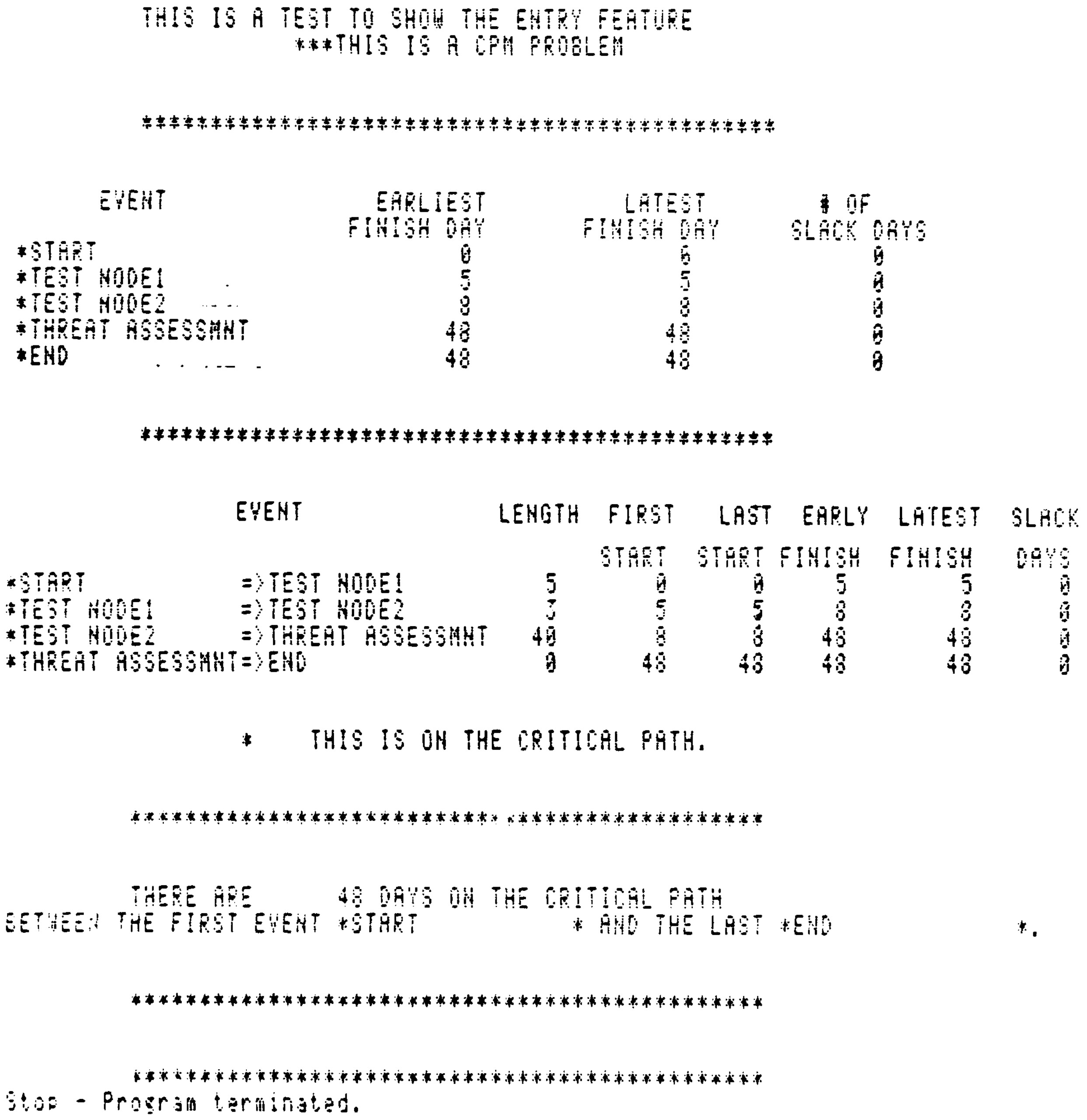
The created file needs a label to identify it.



DO YOU WANT THE OPTION OF ADDING THIS MODEL
TO OTHER 'AT'S (<Y>es or any other key)

Fig 38. Model Share Screen

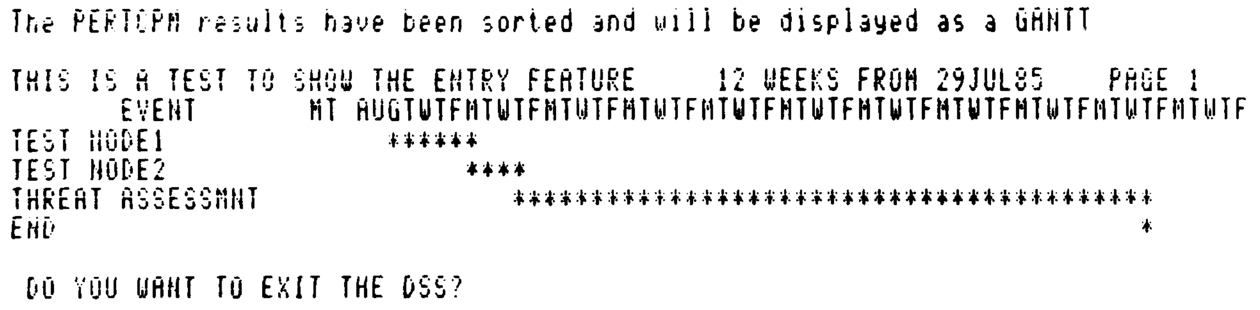
The user can share the current model with the other five ATs by answering "y" to the question. "Y" was depressed since the user wants to transfer the model to the "PERT or CPM Tied To a GANNT" AT.



```
               THIS IS A TEST TO SHOW THE ENTRY FEATURE
                             ***THIS IS A CPM PROBLEM


               ********************************************


       EVENT                          EARLIEST          LATEST        # OF
                                     FINISH DAY       FINISH DAY    SLACK DAYS
*START                                    0                0             0
*TEST NODE1                               5                5             0
*TEST NODE2                               8                8             0
*THREAT ASSESSMNT                        48               48             0
*END                                     48               48             0


               ********************************************


                 EVENT                  LENGTH  FIRST   LAST  EARLY  LATEST  SLACK

                                                START  START FINISH  FINISH   DAYS
*START            =>TEST NODE1             5       0      0     5       5      0
*TEST NODE1       =>TEST NODE2             3       5      5     8       8      0
*TEST NODE2       =>THREAT ASSESSMNT      40       8      8    48      48      0
*THREAT ASSESSMNT=>END                     0      48     48    48      48      0


                 *    THIS IS ON THE CRITICAL PATH.


              ********************************************


              THERE ARE      48 DAYS ON THE CRITICAL PATH
BETWEEN THE FIRST EVENT *START           * AND THE LAST *END            *.


              ********************************************


              ********************************************
Stop - Program terminated.
```

Fig 39. PERT Output

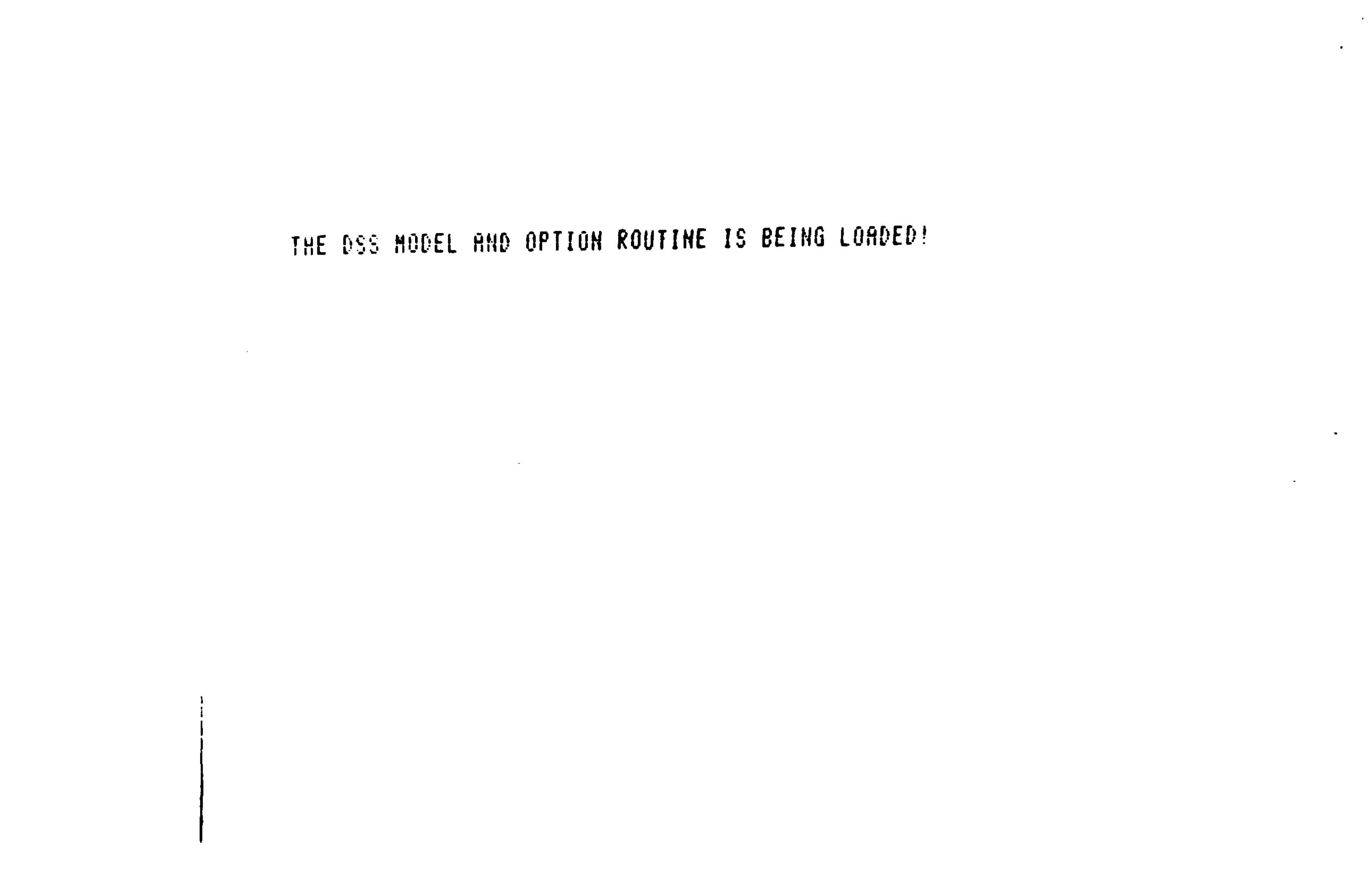
The "*" indicates the critical path. The second chart identifies two activities. The first one (before the =>) is the prerequisite for the second. If only that prerequisite were required the activity could start on the indicated FIRST START.



```
The PERTCPM results have been sorted and will be displayed as a GANTT

THIS IS A TEST TO SHOW THE ENTRY FEATURE     12 WEEKS FROM 29JUL85    PAGE 1
       EVENT           MT AUGTWTFMTWTFMTUTFMTWTFMTWTFMTWTFMTWTFMTWTFMTWTFMTWTFMTWTFMTWTF
TEST NODE1              ******
TEST NODE2                   ****
THREAT ASSESSMNT                ******************************************
END                                                                      *

 DO YOU WANT TO EXIT THE DSS?
```

Fig 40. GANTT Output

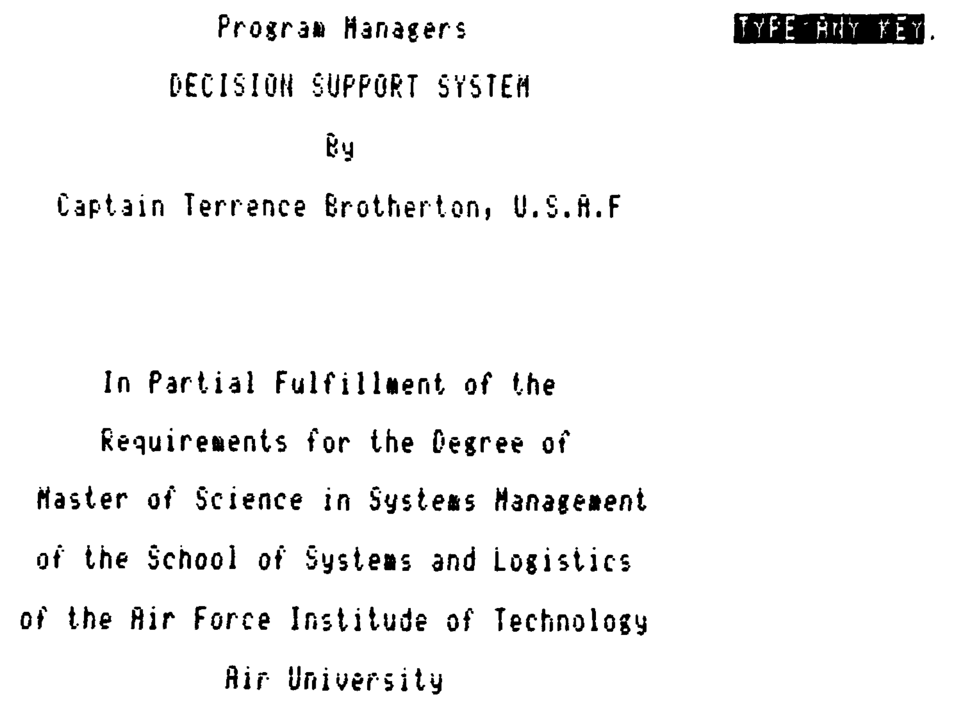
The PERT output is displayed as a GANTT chart. The GANTT output option was "D"(ays), so the WEEKDAYS are shown. Units could have been "W"eeks, "M"onths, or "Q"uarters. Each column represents 1 (one) selected unit of time.



THE DSS MODEL AND OPTION ROUTINE IS BEING LOADED!

Fig 41. Second Option Routine Load Screen

The user replied "n" to the "DO YOU WANT TO LEAVE THE DSS" question. The option routine is reloaded.



Program Managers

DECISION SUPPORT SYSTEM

By

Captain Terrence Brotherton, U.S.A.F

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Systems Management

of the School of Systems and Logistics

of the Air Force Institude of Technology

Air University

TYPE ANY KEY.

Fig 42. Second Controller Introduction Screen

The DSS Controller/User Interface has been loaded and is ready to conduct the terminal session. The user depresses any key to continue.

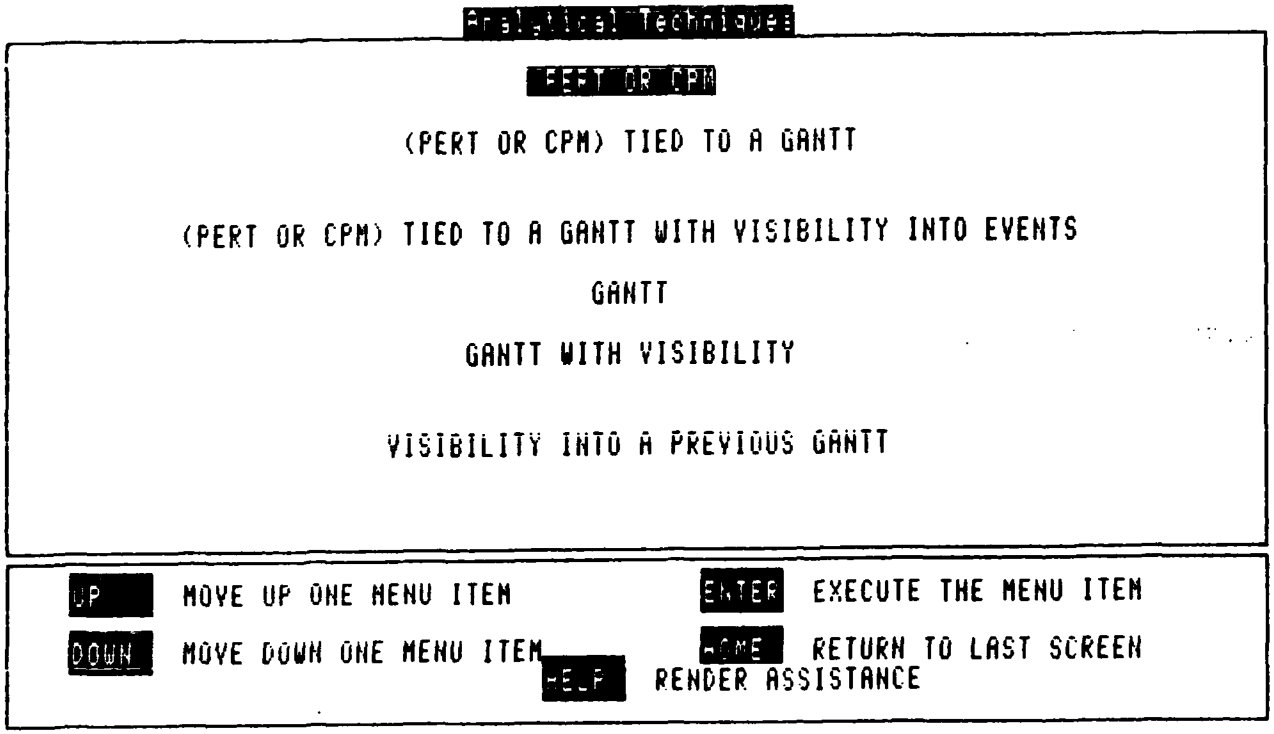

Analytical Techniques
PERT OR CPM
(PERT OR CPM) TIED TO A GANTT
(PERT OR CPM) TIED TO A GANTT WITH VISIBILITY INTO EVENTS
GANTT
GANTT WITH VISIBILITY
VISIBILITY INTO A PREVIOUS GANTT
UP MOVE UP ONE MENU ITEM
DOWN MOVE DOWN ONE MENU ITEM
ENTER EXECUTE THE MENU ITEM
HOME RETURN TO LAST SCREEN
HELP RENDER ASSISTANCE

Fig 43. Second Interaction Initial Analytical Technique Screen

The six Analytical Techniques are presented. The user activated keys appear in the lower portion of the screen. The "UP" and "DOWN" refers to the arrow keys.

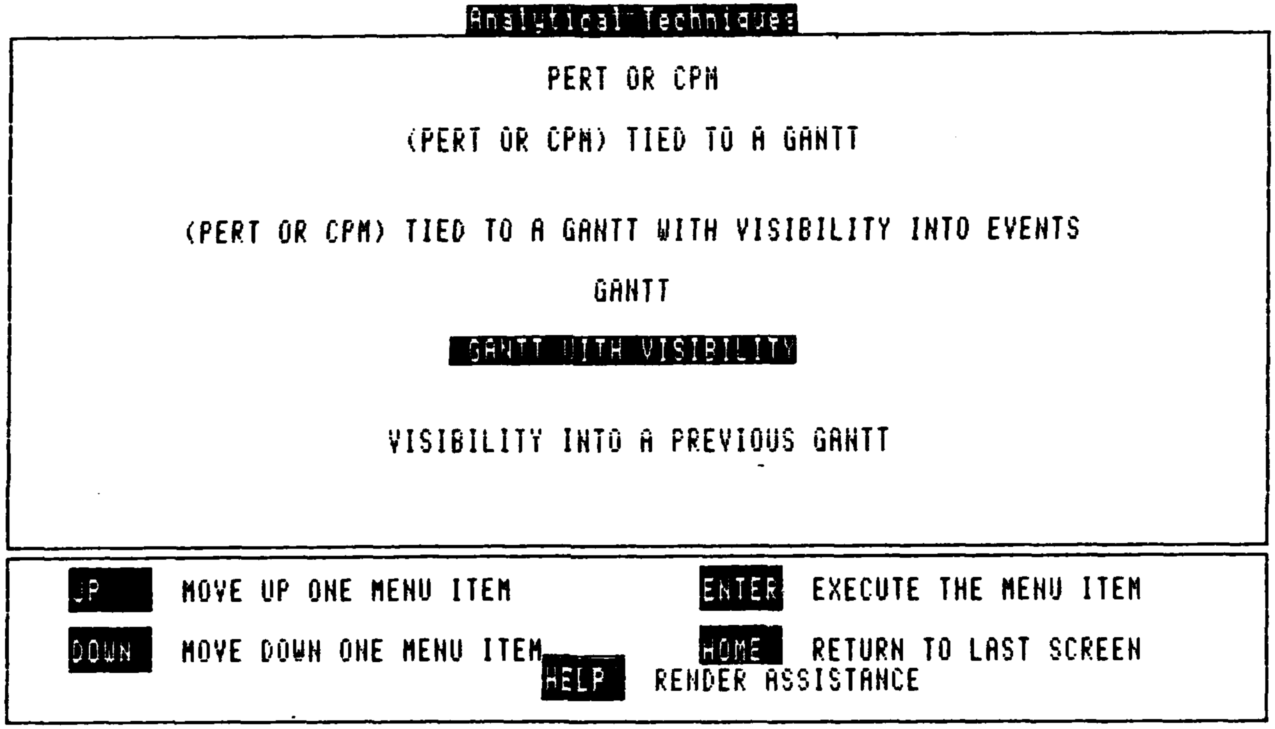
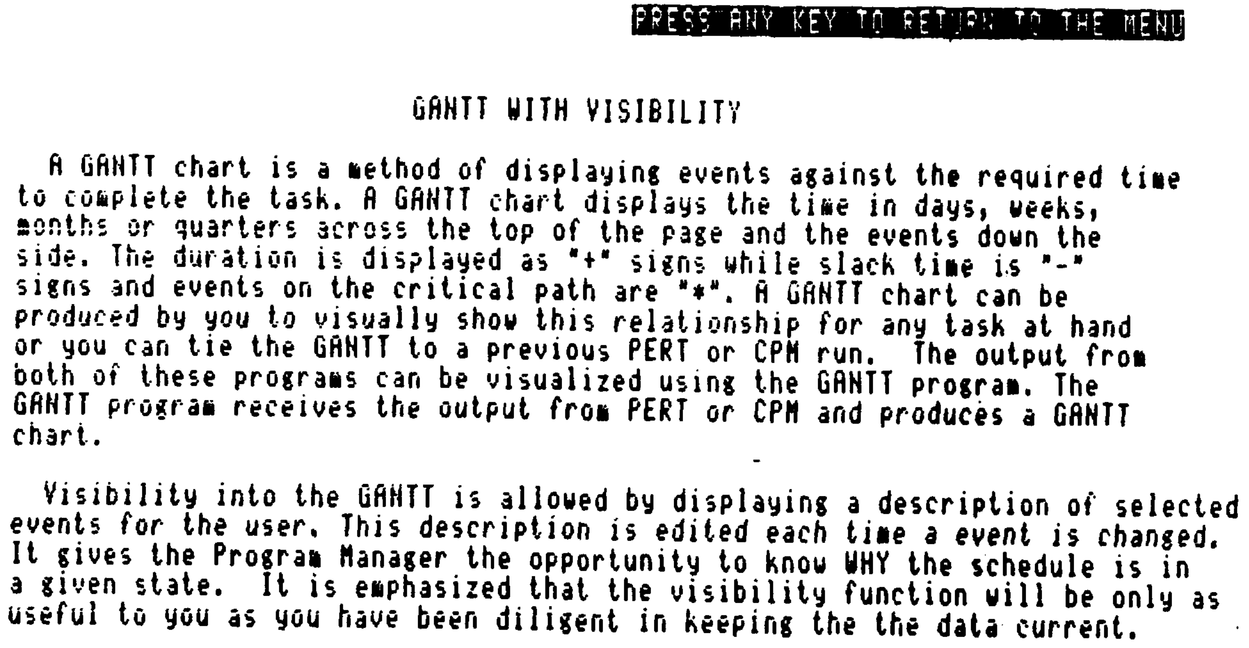

Analytical Techniques
PERT OR CPM
(PERT OR CPM) TIED TO A GANTT
(PERT OR CPM) TIED TO A GANTT WITH VISIBILITY INTO EVENTS
GANTT
GANTT WITH VISIBILITY
VISIBILITY INTO A PREVIOUS GANTT
UP MOVE UP ONE MENU ITEM
DOWN MOVE DOWN ONE MENU ITEM
ENTER EXECUTE THE MENU ITEM
HOME RETURN TO LAST SCREEN
HELP RENDER ASSISTANCE

Fig 44. GANTT WITH VISIBILITY Selection Screen

The menu cursor is moved and the HELP key is depressed.



PRESS ANY KEY TO RETURN TO THE MENU

GANTT WITH VISIBILITY

A GANTT chart is a method of displaying events against the required time to complete the task. A GANTT chart displays the time in days, weeks, months or quarters across the top of the page and the events down the side. The duration is displayed as "+" signs while slack time is "-" signs and events on the critical path are "*". A GANTT chart can be produced by you to visually show this relationship for any task at hand or you can tie the GANTT to a previous PERT or CPM run. The output from both of these programs can be visualized using the GANTT program. The GANTT program receives the output from PERT or CPM and produces a GANTT chart.

Visibility into the GANTT is allowed by displaying a description of selected events for the user. This description is edited each time a event is changed. It gives the Program Manager the opportunity to know WHY the schedule is in a given state. It is emphasized that the visibility function will be only as useful to you as you have been diligent in keeping the the data current.

Fig 45. GANTT WITH VISIBILITY Help Screen

The Help message for the GANTT with visibility AT selection menu item is presented. Every menu selection item has it's own help screen. These can be modified with a standard word processor, such as WORDSTAR.

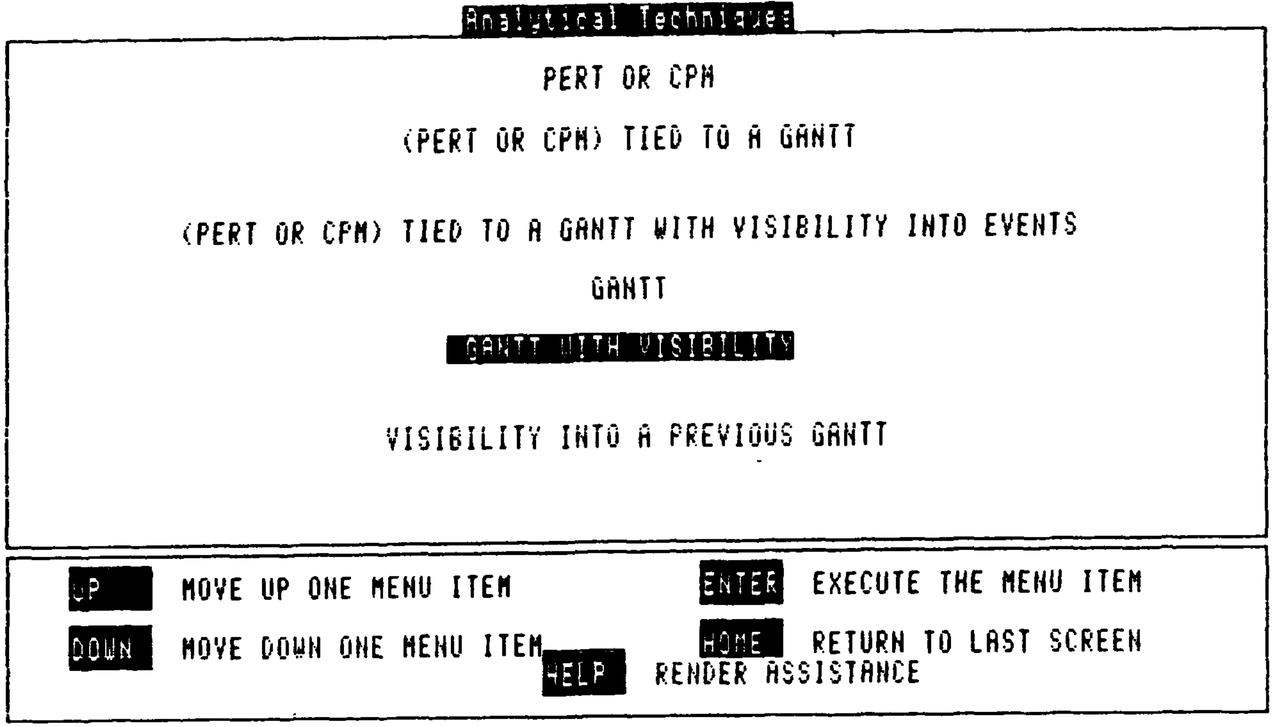
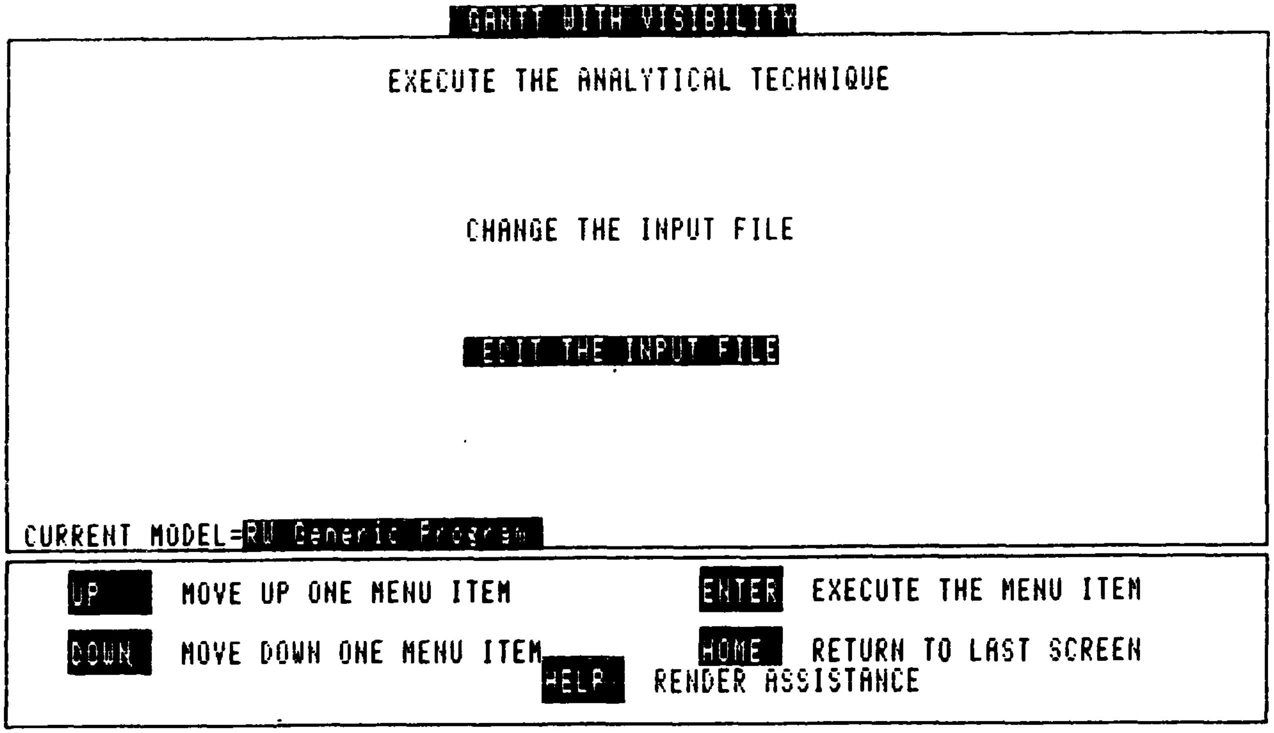

Analytical Techniques
PERT OR CPM
(PERT OR CPM) TIED TO A GANTT
(PERT OR CPM) TIED TO A GANTT WITH VISIBILITY INTO EVENTS
GANTT
GANTT WITH VISIBILITY
VISIBILITY INTO A PREVIOUS GANTT
UP MOVE UP ONE MENU ITEM
DOWN MOVE DOWN ONE MENU ITEM
ENTER EXECUTE THE MENU ITEM
HOME RETURN TO LAST SCREEN
HELP RENDER ASSISTANCE

Fig 46. Second GANTT WITH VISIBILITY Selection Screen

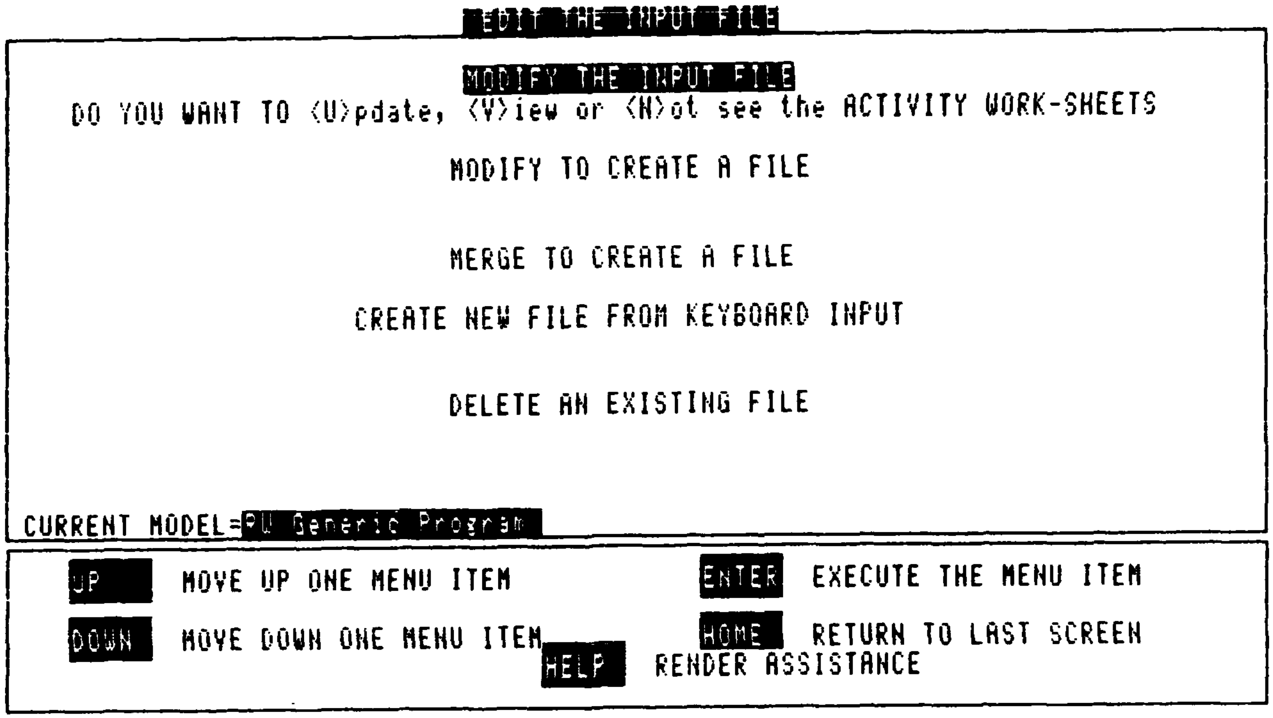
The menu selection screen is redisplayed upon a key strike from the Help message. The user selects this option as the AT to use.


GANTT WITH VISIBILITY
EXECUTE THE ANALYTICAL TECHNIQUE
CHANGE THE INPUT FILE
EDIT THE INPUT FILE
UP MOVE UP ONE MENU ITEM
DOWN MOVE DOWN ONE MENU ITEM
ENTER EXECUTE THE MENU ITEM
HOME RETURN TO LAST SCREEN
HELP RENDER ASSISTANCE

CURRENT MODEL=RW Generic Program

Fig 47. GANTT WITH VISIBILITY Option Screen

The user selects to Edit the CURRENT MODEL. The current input stream is the default of the RW Generic Program



EDIT THE INPUT FILE

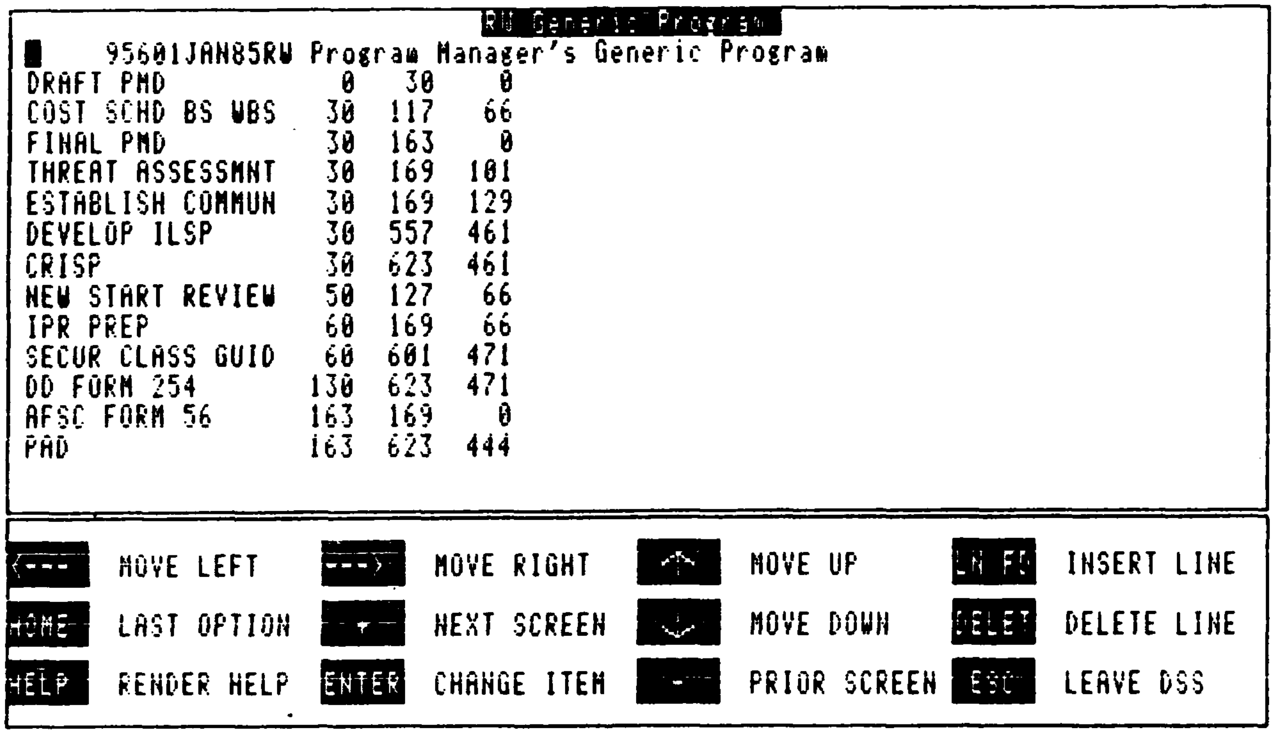
MODIFY THE INPUT FILE

DO YOU WANT TO <U>pdate, <V>iew or <N>ot see the ACTIVITY WORK-SHEETS

CURRENT MODEL=PU Generic Program

Fig 48. Modify Edit Selection Screen

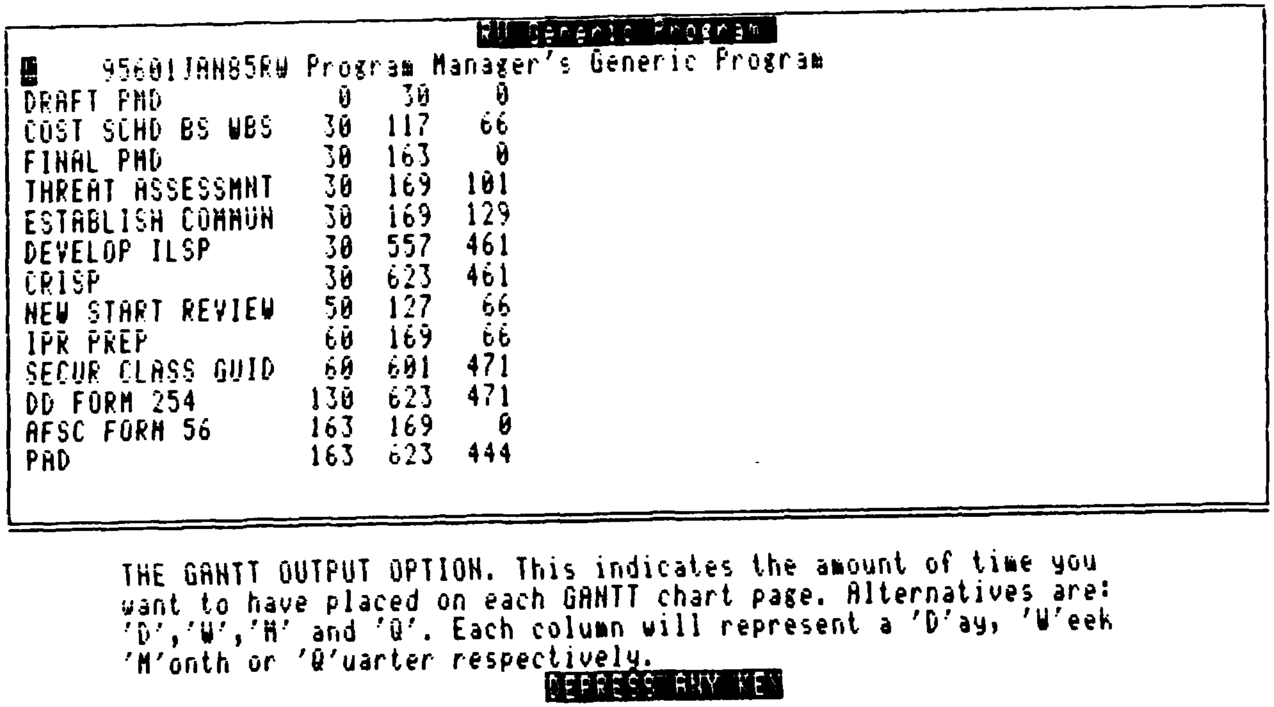
The CURRENT MODEL will be modified. The user replies that he wants to "U"pdate the activity worksheets.



```
                                     RW Generic Program
█    95601JAN85RW Program Manager's Generic Program
DRAFT PMD            0   30    0
COST SCHD BS WBS    30  117   66
FINAL PMD           30  163    0
THREAT ASSESSMNT    30  169  101
ESTABLISH COMMUN    30  169  129
DEVELOP ILSP        30  557  461
CRISP               30  623  461
NEW START REVIEW    50  127   66
IPR PREP            60  169   66
SECUR CLASS GUID    60  601  471
DD FORM 254        130  623  471
AFSC FORM 56       163  169    0
PAD                163  623  444
```

Fig 49. GANTT Edit Screen

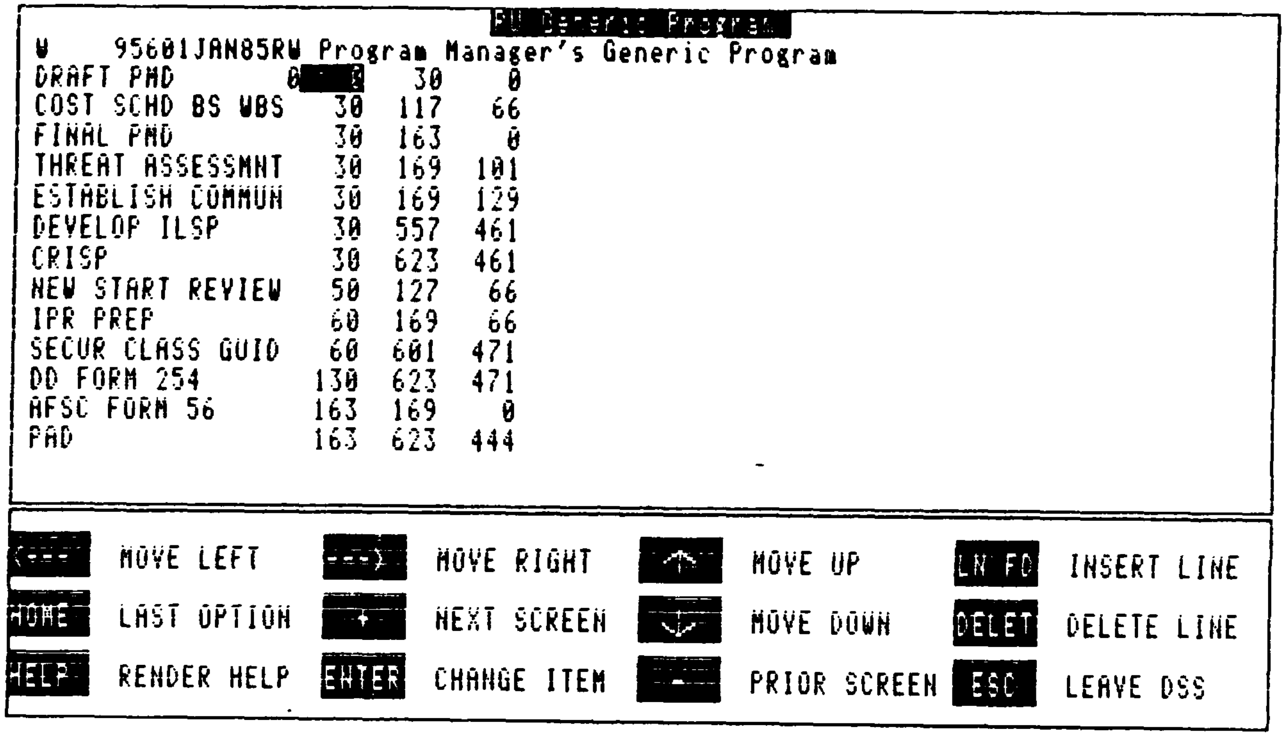
The default GANTT input stream is presented. The Edit cursor highlights the first data field in the file.



```
                                          RW Generic Program
█    95601JAN85RW Program Manager's Generic Program
DRAFT PMD            0   30    0
COST SCHD BS WBS    30  117   66
FINAL PMD           30  163    0
THREAT ASSESSMNT    30  169  101
ESTABLISH COMMUN    30  169  129
DEVELOP ILSP        30  557  461
CRISP               30  623  461
NEW START REVIEW    50  127   66
IPR PREP            60  169   66
SECUR CLASS GUID    60  601  471
DD FORM 254        130  623  471
AFSC FORM 56       163  169    0
PAD                163  623  444

      THE GANTT OUTPUT OPTION. This indicates the amount of time you
      want to have placed on each GANTT chart page. Alternatives are:
      'D','W','M' and 'Q'. Each column will represent a 'D'ay, 'W'eek
      'M'onth or 'Q'uarter respectively.
                                     DEPRESS ANY KEY
```

Fig 50. GANTT Edit Help Screen

The Help key was depressed. An Explanatory message about the highlighted data field, GANTT output option, is displayed.

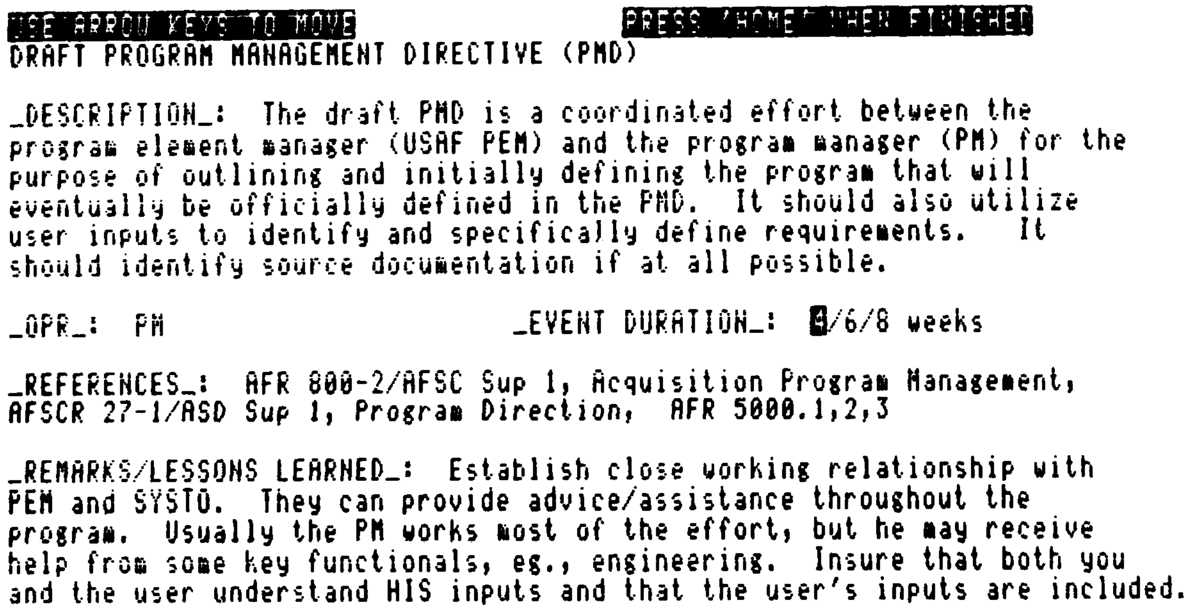


```
                                   PW Generic Program
W    95601JAN85RW Program Manager's Generic Program
DRAFT PMD          0   0   30    0
COST SCHD BS WBS      30  117   66
FINAL PMD             30  163    0
THREAT ASSESSMNT      30  169  101
ESTABLISH COMMUN      30  169  129
DEVELOP ILSP          30  557  461
CRISP                 30  623  461
NEW START REVIEW      50  127   66
IPR PREP              60  169   66
SECUR CLASS GUID      60  601  471
DD FORM 254          130  623  471
AFSC FORM 56         163  169    0
PAD                  163  623  444
```

```
<---  MOVE LEFT     --->   MOVE RIGHT    ↑  MOVE UP        LN FD  INSERT LINE
HOME  LAST OPTION    +     NEXT SCREEN   ↓  MOVE DOWN      DELET  DELETE LINE
HELP  RENDER HELP   ENTER  CHANGE ITEM   -  PRIOR SCREEN   ESC    LEAVE DSS
```

Fig 51. GANTT Data Modification Screen

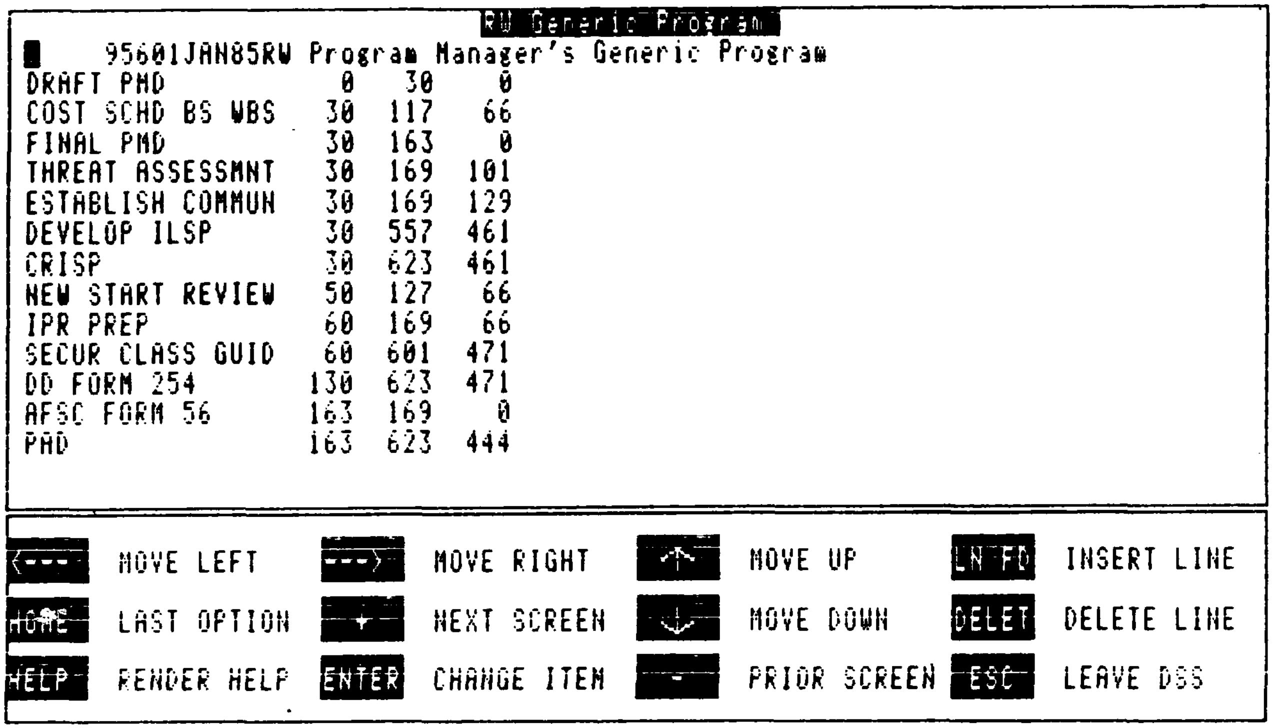
The highlighted data field is changed from the current value of "0" (highlighted) to a new value of "0".



```
USE ARROW KEYS TO MOVE                    PRESS 'HOME' WHEN FINISHED
DRAFT PROGRAM MANAGEMENT DIRECTIVE (PMD)

_DESCRIPTION_:  The draft PMD is a coordinated effort between the
program element manager (USAF PEM) and the program manager (PM) for the
purpose of outlining and initially defining the program that will
eventually be officially defined in the PMD.  It should also utilize
user inputs to identify and specifically define requirements.   It
should identify source documentation if at all possible.

_OPR_:  PM                          _EVENT DURATION_:  4/6/8 weeks

_REFERENCES_:  AFR 800-2/AFSC Sup 1, Acquisition Program Management,
AFSCR 27-1/ASD Sup 1, Program Direction,  AFR 5000.1,2,3

_REMARKS/LESSONS LEARNED_:  Establish close working relationship with
PEM and SYSTO.  They can provide advice/assistance throughout the
program.  Usually the PM works most of the effort, but he may receive
help from some key functionals, eg., engineering.  Insure that both you
and the user understand HIS inputs and that the user's inputs are included.
```

Fig 52. Worksheet Display Screen

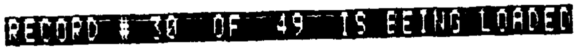
Since the user wants to keep the worksheets updated upon data modification, the activity worksheet is presented. The worksheet cursor can be moved and the screen can be editted. The cursor is highlighting the minimum duration of "4" weeks.



```
RW Generic Program
█    95601JAN85RW Program Manager's Generic Program
DRAFT PMD            0   30    0
COST SCHD BS WBS    30  117   66
FINAL PMD           30  163    0
THREAT ASSESSMNT    30  169  101
ESTABLISH COMMUN    30  169  129
DEVELOP ILSP        30  557  461
CRISP               30  623  461
NEW START REVIEW    50  127   66
IPR PREP            60  169   66
SECUR CLASS GUID    60  601  471
DD FORM 254        130  623  471
AFSC FORM 56       163  169    0
PAD                163  623  444
```

Fig 53. Second GANTT Edit Screen

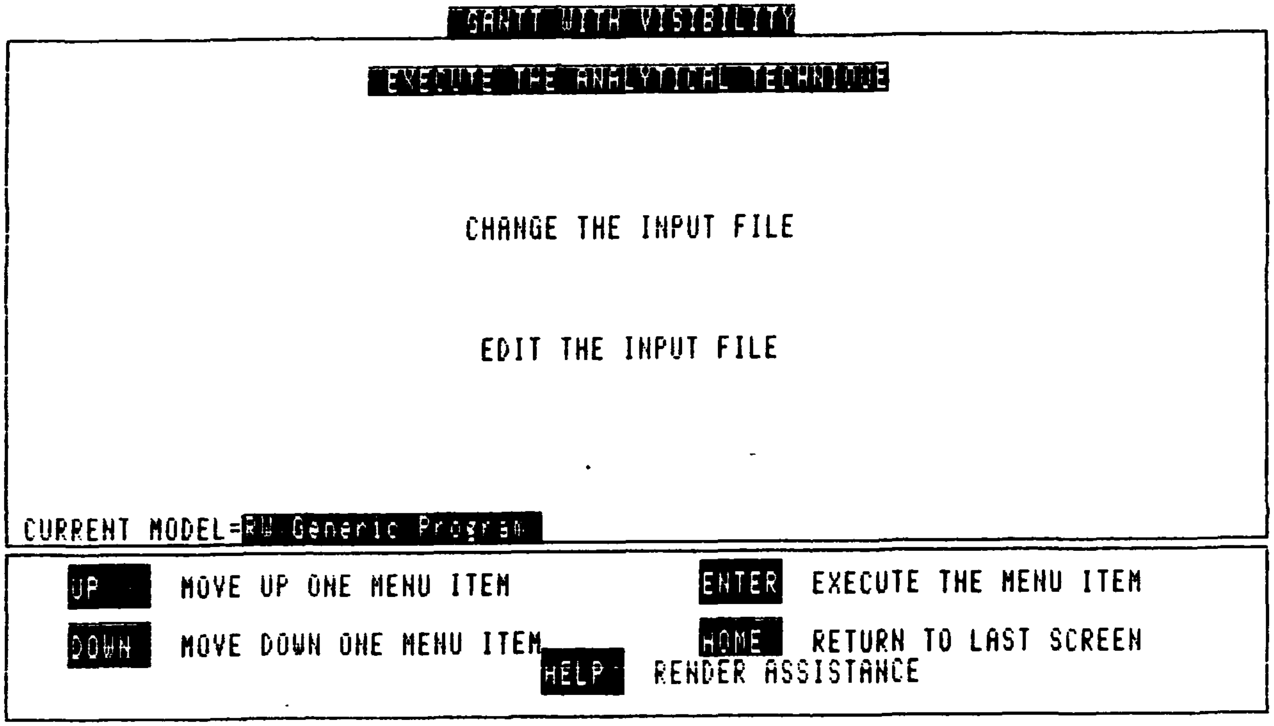
The Edit session is complete so the user depresses the HOME key.



```
RECORD # 30 OF 49 IS BEING LOADED
```

Fig 54. Input Stream Save Screen

The modified CURRENT MODEL is recorded onto the disk. A sentinel message is displayed for every 30 records to let the user know the machine is still working for him. A back-up file with a ".BAK" extension is created.



CURRENT MODEL=RW Generic Program

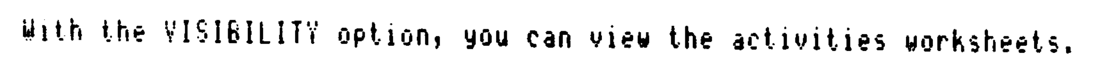
Fig 55. Second GANTT WITH VISIBILITY Option Screen

The user depresses ENTER to EXECUTE the GANTT WITH VISIBILITY analytic technique. The CURRENT MODEL is used as the input stream.

```
RW Program Manager's Generic Program          12 MONTHS FROM 01JAN85    PAGE 1
       EVENT       |J85|FEB|MAR |APR|MAY|JUN |JUL|AUG |SEP |OCT|NOV |DEC
DRAFT PMD          *******
COST SCHD BS WBS         ++++---------------
FINAL PMD                ****************************
THREAT ASSESSMNT         +++++++----------------------
ESTABLISH COMMUN         ++---------------------------
DEVELOP ILSP             +++++++++++++-----------------------------------
CRISP                    +++++++++++++++++++++++++++---------------------
NEW START REVIEW             ++---------------
IPR PREP                       ++++++++---------------
SECUR CLASS GUID               ++++++++++++++----------------------------
DD FORM 254                                   ++++-----------------------
AFSC FORM 56                                        **
PAD                                                 +++------------------
INIT PRGM REVIEW                                     *****
DRAFT SPEC                                                ***************
ACQ PLAN                                                  +++++++++++----
DEV POST IPR EST                                          +++++++--------
DRAFT SOW                                                 +++++++++++++++
DEV PRGM SCHEDUL                                          +++------------
DEVELOP TEMP                                              +++++++++++++++
Pause.
Please press <return> to continue.
```

Fig 57. GANTT Output by Weeks

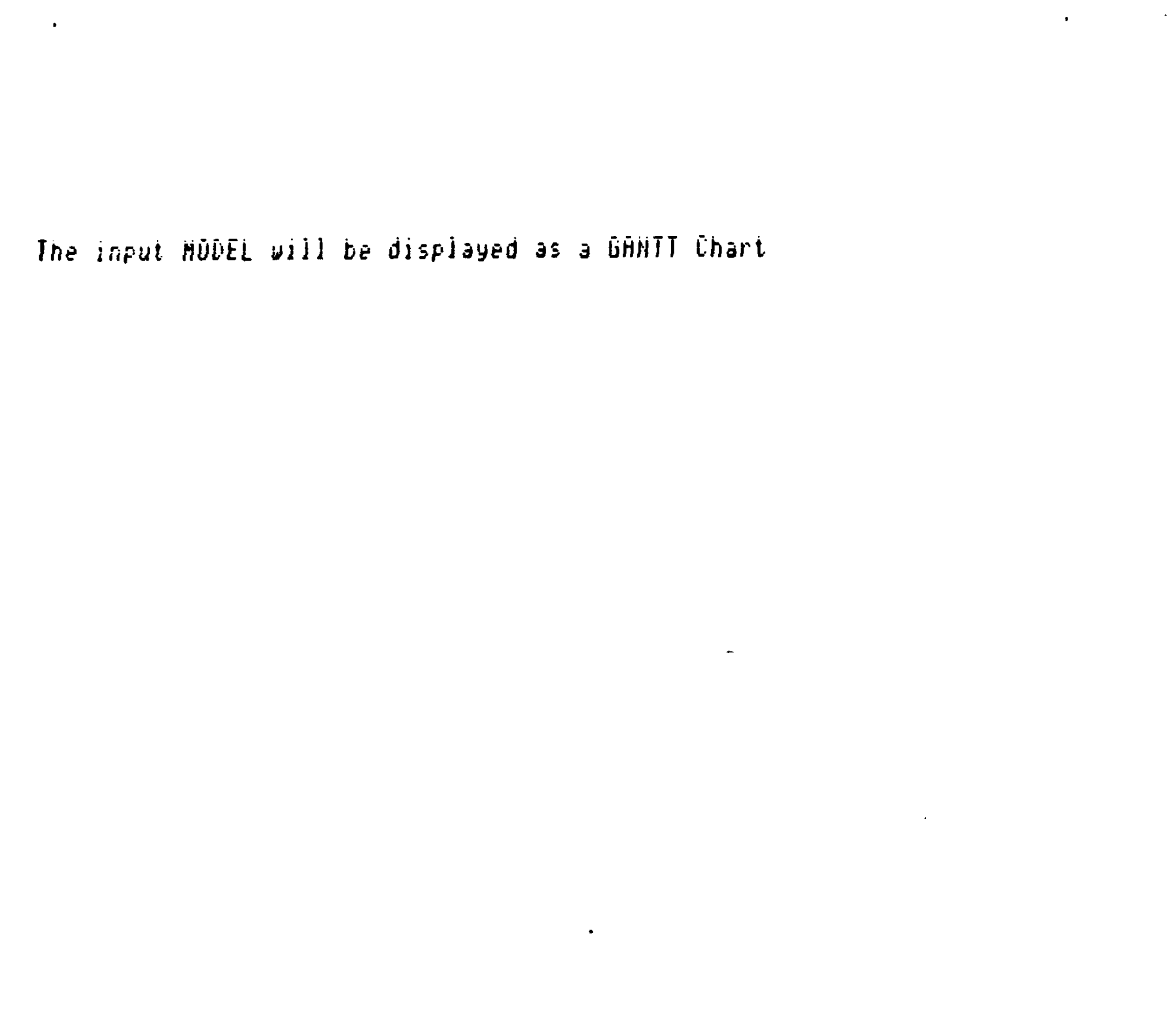
The current model is displayed as a GANTT chart. The "*" represent activities on the critical path. The "+" signs show the activity duration, and the "-" signs indicate the number of weeks of slack for the activity.



With the VISIBILITY option, you can view the activities worksheets.

Fig 58. Visibility Program Load Screen

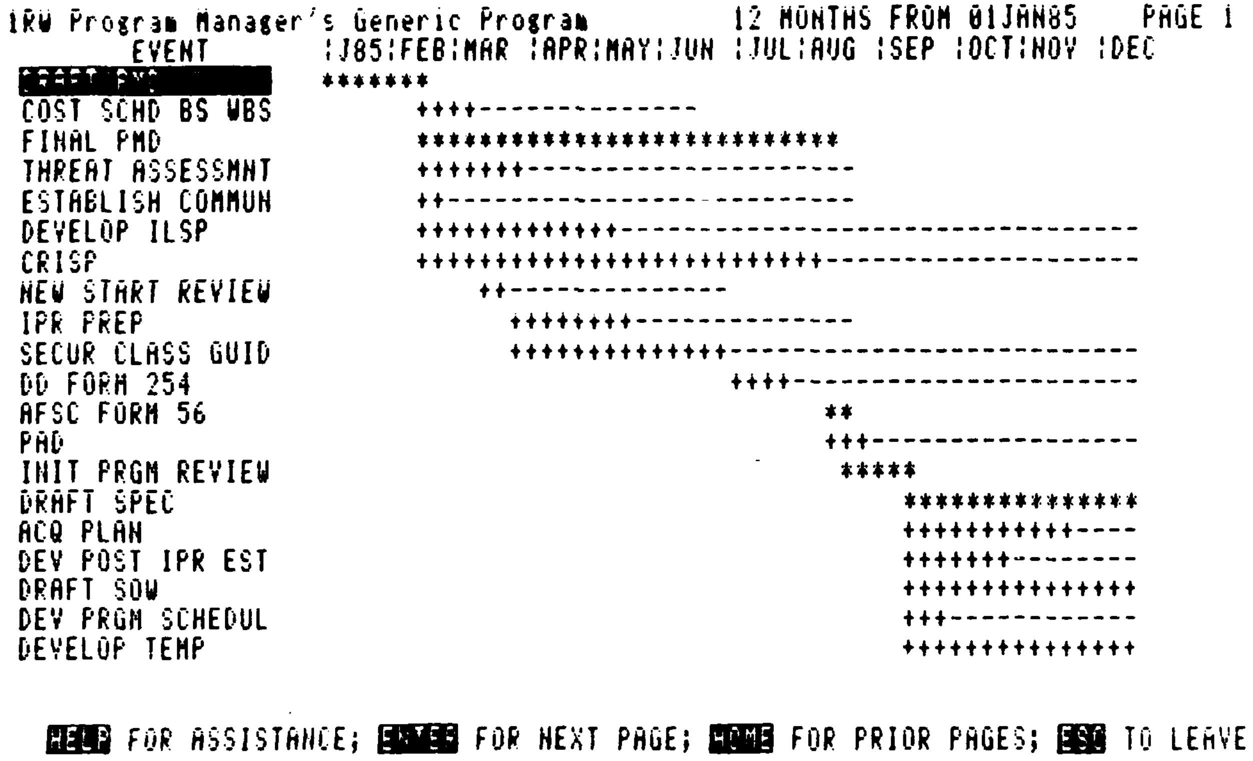
The Visibilty Program is loaded for the user.



The input MODEL will be displayed as a GANTT Chart

Fig 56. GANTT Model Load Screen

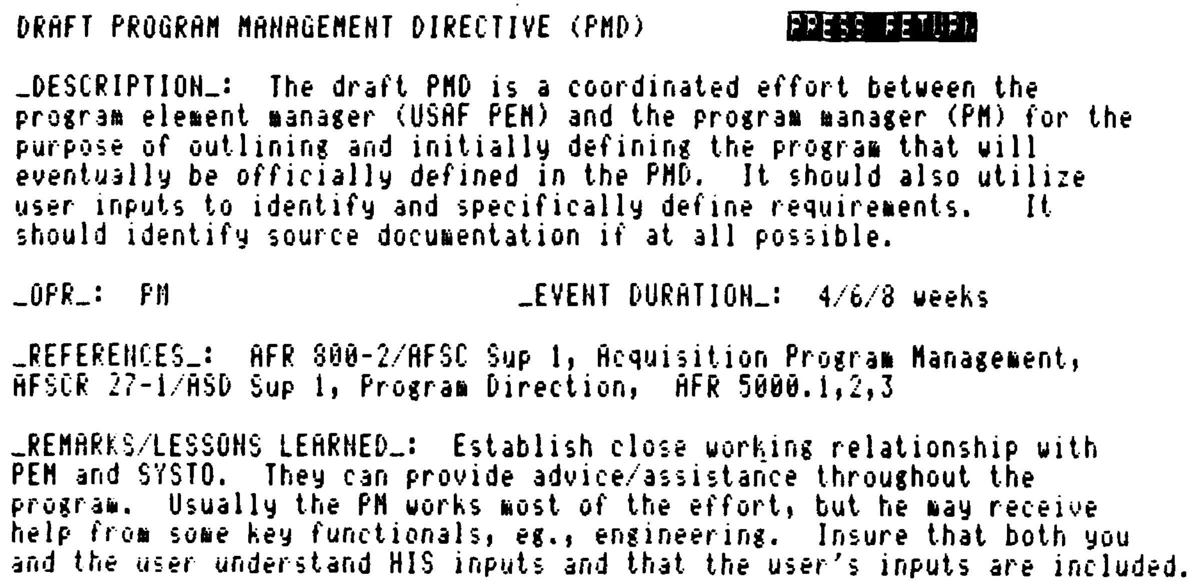
The user is informed that the GANTT program is being loaded.



```
1RW Program Manager's Generic Program         12 MONTHS FROM 01JAN85    PAGE 1
        EVENT       |J85|FEB|MAR |APR|MAY|JUN |JUL|AUG |SEP |OCT|NOV |DEC
 DRAFT PMD          *******
 COST SCHD BS WBS         ++++--------------
 FINAL PMD                ***************************
 THREAT ASSESSMNT         +++++++---------------------
 ESTABLISH COMMUN         ++--------------------------
 DEVELOP ILSP             +++++++++++++---------------------------------
 CRISP                    ++++++++++++++++++++++++++--------------------
 NEW START REVIEW             ++--------------
 IPR PREP                       ++++++++--------------
 SECUR CLASS GUID               ++++++++++++++--------------------------
 DD FORM 254                                  ++++----------------------
 AFSC FORM 56                                       **
 PAD                                                +++-----------------
 INIT PRGM REVIEW                                    *****
 DRAFT SPEC                                              ***************
 ACQ PLAN                                                +++++++++++----
 DEV POST IPR EST                                        +++++++--------
 DRAFT SOW                                               +++++++++++++++
 DEV PRGM SCHEDUL                                        +++------------
 DEVELOP TEMP                                            +++++++++++++++

   HELP FOR ASSISTANCE; ENTER FOR NEXT PAGE; HOME FOR PRIOR PAGES; ESC TO LEAVE
```

Fig 59. Visibility Screen

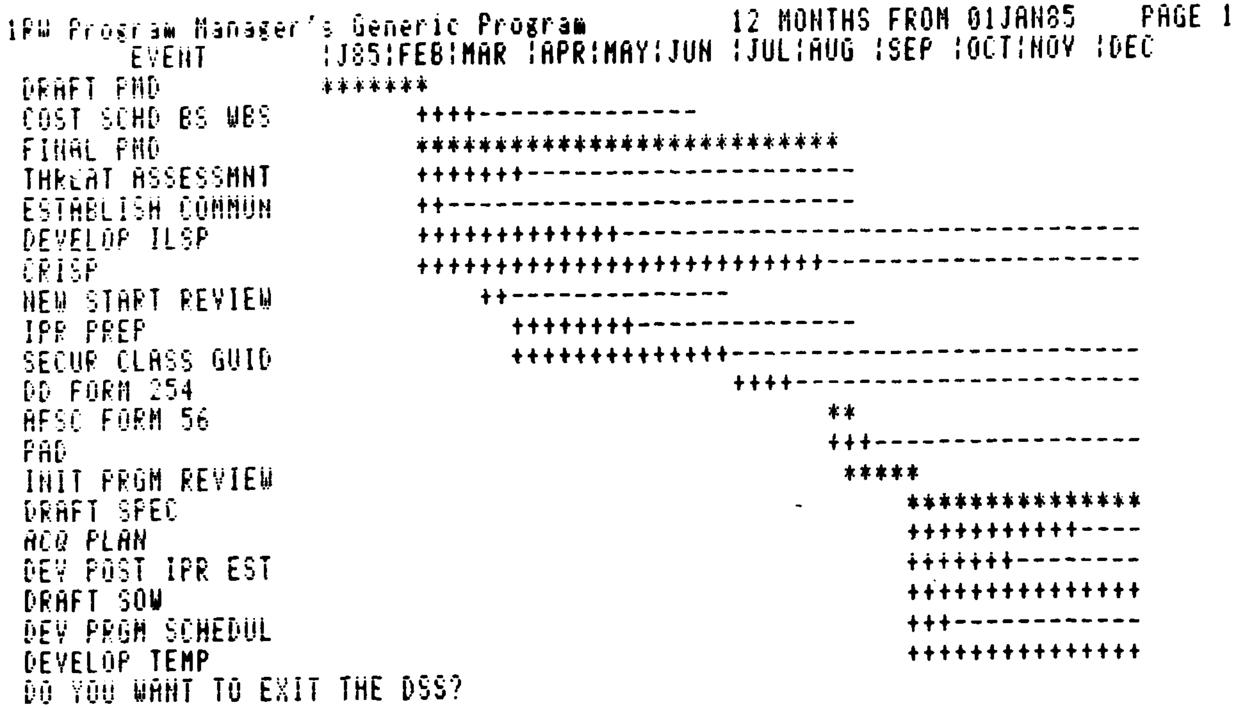
The GANTT chart is redisplayed. The user moves the cursor to the desired event by depressing the arrow keys. At this activity, the user depresses the HELP key to display the worksheet. It can explain the displayed activity duration.



```
DRAFT PROGRAM MANAGEMENT DIRECTIVE (PMD)          PRESS RETURN

_DESCRIPTION_:  The draft PMD is a coordinated effort between the
program element manager (USAF PEM) and the program manager (PM) for the
purpose of outlining and initially defining the program that will
eventually be officially defined in the PMD.  It should also utilize
user inputs to identify and specifically define requirements.   It
should identify source documentation if at all possible.

_OPR_:  PM                         _EVENT DURATION_:  4/6/8 weeks

_REFERENCES_:  AFR 800-2/AFSC Sup 1, Acquisition Program Management,
AFSCR 27-1/ASD Sup 1, Program Direction,  AFR 5000.1,2,3

_REMARKS/LESSONS LEARNED_:  Establish close working relationship with
PEM and SYSTO.  They can provide advice/assistance throughout the
program.  Usually the PM works most of the effort, but he may receive
help from some key functionals, eg., engineering.  Insure that both you
and the user understand HIS inputs and that the user's inputs are included.
```

Fig 60. Visibility Activity Worksheet Screen

The desired activity worksheet is displayed. If the Program Manager has been diligent in keeping the worksheets up-to-date, than a reason can be found for the observed activity duration.



```
1PW Program Manager's Generic Program         12 MONTHS FROM 01JAN85    PAGE 1
        EVENT       |J85|FEB|MAR |APR|MAY|JUN |JUL|AUG |SEP |OCT|NOV |DEC
 DRAFT PMD          *******
 COST SCHD BS WBS         ++++--------------
 FINAL PMD                ***************************
 THREAT ASSESSMNT         +++++++---------------------
 ESTABLISH COMMUN         ++--------------------------
 DEVELOP ILSP             +++++++++++++---------------------------------
 CRISP                    ++++++++++++++++++++++++++--------------------
 NEW START REVIEW             ++--------------
 IPR PREP                       ++++++++--------------
 SECUR CLASS GUID               ++++++++++++++--------------------------
 DD FORM 254                                  ++++----------------------
 AFSC FORM 56                                       **
 PAD                                                +++-----------------
 INIT PRGM REVIEW                                    *****
 DRAFT SPEC                                              ***************
 ACQ PLAN                                                +++++++++++----
 DEV POST IPR EST                                        +++++++--------
 DRAFT SOW                                               +++++++++++++++
 DEV PRGM SCHEDUL                                        +++------------
 DEVELOP TEMP                                            +++++++++++++++
 DO YOU WANT TO EXIT THE DSS?
```

Fig 61. Visibility Termination Screen

The user depressed ESC to leave the Visibility program. All the highlighted item revert to normal display and the EXIT question is asked. The terminal session is over so the user answers the question with a "y".

```
ECHO OFF
:START
   PROCESS
   SETASCII  Do you want to do more?
   IF ERRORLEVEL 89 GOTO START
REM    HE DID NOT ANSWER "Y"
OTHER PROCESS
```

The ASCII value for "y" is 89. If the user answered "y" to the question "Do you want to do more?". The ERRORLEVEL would equal "89", so the routine would jump to the :START label and PROCESS would re-execute.

Appendix C: DSS Evaluation Instrument

FEEDBACK SURVEY

The Program Manager's Decision Support System (PMDSS) has been demonstrated to you and we would like to obtain your initial impression. This system has been developed for the Program Manager in the field. Please complete this questionaire from that vantage point.

All responses are ANONYMOUS

For each response you are asked to pair your response, an adverbial qualifier, with one of the two descriptions of the quality. The following example is presented for clarification:

Example question.

Degree of system trai ing. The amount of training needed for the system relative to that amount given.

SUFFICIENT	__	__	__	__	__	__	__	INSUFFICIENT
	EXTREMELY	QUITE	SLIGHTLY	NEITHER	SLIGHTLY	QUITE	EXTREMELY	

If you thought that the system training was EXTREMELY SUFFICIENT, than the far left box should be checked.

ADVERBIAL KEY: |__|__|__|__|__|__|__|

EXTREMELY	QUITE	SLIGHTLY	NEITHER	SLIGHTLY	QUITE	EXTREMELY

1V. Probability that you would use the system.

HIGH |__|__|__|__|__|__|__| LOW

2V. Probability that other managers will use the system.

HIGH |__|__|__|__|__|__|__| LOW

3V. Probability that the system will be a success.

HIGH |__|__|__|__|__|__|__| LOW

4V. Managers evaluation of the worth of the system.

GOOD |__|__|__|__|__|__|__| BAD

5V. The level of timesaving you expect from the system.

SIGNIFICANT |__|__|__|__|__|__|__| INSIGNIFICANT

6V. The degree of training you would need before you could use the system.

LITTLE |__|__|__|__|__|__|__| MUCH

7V The extent the system will support your scheduling decisions.

HIGH |__|__|__|__|__|__|__| LOW

8E. Confidence in the system. The extent of your assurance or confidence in the system results.

HIGH |__|__|__|__|__|__|__| LOW
this item is IMPORTANT |__|__|__|__|__|__|__| UNIMPORTANT

9E. Relevancy. The degree of congruence between what you want from the system and what is provided.

USEFUL |__|__|__|__|__|__|__| USELESS
this item is IMPORTANT |__|__|__|__|__|__|__| UNIMPORTANT

10E. Job Effects. The changes in job freedom and job performance that are ascertained by you by using the PMDSS.

SIGNIFICANT |__|__|__|__|__|__|__| INSIGNIFICANT
this item is IMPORTANT |__|__|__|__|__|__|__| UNIMPORTANT

11E. User Friendly. The DSS allows novice users to operate effectively.

EASE SIMPLE TO USE								DIFFICULT HARD TO USE

Appendix D: PMDSS System BATCH Files

The PMDSS is highly dependent on the use of BATCH files. The system is entered using the automatic call that MS-DOS makes to the AUTOEXEC.BAT file. This BATCH file in turn transfers control to the PMDSS.BAT file. The PMDSS.BAT file loads the user interface routine which will create the BAT.BAT file on the "B" drive. The BAT.BAT file is a one line BATCH file identifying the desired Analytic Technique to be loaded and the specific input stream to use. The system BATCH files will be listed in alphabetic order.

AUTOEXEC is the name of the following BATCH file:

```
ECHO OFF
DATE
TIME
ECHO You may receive a printed copy of any of the screens at
ECHO  any time by depressing the SHIFT and F12 keys at the
ECHO                     same time.
psc
B:
CD USERDATA
A:
pause
CLS
TYPE PMDSS.TXT
pmdss
```

CREATE is the name of the following BATCH file:

```
echo off
CLS
ECHO You require two formatted EMPTY disks
echo Format them at 9 sector/track,
echo    and the PMDSS-SYS with a system
echo   FORMAT B:/9     and        FORMAT B:/S/9
ECHO
ECHO Place the PMDSS-SYS disk in drive A: (Top one)
echo
echo Place the disk formatted with the system into
echo           drive B: (The other one)
echo
setascii Are you ready???   HIT ANY KEY TO PROCEED
del b:*.*
copy *.* b:
echo Place the other EMPTY disk into drive B: (The lower one)
setascii Are you ready???   HIT ANY KEY TO PROCEED
b:
del *.*
mkdir userdata
cls
echo
echo Place the PMDSS-USR disk in drive A: (Top one)
ECHO
echo TYPE the following two lines:
echo
echo COPY A:*.*
ECHO COPY A:USERDATA\*.* USERDATA\*.*
```

ESCAP is the name of the following BATCH file:

```
CLS
TYPE ESCAP.TXT
A:
```

GAN is the name of the following BATCH file:

```
echo OFF
b:
CLS
echo The input MODEL will be displayed as a GANTT chart.
a:SORT /+17 <GANTT.INP >GANTT.SRT
a:GANTT
IF "%2" == "N" GOTO EXIT
rem see if the input for pert is wanted elsewhere
SET EXT=INP
set dir1=gvs
:start
A:SETASCII PRESS ANY KEY TO CONTINUE
  cls
  a:more < a:%dir1%.HLP
  a:setascii DO YOU WANT TO ADD THE CURRENT INPUT TO THIS AT?
  IF NOT ERRORLEVEL 89 GOTO 2
:Y
    REM THE ANSWER WAS "Y"
    IF NOT EXIST %DIR1%%1.INP COPY CHG%DIR1%%IN.MNU+RECORD
    SET NEW=%dir1%%1
    copy gantt.%EXT% %new%.inp
:2
if "%ext%"=="out" goto exit
rem see if the input for EVENT is wanted as another file
  set ext=out
  set dir1=vis
goto start
:exit
a:
pmdss
```

GVS is the name of the following BATCH file:

```
echo OFF
b:
CLS
echo The input MODEL will be displayed as a GANTT Chart
a:SORT /+17 <GANTT.INP >GANTT.SRT
a:GANTT
CLS
echo With the VISIBILITY option, you can view the
echo  activity worksheets.
COPY SCR%1.IDX EVENTS.INP
a:event
IF "%2" == "N" GOTO EXIT
A:SETASCII PRESS ANY KEY TO CONTINUE
rem see if the input for GANTT is wanted elsewhere
SET EXT=INP
set dirl=gan
:start
  cls
  a:more < a:%dirl%.HLP
  a:setascii DO YOU WANT TO ADD THE CURRENT INPUT TO THIS AT?
  IF NOT ERRORLEVEL 89 GOTO 2
:Y
    REM THE ANSWER WAS "Y"
    IF NOT EXIST %DIR1%%1.INP COPY CHG%DIR1%%IN.MNU+RECORD
    SET NEW=%dirl%%1
    COPY %NOW%.%EXT% %NEW%.inp
:2
if "%ext%"=="out" goto exit
rem see if the input for EVENT is wanted as another file
  set ext=out
  set dirl=vis
goto start
:exit
a:
pmdss
```

P2G is the name of the following BATCH file:

```
echo OFF
b:
CLS
COPY GANTT.OPT GANTT.INP
copy P2G%1.inp pertcp.inp
echo The PERTCPM Analytical Technique is being loaded
a:PERTCP
a:SORT <GANTT.INP >PERTCP.SRT
DEL GANTT.INP
a:CP2GNT
cls
echo The PERTCPM results have been sorted and will be
echo  displayed as a GANTT
a:SORT /+17 <GANTT.INP >GANTT.SRT
a:GANTT
IF "%2" == "N" GOTO EXIT
rem see if the input for pert is wanted elsewhere
A:SETASCII PRESS ANY KEY TO CONTINUE
set time=1
SET NOW=pertcp
set ext=inp
set dir1=per
set dir2=pgv
:start
  cls
  a:more < a:%dir1%.HLP
  a:setascii DO YOU WANT TO ADD THE CURRENT INPUT TO THIS AT?
  IF NOT ERRORLEVEL 89 GOTO SHIFT
:Y
    REM THE ANSWER WAS "Y"
    IF NOT EXIST %DIR1%%1.INP COPY CHG%DIR1%%IN.MNU+RECORD
    SET NEW=%dir1%%1
    IF NOT "%time%"=="2" COPY %NOW%.%ext% %NEW%.INP
    if "%time%"=="2" copy gantt.opt+GANTT.SRT %NEW%.INP
:SHIFT
  set dir1=%dir2%
  set dir2=""
  IF NOT %dir1%=="" GOTO START
if not "%time%"=="1" goto 2
rem see if the input for GANTT is wanted elsewhere
  set time=2
  set now=gantt
  set dir1=gan
  set dir2=gvs
goto start
:2
if "%ext%"=="out" goto exit
rem see if the input for EVENT is wanted as another file
```

```
  set ext=out
  set dirl=vis
  set time=3
goto start
:exit
a:
pmdss
```

PER is the name of the following BATCH file:

```
rem echo OFF
b:
CLS
ECHO The PERTCPM Analytical Technique is being loaded
COPY GANTT.OPT GANTT.INP
copy PER%1.inp pertcp.inp
a:PERTCP
IF "%2" == "N" GOTO PMDSS
A:SETASCII PRESS ANY KEY TO CONTINUE
set DIR1=p2g
set DIR2=pgv
:start
  cls
  a:more < a:%DIR1%.HLP
  a:setascii DO YOU WANT TO ADD THE CURRENT INPUT TO THIS AT?
  IF NOT ERRORLEVEL 89 GOTO SHIFT
:Y
    REM THE ANSWER WAS "Y"
    IF NOT EXIST %DIR1%%1.INP COPY CHG%DIR1%%IN.MNU+RECORD
    COPY pertcp.inp %dir1%%1.inp
:SHIFT
  SET DIR1=%DIR2%
  SET DIR2=""
  IF NOT %DIR1%=="" GOTO START
:PMDSS
a:
pmdss
```

PGV is the name of the following BATCH file:

```
echo OFF
b:
CLS
echo The PERTCPM Analytical Technique is being loaded
COPY GANTT.OPT GANTT.INP
copy PGV%1.inp pertcp.inp
a:PERTCP
a:SORT <GANTT.INP >PERTCP.SRT
DEL GANTT.INP
echo The results of the PERTCPM will be displayed as a GANTT
a:CP2GNT
a:SORT /+17 <GANTT.INP >GANTT.SRT
a:GANTT
CLS
echo With the VISIBILITY option you can see the
echo  activity worksheets
COPY SCR%1.IDX EVENTS.INP
a:event
IF "%2" == "N" GOTO EXIT
A:SETASCII PRESS ANY KEY TO CONTINUE
rem see if the input for pert is wanted elsewhere
set time=1
SET NOW=pertcp
set ext=inp
set dir1=per
set dir2=p2g
:start
  cls
  a:more < a:%dir1%.HLP
  a:setascii DO YOU WANT TO ADD THE CURRENT INPUT TO THIS AT?
  IF NOT ERRORLEVEL 89 GOTO SHIFT
:Y
    REM THE ANSWER WAS "Y"
    IF NOT EXIST %DIR1%%1.INP COPY CHG%DIR1%%IN.MNU+RECORD
    SET NEW=%dir1%%1
    IF NOT "%time%"=="2" COPY %NOW%.%ext% %NEW%.%ext%
    if "%time%"=="2" copy gantt.opt+gantt.inp %new%.inp
:SHIFT
  set dir1=%dir2%
  set dir2=""
  IF NOT %dir1%=="" GOTO START
if not "%time%"=="1" goto 2
rem see if the input for GANTT is wanted elsewhere
  set time=2
  set now=gantt
  set dir1=gan
  set dir2=gvs
goto start
```

```
:2
if "%ext%"=="out" goto exit
rem see if the input for EVENT is wanted as another file
  set ext=out
  set time=3
  set dirl=vis
goto start
:exit
a:
pmdss
```

PMDSS is the name of the following BATCH file:

```
echo off
a:
setascii DO YOU WANT TO EXIT THE DSS?
IF ERRORLEVEL 89  GOTO exit
CLS
ECHO THE DSS MODEL AND OPTION ROUTINE IS BEING LOADED!
COPY ESCAP.BAT B:BAT.BAT
BLDMNU
b:BAT
cls
:exit
EXIT
```

VIS is the name of the following BATCH file:

```
echo OFF
b:
CLS
echo With the VISIBILITY option you can view the
echo  activity worksheets.
copy VISN1.inp gantt.out
COPY SCRN1.IDX EVENTS.INP
a:event
CLS
:exit
a:
pmdss
```

Appendix E: PMDSS ZBASIC and FORTRAN Source Code

The PMDSS Analytic Techniques are compiled FORTRAN and ZBASIC program. The BLDMNU program is the User Interface Routine. Other routines are called dependent on the user interaction. The source code for the ZBASIC and FORTRAN programs appear in alphabetic order. The ZBASIC routines are all listed before the FORTRAN programs.

```
10 ON ERROR GOTO 6740
20 DIM LINS$(100),ITEMFL$(100),LEVELS$(10),MERGLN$(5) ,LNHLP$(4)
30 DIM CRDTYP( 4), FILDES(4,10,4 ) , CRDNUM ( 4 ), CRDFLD(4)
40 DIM KEYS$(12),XPOS(12),YPOS(12), EXPLAN$(12)
50 CLS
60 I=0
70 FOR J=1 TO 10:READ TITLES$
80 DATA "Program Managers","DECISION SUPPORT SYSTEM","By"
85 DATA "Captain Terrence Brotherton, U.S.A.F"
90 DATA "In Partial Fulfillment of the ","Requirements for the Degree of"
100 DATA "Master of Science in Systems Management"
110 DATA "of the School of Systems and Logistics"
120 DATA "of the Air Force Institude of Technology","Air University"
130 IF J>4 THEN I=1
140 LOCATE (J-1)*2+1+I*4,40-LEN(TITLES$)/2 :PRINT TITLES$
150 NEXT J
160 COLOR 0,7:LOCATE 1,65:PRINT "TYPE ANY KEY":COLOR 7,0
170 A$=INKEY$ : IF A$ = "" THEN 170
180 SCRNFL$ = "ATSLCT":TITLES$="Analytical Techniques"
190 LEVELS$(1)="ATSLCT  "+TITLES$:LEVEL=1
200 UPAROW$ = CHR$(30):MNSIGN$ = CHR$(45)
210 DNAROW$ = CHR$(31):PLSIGN$ = CHR$(43)
220 LFAROW$ = CHR$(29):LTSIGN$ = CHR$(60)
230 RTAROW$ = CHR$(28):GTSIGN$ = CHR$(62)
240 LINFED$ = CHR$(10) : DELET$ = CHR$(127)
250 QUOTE$=CHR$(34)
260 ADISK$="A:" : BDISK$="B:"
270 CR$ = CHR$(13)
280 HELP$=CHR$(1) :MLTSGN$=CHR$(42)
290 ESC$ = CHR$(3): CNTLC$= CHR$(27)
300 HOME$=CHR$(11):DVSIGN$=CHR$(47)
310 ITEM = 1:LSTITM=1:INFIL$="PERRELRW":MODELN$="RW Generic Program "
320 SCRNLN = 14
330 TRUE = -1
340 HILIT = -1                                'WHEN HILIT = TRUE THEN HIGHLIGHT
350 GOSUB 600                                ' READ THE SCREEN FILE
360 REM                                       ' DISPLAY THE HELP KEYS
370 GOSUB 810                                ' DISPLAY THE 1ST SCREEN
380 HILIT = 0:ISAV=ITEM:ITEM=LSTITM:GOSUB 940:HILIT = -1:ITEM=ISAV 'MENU BLINK
390 SCRNFL$=ITEMFL$(ITEM)
400 GOSUB 940                                ' HIGHLIGHT THE MENU ITEM
410 REM
420 REM       RECIEVE THE USERS KEY INPUT
430 REM
440 LSTITM = ITEM
450 A$ = INKEY$:IF A$="" THEN 450          ' CHECK FOR TERMINAL INPUT
460 IF A$=UPAROW$ OR A$=MNSIGN$ THEN ITEM = ITEM - 1:GOTO 550 ELSE  'MOVE UP?
470 IF A$=DNAROW$ OR A$=PLSIGN$ THEN ITEM = ITEM + 1:GOTO 550 ELSE  'MOVE DOWN?
```

```
480 IF A$=ESC$ OR A$=CNTLC$  THEN 6690 ELSE       ' LEAVE THE DSS?
490 IF A$ = HOMES$ OR A$ = DVSIGN$ THEN GOSUB 1260 ELSE    ' go back one screen
500 IF A$=HELP$ OR A$ = MLTSGN$ THEN  GOSUB 6490    ELSE            'WANT HELP?
510 IF A$=CR$ THEN GOSUB 1360 ELSE 380
520 REM
530 REM          SEE IF THE MOVED CURSOR IS ON THE CURRENT SCREEN
540 REM
550 IF ITEM > MAXLIN THEN ISTLIN = SCRNLN+ISTLIN:IF ITEM > ILINE THEN ISTLIN = 1
:ITEM = ISTLIN:IF ILINE>SCRNLN THEN GOSUB 790           'MOVE BACK A SCREEN
560 IF ITEM < ISTLIN THEN ISTLIN = SCRNLN*FIX(ISTLIN-1/SCRNLN):IF ISTLIN<1 THEN
ISTLIN = 1:ITEM = ISTLIN:IF ILINE>SCRNLN THEN GOSUB 790      'MOVE UP A SCREEN
570 IF LSTITM<ISTLIN OR LSTITM>MAXLIN THEN LSTITM = ITEM:GOTO 450
580 A$=""
590 GOTO 380
600 REM
610 REM        THIS ROUTINE READS THE SCREEN FILE, AND BUILDS A SCREEN
620 REM
630 IF LEVEL > 3 THEN DSK$=BDISK$ ELSE DSK$=ADISK$
640 IF LEFT$(SCRNFL$,3)  = "chg" THEN DSK$=BDISK$
650 OPEN "I",#1,DSK$+SCRNFL$+".MNU"
660 ILINE = 0
670    IF EOF(1) THEN 750
680    ILINE = ILINE + 1
690    LINE INPUT #1, LIN$(ILINE)
700    LNGTH = 70
710    ITEMFL$(ILINE)=MID$(LIN$(ILINE),1,8):LIN$(ILINE)=MID$(LIN$(ILINE),9,70)
720    IF(RIGHT$(LIN$(ILINE),1) = " ") THEN LNGTH=LNGTH-1:LIN$(ILINE)=MID$(LIN$(
ILINE),1,LNGTH):GOTO 720
730 GOTO 670
740 REM
750 REM    ERASE THE OPERATIVE PART OF THE LAST SCREEN
760 REM
770 CLOSE #1
780 ISTLIN = 1
790 CLS
800 GOSUB 1020
810 REM
820 REM      WRITE THE CURRENT SCREEN
830 REM
840 MAXLIN = ISTLIN + SCRNLN-1:IF MAXLIN > ILINE THEN MAXLIN = ILINE
850 INCREM = SCRNLN / (MAXLIN-ISTLIN+1)
860 START = 3               '    SHOULD CENTER IT
870 FOR I = ISTLIN TO MAXLIN
880    LOCATE START+(I-1)*INCREM,40-LEN(LIN$(I))/2
890    PRINT LIN$(I)
900 NEXT I
910 IF LEVEL > 1 THEN LOCATE 18,2:PRINT "CURRENT MODEL=";:COLOR 0,7:PRINT MODELN
$:COLOR 7,0
920 IF LEVEL=2 THEN INFIL$=LEFT$(LEVEL$(2),3)+MID$(INFIL$,4,LEN(INFIL$)-3)
930 RETURN
```

```
940 REM
950 REM          HIGHLIGHT THE CURRENT MENU ITEM
960 REM
970 IF HILIT = TRUE THEN COLOR 0,7 ELSE COLOR 7,0
980 LOCATE START+(ITEM-1)*INCREM,40-LEN(LIN$(ITEM))/2
990 PRINT LIN$(ITEM)
1000 IF HILIT = TRUE THEN COLOR 7,0 ELSE COLOR 0,7
1010 RETURN
1020 REM
1030 REM    THIS ROUTINE BUILDS THE HELP KEY SCREEN
1040 REM
1050 KEYS$(1)="UP":KEYS$(2)="DOWN":KEYS$(3)="ENTER":KEYS$(4)="HOME"
1060 KEYS$(5) = "HELP"
1070 XPOS(1)=5: XPOS(2)=5: XPOS(3)=45:XPOS(4)=45:XPOS(5)= 35
1080 YPOS(1)=20:YPOS(2)=22:YPOS(3)=20:YPOS(4)=22:YPOS(5)=23
1090 EXPLAN$(1) = "MOVE UP ONE MENU ITEM"
1100 EXPLAN$(2) = "MOVE DOWN ONE MENU ITEM"
1110 EXPLAN$(3) = "EXECUTE THE MENU ITEM"
1120 EXPLAN$(4) = "RETURN TO LAST SCREEN"
1130 EXPLAN$(5) = "RENDER ASSISTANCE"
1140 CLS
1150 A=FRE(" ")
1160 LOCATE 1,40-LEN(TITLE$)/2:COLOR 0,7:PRINT TITLE$:COLOR 7,0
1170 LINE (0,9)-(639,162),7,B                              'DRAW TOP BOX
1180 LINE (0,165)-(639,215),7,B                             'DRAW HELP BOX
1190 FOR KEYS=1 TO 5
1200    LINE ((XPOS(KEYS)-1)*8,YPOS(KEYS)*9-12)-((XPOS(KEYS)+4)*8+1,YPOS(KEYS)*9
+1),7,BF
1210    LOCATE YPOS(KEYS),XPOS(KEYS)
1220    COLOR 0,7:PRINT KEYS$(KEYS):COLOR 7,0
1230    LOCATE YPOS(KEYS),XPOS(KEYS)+7:PRINT EXPLAN$(KEYS)
1240 NEXT KEYS
1250 RETURN
1260 REM
1270 REM         BACK-UP ONE MENU FOR THE USER
1280 REM
1290 IF LEVEL = 1 THEN RETURN
1300 LEVEL = LEVEL - 1
1310 SCRNFL$ = MID$(LEVEL$(LEVEL),1,8)
1320 TITLE$=MID$(LEVEL$(LEVEL), 9, LEN(LEVEL$(LEVEL))-8 )
1325 LSTITM = 1
1330 ITEM = 1
1340 GOSUB 600
1350 RETURN
```

```
1360 REM
1370 REM     THE EXECUTION ROUTINE, FIRST READ THE NEW MENU FILE
1380 REM       DISPLAY IT, SEE IF THE USER WANTS TO EXECUTE THE DEFAULT
1390 REM
1400 LEVEL = LEVEL + 1
1410 ON LEVEL-1 GOTO 1420,1480,1860,3230,1830          'HOW MANY LEVELS DEEP
1420 TITLE$ = LIN$(ITEM):LEVEL$(LEVEL)=ITEMFL$(ITEM)+TITLE$
1430 SCRNFL$ = ITEMFL$(ITEM)
1440 ITEM = 1
1445 LSTITM = 1
1450 GOSUB 600
1460 RETURN
1470 REM
1480 ON ITEM GOTO 1520,1740,1800                   'WHAT FILE ACTION(USE,CHG,CREATE)
1490 REM
1500 REM         THIS IS A PATCH.... FOR gantt DERIVED RUNS, THEY NEED
1510 REM                             TWO FILES...GANTT.OPT & GANTT.INP
1520 REM
1530 IF LEFT$(INFIL$,1) <>"g" THEN 1600
1540 OPEN "I",#1,"B:"+INFIL$+".INP"
1550 OPEN "O",#2,"B:GANTT.OPT"
1560 LINE INPUT #1,LN$:PRINT #2,LN$:CLOSE #2:OPEN "O",#2,"B:GANTT.INP"
1570 IF EOF(1) THEN 1590
1580    LINE INPUT #1,LN$:PRINT #2,LN$:GOTO 1570
1590 CLOSE
1600 REM               GOING TO CREATE THE .BAT FILE
1610 REM
1620 CLS:LOCATE 13,20:PRINT "DO YOU WANT THE OPTION OF ADDING THIS MODEL"
      :  LOCATE 14,22 :PRINT "TO OTHER 'AT'S (<Y>es or any other key)";
1630 A$=INKEY$:IF A$="" THEN 1630
1640 IF A$="Y" OR A$="y" THEN OPT$="Y" ELSE OPT$="N"
1660 OPEN "O",#1,"B:BAT.BAT"
1670 PRINT #1,MID$(LEVEL$(2),1,8)+" "+MID$(INFIL$,4,5)+" "+OPT$
1680 PRINT #1,"EXIT"
1690 CLOSE#1
1700 OPEN "O",#1,"B:RECORD"
1710 PRINT #1,INFIL$+MODELN$
1720 CLOSE #1
1730 END
1740 REM
1750 REM               CHANGE INPUT FILE
1760 REM
1770 CHOSFL =  1                         'FLAG THE FACT THAT JUST CHOOSING A FILE
1780 GOSUB 1420                           'DO THE SAME PROCESS AS ANOTHER MENU FILE
1790 RETURN
```

```
1800 REM
1810 REM          CREATE/MODIFY AN EXISTING INPUT FILE
1820 REM
1830 CHOSFL = 2                        ' FLAG THAT CREATING NEW FILE
1840 GOSUB 1420
1850 RETURN
1860 REM
1870 REM          LEVEL = 3, A FILE HAS BEEN CHOSEN
1880 REM
1890 ON CHOSFL GOTO 1900,1950
1900 INFIL$ = ITEMFL$(ITEM):MODELN$=LIN$(ITEM)
1910 LEVEL = 2: ITEM = 1
1920 TITLE$=MID$(LEVEL$(2),9,LEN(LEVEL$(2))-8):SCRNFL$=MID$(LEVEL$(2),1,8)
1930 GOSUB 600
1940 RETURN
1950 REM
1960 REM                IN THE EDIT MODE
1970 REM
1980 KNTSAV = SCRNLN
1990 NOWAT$=LEFT$(SCRNFL$,3)
2000 DGT = 5
2010 FOR I = 1 TO 5
2020    IF MID$(INFIL$,3+I,1) = " " THEN DGT=I-1:GOTO 2040
2030 NEXT I
2040 NOWMDL$=MID$(INFIL$,4,DGT)+STRING$(5-DGT,"_")
2050 FILEIN$ = "CHG"+NOWAT$+"IN"
2060 IF NOWAT$ = "vis" THEN IF ITEM = 5 THEN 2360 ELSE 6220 'ONLY ALLOWED TO DEL
2070 REM
2080 REM          IS THE ACTION A merge FROM ANOTHER FILE
2090 REM
2100 IF ITEM <> 3 THEN 2350
2110     CLS
2120     LOCATE 1,35:COLOR 0,7:PRINT "CURRENT FILES ARE:":LOCATE 2,2:
         PRINT " FILE": LOCATE 2,20: PRINT "DESCRIPTION":COLOR 7,0
2130     CLOSE:OPEN "I",#1,"B:"+FILEIN$+".MNU"
2140     I = 4
2150     IF EOF(1) THEN ITEM = 2 :GOTO 2310
2160        LINE INPUT #1,LN$
2170      IF LEFT$(LN$,3) = INFIL$ THEN 2150
2180        I = I + 1:IF I > 22 THEN I=22
2190        LOCATE I,1 : PRINT LEFT$(LN$,3)+" "+RIGHT$(LN$,LEN(LN$)-8);
2200        LOCATE 23,20:PRINT SPACE$(58);:LOCATE 23,20:PRINT "DEPRESS ";:
            COLOR 0,7:PRINT "LINE FEED";:COLOR 7,0:  PRINT " TO USE THIS FILE"
2210        A$=INKEY$: IF A$ = "" THEN 2210
2220        IF A$ <> LINFED$ THEN 2150
2230   MERGFL$ = LEFT$(LN$,3)
2240   OPEN "I",#4,"B:"+MERGFL$+".INP"
2250   MRGREC = 1
2260   FOR I = 1 TO 5
2270      IF EOF(4) THEN CLOSE #4:MERGMX=-MERGMX:GOTO 2300
2280      LINE INPUT #4, MERGLN$(I) : MERGMX = I
2290   NEXT I
```

```
2300   KNTSAV = SCRNLN+2: SCRNLN = SCRNLN - 6
2310   CLOSE #1
2320 REM
2330 REM          SEE IF NEED TO DELETE A FILE
2340 REM
2350 IF ITEM <> 5 THEN 2570
2360     CLS
2370     LOCATE 1,35:COLOR 0,7:PRINT "CURRENT FILES ARE:":LOCATE 2,2:
         PRINT " FILE": LOCATE 2,20: PRINT "DESCRIPTION":COLOR 7,0
2380     OPEN "O",#1,"B:"+FILEINS+".BAK":CLOSE #1:KILL "B:"+FILEINS+".BAK" :
                    NAME "B:"+FILEINS+".MNU" AS "B:"+FILEINS+".BAK"
2390     CLOSE:OPEN "I",#1,"B:"+FILEINS+".BAK":OPEN "O",#2,"B:"+FILEINS+".MNU"
2400     I = 4
2410     IF EOF(1) THEN 2550
2420        LINE INPUT #1,LNS
2430        IF LEFT$(LN$,8) = INFILS THEN 2410   'DON'T DELETE THE CURRENT MODEL
2440        I = I + 1:IF I > 22 THEN I=22
2450        LOCATE I,1 : PRINT LEFT$(LNS,8)+" "+RIGHT$(LNS,LEN(LNS)-8);
2460        LOCATE 23,20:PRINT "DEPRESS ";:COLOR 0,7:PRINT "DELETE";:COLOR 7,0:
              PRINT " TO DELETE THIS FILE"
2470        AS=INKEY$: IF AS = "" THEN 2470
2480        IF AS <>DELETS THEN PRINT #2,LNS:GOTO 2410
2490 KILFLS=LEFT$(LNS,8)
2500 FOR J = 1 TO 8
2510     IF MID$(LN$,J,1)=" " THEN KILFLS=LEFT$(LN$,J-1):GOTO 2530
2520 NEXT J
2530         KILL "B:"+KILFLS+".INP"
2540         GOTO 2410
2550     CLOSE #1
2560 GOTO 6230
2570 REM          CREATE A NEW FILE, SEE IF SIMPLE CHANGE,MERGE,FROM SCRATCH
2580 REM
2590 COMNTS = -1
2600 REM IF ONLY WANT TO UPDATE THE COMMENT SCREENS DURING PERTCP RUNS USE NEXT
LINE
2610 'IF NOWAT$ = "per" OR NOWATS = "p2g" OR NOWAT$ = "pgv" THEN LOCATE CSRLIN,5
      :PRINT "DO YOU WANT TO <U>date, <V>iew or <N>ot see the COMMENT SCREENS";
2620 LOCATE CSRLIN,5         :
  PRINT "DO YOU WANT TO <U>pdate, <V>iew or <N>ot see the ACTIVITY WORK-SHEETS";
2630 AS=INKEY$:IF AS<>"U" AND AS<> "V" AND AS<> "N"
             AND AS<>"u" AND AS<> "v" AND AS<> "n" THEN 2630
2640 IF AS="N" OR AS="n" THEN COMNTS=-1 ELSE IF AS="V" OR AS="v" THEN COMNTS = 0
       ELSE COMNTS = 1
2650     OPEN "I",#1,"B:SCR"+NOWMDLS+".IDX":EVNTS=0
2660     IF EOF(1) THEN 2710
2670        INPUT #1,EVENTS,SCRENS
2680        EVNTS = EVNTS + 1
2690        LINS(EVNTS) = SCRENS+SPACES(8-LEN(SCRENS) )+EVENTS
2700        GOTO 2660
2710     CLOSE #1
```

```
2720 REM                FIRST GET A DISCRIPTION OF THE FILE
2730 REM
2740 IFLD=2:IF LEFT$(NOWAT$,1) = "g" THEN IFLD=1
2750 SCRNFL$=LEFT$(SCRNFL$,3)+"MODRC"
2760 OPEN "I",#1,"A:"+LEFT$(SCRNFL$,1)+".FLD"
2770 INPUT #1, NUMCRD                         ' THE NUMBER OF DISTINCT DATA CARDS
2780 FOR CRDTYP = 1 TO NUMCRD
2790     INPUT #1,CRDFLD(CRDTYP), CRDNUM(CRDTYP)       'NUMBER OF FIELDS & CARDS
2800     FOR FLD = 1 TO CRDFLD(CRDTYP)
2810        FOR NUMFLD = 1 TO 4
2820            INPUT #1, FILDES(NUMFLD,FLD,CRDTYP) '1=START POS, 2=LENGTH
2830        NEXT NUMFLD                             '3=NUM MIN,4=MAX (0s=ALPHA)
2840     NEXT FLD
2850 NEXT CRDTYP
2860 CLOSE #1
2870 REM
2880 OPEN "R",#3,"A:"+LEFT$(NOWAT$,1)+"FLD.HLP",256
2890 FIELD #3, 64 AS LNHLP$(1),64 AS LNHLP$(2),64 AS LNHLP$(3),64 AS LNHLP$(4)
2900 OPEN "R",#2,"B:WORKING.FIL",80
2910 FIELD #2, 80 AS DATALN$
2920 REM
2930 REM    CREATE A NEW FILE
2940 REM
2950 IF ITEM <> 4 THEN 3140       ' CREATE A NEW FILE FROM SCRATCH
2960   MODELN$ = "USE THE 'LINE FEED' TO INSERT NEW LINES"
2970   FOR CRDTYP = 1 TO NUMCRD
2980    MXREC=CRDNUM(CRDTYP+1):IF MXREC<CRDNUM(CRDTYP) THEN MXREC=CRDNUM(CRDTYP)
2990      LN$ = SPACE$(80)
3000      FOR I = 1 TO CRDFLD(CRDTYP)
3010     NOWLNG =FILDES(2,I,CRDTYP):NW$=RIGHT$(STR$(FILDES(3,I,CRDTYP)),NOWLNG)
3020     NW$=SPACE$(NOWLNG-LEN(NW$))+NW$
3030        IF(FILDES(4,I,CRDTYP))=0 THEN NW$="|"+STRING$(NOWLNG-1,"_")
3040        LN$=LEFT$(LN$,FILDES(1,I,CRDTYP)-1)+NW$
3050       NEXT I
3060    FOR CRDS = CRDNUM(CRDTYP) TO MXREC
3070      LSET DATALN$ = LN$
3080      PUT #2, CRDS
3090    NEXT CRDS
3100  NEXT CRDTYP
3110 EVNTS = 0
3120 MAXFIL = CRDNUM(NUMCRD)
3130 GOTO 3240
3140 MAXFIL = 0
3150 OPEN "I",#1,"B:"+INFIL$+".INP"
3160 IF EOF(1) THEN 3230
```

```
3170     LINE INPUT #1,LN$
3180     MAXFIL = MAXFIL + 1
3190     IF MAXFIL MOD 30 = 0 THEN CLS:LOCATE 13,31:COLOR 0,7:
           PRINT "RECORD #";MAXFIL;" IS BEING LOADED";:COLOR 7,0
3200     LSET DATALN$ = LN$
3210     PUT #2,MAXFIL
3220 GOTO 3160
3230 CLOSE #1
3240 XOFF=1 : YOFF=1
3250 REM
3260 REM                THE FILE IS INSIDE THE RANDOM WORKING FILE
3270 REM
3280 FRSTLN = 1 : NOWLN=1 : NOWFLD=1  : NOWTYP = 1:NOWLN = 1
3290 NOWXPS = FILDES(1,NOWFLD,NOWTYP):NOWLNG=FILDES(2,NOWFLD,NOWTYP)
3300 LNSAV=NOWLN:XSAV=NOWXPS:STRSAV$=""
3310 NOWXPS = FILDES(1,NOWFLD,NOWTYP):NOWLNG=FILDES(2,NOWFLD,NOWTYP)
3320 GOSUB 5120                     'DRAW THE EDIT SCREEN
3330 SCRNFL$=LEFT$(SCRNFL$,3)+"MODRC"
3340 LSTREC=FRSTLN+SCRNLN-1:IF LSTREC > MAXFIL THEN LSTREC = MAXFIL
3350 XSHOW = XOFF + 1
3360 FOR IRECRD=FRSTLN TO LSTREC
3370      GET #2,IRECRD
3380      LOCATE IRECRD-FRSTLN+YOFF+1,XSHOW
3390      LN$=LEFT$(DATALN$,77) : PRINT LN$        ' SHOW THE CURRENT LINE
3400 NEXT IRECRD
3410 NOWXPS = FILDES(1,NOWFLD,NOWTYP):NOWLNG=FILDES(2,NOWFLD,NOWTYP)
3420 GET #2,NOWLN+FRSTLN-1
3430 NOWFLD$=MID$(DATALN$,NOWXPS,NOWLNG)         'CURRENT FIELD
3440 GOSUB 6240
3450 A$ = INKEY$:IF A$="" THEN 3450          ' CHECK FOR TERMINAL INPUT
3460 MOVE = 0 : MVFLD = 0: MVSCRN = 0
3470 IF A$=DNAROW$ THEN MOVE =  1:GOTO 4290 ELSE  'MOVE DOWN?
3480 IF A$=UPAROW$ THEN MOVE = -1:GOTO 4290 ELSE  'MOVE UP?
3490 IF A$=RTAROW$ THEN MVFLD = 1 :GOTO 4290 ELSE  'MOVE RIGHT?
3500 IF A$=LFAROW$ THEN MVFLD= -1:GOTO 4290 ELSE  'MOVE LEFT?
3510 IF A$=MNSIGN$ THEN MVSCRN=-1 :STRSAV$="":GOTO 4290 ELSE  'BACK A SCREEN?
3520 IF A$=PLSIGN$ THEN MVSCRN=1 :STRSAV$="":GOTO 4290 ELSE  'FORWARD A SCREEN?
3530 IF A$=LINFED$ THEN 4420 ELSE                   'INSERT A LINE?
3540 IF A$=DELET$ THEN 5020 ELSE                   'DELETE A LINE?
3550 IF A$=ESC$ OR A$=CNTLC$  THEN 6690 ELSE        ' LEAVE THE DSS?
3560 IF A$ = HOME$ OR A$ = DVSIGN$ THEN  5580 ELSE     ' go back one screen
3570 IF A$=HELP$ OR A$ = MLTSGN$ THEN  GOSUB 6310    ELSE             'WANT HELP?
3580 IF A$=CR$ THEN A$=" ":GOTO 3610
3590 IF A$<" " OR A$>"~" THEN 3440
3600 IF A$=HELP$ THEN 3440
```

```
3610 REM
3620 REM                    ACTUALLY EDITING
3630 REM
3640 LOCATE LNSAV+YOFF,XSAV+XOFF:PRINT A$:LOCATE LNSAV+YOFF,XSAV+XOFF+1
3650    LINE INPUT "",REST$
3660    NW$=A$+LEFT$(REST$,NOWLNG-1): NWLONG=LEN(NW$)
3662  FOR I = 1 TO NWLONG:C=ASC(MID$(NW$,I,1))
3665        IF C > 96 AND C < 123 THEN C=C-32
3667        NW$=LEFT$(NW$,I-1)+CHR$(C)+RIGHT$(NW$,NWLONG-I)
3668  NEXT I
3670    IF FILDES(3,NOWFLD,NOWTYP) = 0 AND FILDES(4,NOWFLD,NOWTYP) = 0 THEN LN$=
        LEFT$(DATALN$,NOWXPS-1)+NW$+SPACE$(NOWLNG-LEN(NW$))+RIGHT$(DATALN$,LEN(
      DATALN$)-NOWXPS+1-NOWLNG):LSET DATALN$=LN$:PUT #2,NOWLN+FRSTLN-1:GOTO 3700
3680    NW = VAL(NW$): NW$=STR$(NW) : NW$=SPACE$(NOWLNG-LEN(NW$))+NW$
3690    IF NW => FILDES(3,NOWFLD,NOWTYP) AND NW <= FILDES(4,NOWFLD,NOWTYP) THEN
        LN$=LEFT$(DATALN$,NOWXPS-1)+NW$+RIGHT$(DATALN$,LEN(DATALN$)-NOWXPS+1-
          NOWLNG):LSET DATALN$=LN$:PUT #2,NOWLN+FRSTLN-1
3700 REM
3710 REM                  IF THIS IS A SCHEDULING AT, ASK WHY THE CHANGE
3720 REM
3740 IF NOWTYP < NUMCRD THEN 3410
3750    NOWEVT$=MID$(DATALN$,FILDES(1,IFLD,NOWTYP),FILDES(2,IFLD,NOWTYP) )
3760    FOR I = 1 TO EVNTS
3770        IF NOWEVT$ = MID$(LIN$(I),9,LEN(NOWEVT$)) THEN 3870
3780    NEXT I
3790       EVNTS=EVNTS+1:DGT=1:IF EVNTS>9 THEN DGT=2:IF EVNTS=100 THEN  DGT=3
3800      LIN$(EVNTS)=NOWMDL$+RIGHT$(STR$(EVNTS),DGT)+STRING$(3-DGT,"_")+NOWEVT$
3810      VISFL$=LEFT$(LIN$(EVNTS),8): I = EVNTS
3820      OPEN "I",#1,"B:NOT-YET.SCR": OPEN "O",#5, "B:"+VISFL$+".SCR"
3825     LINE INPUT #1, LN$: PRINT #5, NOWEVT$              'PUT THE HEADER LINE
3830      IF EOF(1) THEN 3860                         'COPY "NOT-YET.SCR" TO NEW FILE
3840         LINE INPUT #1,LN$
3850         PRINT #5,LN$ : GOTO 3830
3860      CLOSE #1 : CLOSE #5
3870     VISFL$=LEFT$(LIN$(I),8)
3872     EXT$="SCR"
3875     IF COMNTS = 0 THEN 3910 ELSE IF COMNTS = -1 THEN 3410 ELSE EXT$="BAK"
3880    OPEN "O",#1,"B:"+VISFL$+".BAK":CLOSE #1
3890     KILL "B:"+VISFL$+".BAK"
3900        NAME "B:"+VISFL$+".SCR" AS "B:"+VISFL$+".BAK"       'MAKE A BACKUP
3910    CLS  : NOWX = 1 : NOWY = 2
3920    LOCATE 1,1:COLOR 0,7:PRINT "USE ARROW KEYS TO MOVE";:
        LOCATE 1,40:PRINT "PRESS 'HOME' WHEN FINISHED":COLOR 7,0
3930 OPEN "I",#1,"B:"+VISFL$+"."+EXT$: I = 0
3940 IF EOF(1) THEN 3980
3950     LINE INPUT #1,LN$
3960     IF I =>23 THEN 3980 ELSE  I = I + 1: PRINT LN$;
3970     GOTO 3940
3980 CLOSE #1
3990 NOWCHR$=CHR$(SCREEN(NOWY,NOWX) ):COLOR 0,7:LOCATE NOWY,NOWX:PRINT NOWCHR$
3995 COLOR 7,0
```

```
4000 A$ = INKEY$:IF A$="" THEN 4000         ' CHECK FOR TERMINAL INPUT
4010 MOVE = 0 : MVFLD = 0
4020 IF A$=HOME$ THEN 4140 ELSE
4030 IF A$=DNAROW$ THEN MOVE =  1 : GOTO 4080 ELSE  'MOVE DOWN?
4040 IF A$=UPAROW$ THEN MOVE = -1 : GOTO 4080 ELSE  'MOVE UP?
4050 IF A$=RTAROW$ THEN MVFLD = 1 : GOTO 4080 ELSE  'MOVE RIGHT?
4060 IF A$=LFAROW$ THEN MVFLD= -1 : GOTO 4080 ELSE  'MOVE LEFT?
4070 IF A$=>" " AND A$<="~" THEN NOWCHR$=A$:MVFLD = 1 ELSE 4000
4080 LOCATE NOWY,NOWX : PRINT NOWCHR$;
4090 NOWX = NOWX + MVFLD: IF NOWX > 80 OR NOWX < 1 THEN NOWX = 1
4100 NOWY = NOWY + MOVE : IF NOWY > 24 OR NOWY < 2 THEN NOWY = 2
4110 NOWCHR$ = CHR$(SCREEN(NOWY,NOWX) )
4120 COLOR 0,7 : LOCATE NOWY,NOWX : PRINT NOWCHR$;:COLOR 7,0
4130 GOTO 4000
4140 REM
4150 REM           READ THE SCREEN AND LOAD THE FILE
4160 REM
4170 IF COMNTS = 0 THEN COLOR 7,0:GOTO 3310  ' JUST VIEWING THE COMMENT SCREEN
4180 OPEN "O",#1,"B:"+VISFL$+".SCR"       'ACTUALLY UPDATING THE SCREEN
4190 FOR I = 1 TO 23
4200    LN$=""
4210    FOR J = 1 TO 80
4220      LN$=LN$+CHR$(SCREEN(I+1,J))
4230    NEXT J
4240    PRINT #1,LN$
4250 NEXT I
4260 CLOSE #1
4270 COLOR 7,0
4280 GOTO 3310
4290 REM
4300 REM          SEE IF THE MOVED CURSOR IS ON THE CURRENT SCREEN
4310 REM
4320 NOWFLD=NOWFLD+MVFLD+100*MVSCRN:IF NOWFLD > CRDFLD(NOWTYP) OR NOWFLD < 1
     THEN NOWFLD=1
4330 NOWLN=NOWLN+MOVE+SCRNLN*MVSCRN
4340 IF NOWLN < 1 OR NOWLN > SCRNLN OR NOWLN-1+FRSTLN > MAXFIL THEN NOWLN=1
4350 FRSTLN=FRSTLN+SCRNLN*MVSCRN: IF FRSTLN < 1 OR FRSTLN > MAXFIL THEN FRSTLN=1
4360 IRECD= NOWLN-1+FRSTLN
4370     FOR I = 1 TO NUMCRD:IF IRECD => CRDNUM(I) THEN NOWTYP=I: NEXT I
4380 IF NOWFLD > CRDFLD(NOWTYP) THEN NOWFLD=1
4390 IF MVSCRN = 0 THEN 3410 ELSE 3310
4400 REM
4410 RETURN
```

```
4420 REM
4430 REM          INSERT  LINES
4440 REM
4450 NEWLIN = FRSTLN+NOWLN
4460 IF NEWLIN = CRDNUM(NOWTYP+1) THEN NEWTYP = NOWTYP + 1 ELSE NEWTYP = NOWTYP
4470      LN$ = SPACE$(30)                          ' BLANK FILL NEW LINE
4480      FOR I = 1 TO CRDFLD(NEWTYP)          'SET NUMERICS TO THEIR MINIMUMS
4490        NMLNG =FILDES(2,I,NEWTYP):NM$=RIGHT$(STR$(FILDES(3,I,NEWTYP)),NMLNG)
4500          NM$=SPACE$(NMLNG-LEN(NM$))+NM$
4510         IF(FILDES(4,I,NEWTYP))=0 THEN NM$="|"+STRING$(NMLNG-1,"_")
4520         LN$=LEFT$(LN$,FILDES(1,I,NEWTYP)-1)+NM$
4530        NEXT I
4540 MAXFIL = MAXFIL + 1
4550 MOVLIN = MAXFIL - NEWLIN
4560 FOR MOVIT = 1 TO MOVLIN
4570     GET #2,MAXFIL-MOVIT
4580     PUT #2,MAXFIL+1-MOVIT
4590 NEXT MOVIT
4600 REM
4610 REM            IF THIS IS A merge NEED TO PRINT THE 5 MERGE RECORDS
4620 REM
4630 IF ITEM <> 3 THEN 4850
4640    COLOR 0,7: LOCATE KNTSAV - 6 + YOFF,1
4650    PRINT  " DEPRESS LINE FEED TO INSERT THE LINE, DELETE TO IGNOR ";
4660    PRINT "UP & DOWN ARROWS TO MOVE";:COLOR 7,0:
4670    LOCATE CSRLIN,10 :PRINT "LINE FEED";:LOCATE CSRLIN,40:PRINT "DELETE"
4680    A$=INKEY$: IF A$="" THEN 4680
4690    IF A$ = DELET$ THEN 4850  ELSE
4700    IF A$ = LINFED$ THEN LN$ = MERGLN$(MRGREC): GOTO 4850   ELSE
4710    IF A$ = UPAROW$ THEN MOVE=-1 ELSE IF A$ = DNAROW$ THEN MOVE=1 ELSE 4680
4720    COLOR 7,0:LOCATE KNTSAV-6+YOFF+MRGREC,2:PRINT LEFT$(MERGLN$(MRGREC),78);
4730     MRGREC = MRGREC + MOVE
4740     IF MRGREC < 1 THEN MRGREC = 1
4750     IF MRGREC > ABS(MERGMX) AND MERGMX < 0 THEN MRGREC = 1
4760     IF MRGREC <= ABS(MERGMX) THEN COLOR 0,7:LOCATE KNTSAV-6+YOFF+MRGREC,2:
           PRINT LEFT$(MERGLN$(MRGREC),78);:COLOR 7,0:GOTO 4680
4770       MRGREC = 1
4780       FOR I = 1 TO 4
4790         MERGLN$(I) = MERGLN$(I+1)
4800         COLOR 7,0:LOCATE KNTSAV-6+YOFF+I,2:
             PRINT LEFT$(MERGLN$(I),78);
4810       NEXT I
4820       IF EOF(4) THEN MERGMX = -4 : CLOSE #4:LOCATE KNTSAV-6+YOFF+5,2:
           PRINT SPACE$(78): GOTO 4720
4830       LINE INPUT #4,MERGLN$(5)
4835       LOCATE KNTSAV-6+YOFF+5,2: PRINT LEFT$(MERGLN$(5),78)
4840       GOTO 4720
```

```
4850 LSET DATALN$ = LN$
4860 PUT #2,NEWLIN
4865 IF ITEM <> 3 THEN 3320
4870 FOR ICHEK = 1 TO EVNTS                'SEE IF THE NEW RECORD IS IN IDX
4880   IF MID$(LN$,FILDES(1,IFLD,NEWTYP),FILDES(2,IFLD,NEWTYP) ) = LIN$(ICHEK)
             THEN 3320
4890 NEXT ICHEK
4900 OPEN "I",#5,"B:SCR"+MID$(MERGFL$,4,5) + ".IDX"
4910 IF EOF(5) THEN MSCREN$ = "NOT-YET": GOTO 4960
4920    INPUT #5,MEVENT$,MSCREN$
4930    IF MEVENT$ = MID$(LN$,FILDES(1,IFLD,NEWTYP), FILDES(2,IFLD,NEWTYP) )
           THEN 4960
4940 GOTO 4910
4950 REM
4960 REM          ADD TO THE IDX
4970 REM
4980 EVNTS = EVNTS + 1
4990 LIN$(EVNTS) = MSCREN$+SPACE$(8-LEN(MSCREN$)) + MEVENT$
5000 CLOSE #5
5010 GOTO 3320
5020 REM
5030 REM               DELETE THE CURRENT  LINE
5040 REM
5050 MAXFIL = MAXFIL - 1
5060 NEWLIN = FRSTLN+NOWLN - 1
5070 FOR MOVIT = NEWLIN TO MAXFIL
5080     GET #2,MOVIT + 1
5090     PUT #2,MOVIT
5100 NEXT MOVIT
5110 GOTO 3320
5120 REM
5130 REM    THIS ROUTINE BUILDS THE EDIT HELP KEY SCREEN
5140 REM
5150 KEYS$(1)="<——":KEYS$(2)="——>":KEYS$(3)="  |":KEYS$(4)="LN FD"
5160 KEYS$(5)="HOME":KEYS$(6)="  +":KEYS$(7)="  |":KEYS$(8)="DELET"
5170 KEYS$(9)="HELP":KEYS$(10)="ENTER":KEYS$(11)="  -":KEYS$(12)=" ESC"
5180 XPOS(1)=1: XPOS(2)=21: XPOS(3)=41:XPOS(4)=61
5190 XPOS(5)=1: XPOS(6)=21: XPOS(7)=41:XPOS(8)=61
5200 XPOS(9)=1: XPOS(10)=21: XPOS(11)=41:XPOS(12)=61
5210 YPOS(1)=19:YPOS(2)=19:YPOS(3)=19:YPOS(4)=19
5220 YPOS(5)=21:YPOS(6)=21:YPOS(7)=21:YPOS(8)=21
5230 YPOS(9)=23:YPOS(10)=23:YPOS(11)=23:YPOS(12)=23
5240 EXPLAN$(1) = "MOVE LEFT":EXPLAN$(2)="MOVE RIGHT":EXPLAN$(3)="MOVE UP"
5250 EXPLAN$(4) = "INSERT LINE":EXPLAN$(8)="DELETE LINE":EXPLAN$(12)="LEAVE DSS"
5260 EXPLAN$(5) = "LAST OPTION":EXPLAN$(6)="NEXT SCREEN":EXPLAN$(7)="MOVE DOWN"
5270 EXPLAN$(9) = "RENDER HELP":EXPLAN$(10)="CHANGE ITEM":EXPLAN$(11)="PRIOR SCR
EEN"
5280 CLS
5290 A=FRE(" ")
```

```
5300 LOCATE 1,40-LEN(MODELN$)/2:COLOR 0,7:PRINT MODELN$:COLOR 7,0
5310 LINE (0,0)-(639,150),7,B                               'DRAW TOP BOX
5320 LINE (0,152)-(639,215),7,B                               'DRAW HELP BOX
5330 FOR KEYS=1 TO 12
5340     LINE ((XPOS(KEYS)-1)*8,YPOS(KEYS)*9-12)-((XPOS(KEYS)+4)*8+1,YPOS(KEYS)*9
+1),7,BF
5350     LOCATE YPOS(KEYS),XPOS(KEYS)
5360     COLOR 0,7:PRINT KEYS$(KEYS):COLOR 7,0
5370    LOCATE YPOS(KEYS),XPOS(KEYS)+7:PRINT EXPLAN$(KEYS)
5380 NEXT KEYS
5390 COLOR 0,7
5400 LINE ((XPOS(3)-1)*8+12,YPOS(3)*9-5)-STEP(8,-3)
5410 LINE -STEP(8 , 3)
5420 LINE ((XPOS(7)-1)*8+12,YPOS(7)*9-4)-STEP(8,3 )
5430 LINE -STEP(8 ,-3)
5440 COLOR 7,0
5450 REM
5460 REM              THIS INSERTION IS FOR merge OPTION
5470 REM
5480 IF ITEM <> 3 THEN 5570
5490     COLOR 0,7: LOCATE KNTSAV - 6 + YOFF,1
5500     PRINT SPACE$(36)+MERGFL$+SPACE$(36): COLOR 7,0
5510     FOR I = 1 TO ABS(MERGIX)
5520      LOCATE KNTSAV - 6 + YOFF + I , 2
5530      PRINT LEFT$(MERGLN$(I),78);
5540     NEXT I
5550     LOCATE KNTSAV - 6 + YOFF + MRGREC,2: COLOR 0,7:
5560     PRINT LEFT$(MERGLN$(MRGREC),78);: COLOR 7,0
5570 RETURN
5580 REM
5590 REM    GOING BACK A SCREEN
5600 REM
5610 CLOSE #1: CLOSE #3
5620 ON ITEM GOTO 5630,5920,5880,5920,5570
5630 OUTFIL$ = INFIL$
5640 KILFL$=OUTFIL$
5650 FOR J = 1 TO 8
5660       IF MID$(OUTFIL$,J,1)=" " THEN KILFL$=LEFT$(OUTFIL$,J-1):GOTO 5680
5670 NEXT J
5680 OPEN "O",#1,"B:"+OUTFIL$+".BAK":CLOSE #1:KILL "B:"+KILFL$+".BAK" :
                NAME "B:"+OUTFIL$+".INP" AS "B:"+OUTFIL$+".BAK"
5690 OPEN "O",#1,"B:"+OUTFIL$+".INP"
5700 CLS
5710 FOR IREC = 1 TO MAXFIL
5720     GET #2,IREC
5730      IF IREC MOD 30 = 0 THEN CLS:LOCATE 13,31:COLOR 0,7:
          PRINT "RECORD #";IREC;" OF ";MAXFIL;" IS BEING LOADED";:COLOR 7,0
5740     PRINT #1,DATALN$
5750 NEXT IREC
5760 CLOSE
```

```
5770 KILFL$=NOWMDL$
5780 FOR J = 1 TO 5
5790      IF MID$(NOWMDL$,J,1)=" " THEN KILFL$=LEFT$(NOWMDL$,J-1):GOTO 5810
5800 NEXT J
5810 OPEN "O",#1,"B:SCR"+KILFL$+".BAK":CLOSE #1:KILL "B:SCR"+KILFL$+".BAK" :
        NAME "B:SCR"+NOWMDL$+".IDX" AS "B:SCR"+NOWMDL$+".BAK"
5820 OPEN "O",#1,"B:SCR"+NOWMDL$+".IDX"
5830 FOR I = 1 TO EVNTS
5840   PRINT #1,QUOTE$+MID$(LIN$(I),9,16)+QUOTE$+" "
              +QUOTE$+LEFT$(LIN$(I),8)+QUOTE$
5850 NEXT I
5860 CLOSE #1
5870 GOTO 6220
5880 REM
5890 REM            FOR MERGE
5900 REM
5910 SCRNLN = KNTSAV-2
5920 REM
5930 REM                 GET A NEW FILE NAME, DESCRIPTION AND ADD TO AVAILABLE FILE
5940 REM
5950 CLS
5960 LOCATE 1,35:COLOR 0,7:PRINT "CURRENT FILES ARE:":LOCATE 2,2:PRINT " FILE":
     LOCATE 2,20: PRINT "DESCRIPTION":COLOR 7,0
5970 OPEN "O",#1,"B:"+FILEIN$+".BAK":CLOSE #1:KILL "B:"+FILEIN$+".BAK" :
                    NAME "B:"+FILEIN$+".MNU" AS "B:"+FILEIN$+".BAK"
5980 OPEN "I",#1,"B:"+FILEIN$+".BAK":OPEN "O",#3,"B:"+FILEIN$+".MNU"
5990 IF EOF(1) THEN 6020
6000   LINE INPUT #1,LN$
6010   PRINT LEFT$(LN$,8);" ";RIGHT$(LN$,LEN(LN$)-8):PRINT #3,LN$:GOTO 5990
6020 CLOSE #1
6030 LOCATE 23,5
6040 INPUT "PLEASE INPUT A FIVE CHARACTER FILE NAME FOR THE NEWLY CREATED FILE "
     ,NEWFL$
6050 NEWFL$=LEFT$(NEWFL$,5):NEWFL$=NOWAT$+NEWFL$+STRING$(5-LEN(NEWFL$),"_")
6060 FOR I = 1 TO 3
6070 IF MID$(NEWFL$,I,1) => "A" AND MID$(NEWFL$,I,1) <= "Z" THEN
     NEWFL$=LEFT$(NEWFL$,I-1) + CHR$(ASC( MID$(NEWFL$,I,1) )+32)
     + MID$(NEWFL$,I+1,LEN(NEWFL$)-I)
6080 NEXT I
6090 NOWMDL$=MID$(NEWFL$,4,5)
6100 OPEN "I",#1,"B:"+FILEIN$+".BAK"
6110 IF EOF(1) THEN 6130
6120   LINE INPUT #1,LN$:IF LEFT$(LN$,8) = NEWFL$ THEN 6020 ELSE 6110
6130 LOCATE 23,5: PRINT SPACE$(74)
6140 LOCATE 23,5:PRINT "PLEASE INPUT UP TO 70 CHARACTER FILE DESCRIPTION";
6150 LOCATE 24,5:LINE INPUT "",NEWTIT$
6160 LN$=NEWFL$  +LEFT$(NEWTIT$,70)
6170 MODELN$=LEFT$(NEWTIT$,70)
6180 PRINT #3,LN$
6190 CLOSE #1
6200 OUTFIL$ = NEWFL$ : INFIL$=OUTFIL$
6210 GOTO 5640
6220 CLS
6230 CLOSE:GOSUB 1300:RETURN
```

```
6240 REM
6250 REM             USE REG VIDEO STARTING AT LNSAV,XSAV FOR NOWLNG
6260 REM
6270 LOCATE LNSAV+YOFF,XSAV+XOFF : COLOR 7,0 : PRINT STRSAV$
6280 COLOR 0,7 : LOCATE NOWLN+YOFF,NOWXPS+XOFF : PRINT NOWFLD$ : COLOR 7,0
6290 LNSAV=NOWLN:XSAV=NOWXPS:STRSAV$=NOWFLD$
6300 RETURN
6310 REM    THIS IS THE EDIT HELP. USE A RANDOM FILE AND ACCESS ONE RECORD/FIELD
6320 REM
6330 J = 0
6340 FOR I = 1 TO NOWTYP - 1
6350    FOR K = 1 TO CRDFLD(I)
6360        J=J+1                    'THE PLACE FOR THE LAST RECORD BEFORE THIS TYPE
6370    NEXT K
6380 NEXT I
6390 IREC = J + NOWFLD               ' THE RECORD FOR THE ITEM HI-LIGHTED
6400 GET #3,IREC
6410 LN$ = SPACE$(80)
6420 FOR I = 18 TO 24:LOCATE I,1:PRINT LN$;:NEXT I          'BLANK THE BOTTOM OUT
6430 FOR I = 1 TO 4: LOCATE 18 + I,8 : PRINT LNHLP$(I): NEXT I 'PRINT THE HELP
6440 COLOR 0,7 : LOCATE 23,35:PRINT "DEPRESS ANY KEY":COLOR 7,0
6450 B$ = INKEY$: IF B$ = "" THEN 6450
6460 FOR I = 18 TO 24:LOCATE I,1:PRINT LN$;:NEXT I          'BLANK THE BOTTOM OUT
6470 GOSUB 5320
6480 RETURN
6490 REM          PRINT THE HELP SCREEN(S)
6500 REM
6510 OPEN "I",#1,"A:"+SCRNFL$+".HLP"
6520 IHLPLN = 0
6530 CLS
6540    IF EOF(1) THEN 6620
6550    LINE INPUT #1,HELPLN$
6560    IHLPLN = IHLPLN + 1
6570    LOCATE 3+IHLPLN,1 : PRINT HELPLN$
6580    IF IHLPLN < 20 THEN 6540
6590 COLOR 0,7:LOCATE 1,50:PRINT"DEPRESS ANY KEY":COLOR 7,0
6600 A$ = INKEY$:IF A$ = "" THEN 6600
6610 GOTO 6520
6620 REM
6630 CLOSE #1
6640 COLOR 0,7:LOCATE 1,40:PRINT"PRESS ANY KEY TO RETURN TO THE MENU":COLOR 7,0
6650 A$=INKEY$: IF A$="" THEN 6650
6660 GOSUB 790
6670 A=FRE(" ")
6675 A$=""
6680 RETURN
```

```
6690 REM
6700 REM          NEED TO LEAVE THE SYSTEM
6710 REM
6720 CLOSE
6730 END
6740 IF ERR=53 AND ERL=6510 THEN PRINT "THERE IS NO MODEL HELP";:RESUME 6640
6750 IF ERR=53 AND ERL=5680 THEN RESUME 5690
6760 IF ERR=53 AND ERL=5810 THEN RESUME 5820
6770 IF ERR=53 AND ERL = 6900 THEN PRINT "THE FILE DOES NOT EXIST";:RESUME 6880
6780 IF ERR<>61 THEN 6920
6790    CLS:PRINT "YOU HAVE RUN OUT OF DISK SPACE!!!"
6800    PRINT:PRINT "You can free up space by deleting the .BAK (backup files)
6810    PRINT " The following are back up files:"
6820    FILES "B:*.BAK"
6830   PRINT "Do you want <A>ll the back-ups erased, <S>elected ones or <N>one"
6840   A$ = INKEY$: IF A$="" THEN 6840
6850   IF A$="A" OR A$="a" THEN KILL "B:*.BAK":RESUME
6860   IF A$="N" OR A$="n" THEN PRINT "CANNOT CONTINUE.  ":RESUME 6690
6870   IF A$<>"S" AND A$<>"s" THEN 5790
6880      INPUT "INPUT THE FILE NAME (WITHOUT THE '.BAK'), RETURN WHEN DONE";LN$
6890      IF LN$ = "" THEN RESUME
6900      KILL "B:"+LN$+".BAK"
6910   GOTO 6880
6920 PRINT "AN ERROR HAS OCCURED. IT WAS #";ERR;" ON LINE #";ERL;". CONTINUE"
```

```
10 DIM LNHLP$(4),LN$(4)
20 DIM CRDTYP( 4), FILDES(4,10,4 ) , CRDNUM ( 4 ), CRDFLD(4)
30 REM              FIRST GET A DISCRIPTION OF THE FILE
40 REM
50 INPUT "What file (gan,pev)",SCRNFL$
55 IF SCRNFL$ = "" GOTO 340
60 OPEN "I",#1,SCRNFL$+"modrc.fld"
70 INPUT #1, NUMCRD                          ' THE NUMBER OF DISTINCT DATA CARDS
80 FOR CRDTYP = 1 TO NUMCRD
90     INPUT #1,CRDFLD(CRDTYP), CRDNUM(CRDTYP)     'NUMBER OF FIELDS & CARDS
100     FOR FLD = 1 TO CRDFLD(CRDTYP)
110        FOR NUMFLD = 1 TO 4
120            INPUT #1, FILDES(NUMFLD,FLD,CRDTYP) '1=START POS, 2=LENGTH
130        NEXT NUMFLD                              '3=NUM MIN,4=MAX (0s=ALPHA)
140     NEXT FLD
150 NEXT CRDTYP
160 CLOSE #1
170 REM
180 OPEN "R",#3,LEFT$(SCRNFL$,1)+"fld.hlp",256
190 FIELD #3, 64 AS LNHLP$(1),64 AS LNHLP$(2),64 AS LNHLP$(3),64 AS LNHLP$(4)
200 REM    THIS IS THE EDIT HELP. USE A RANDOM FILE AND ACCESS ONE RECORD/FIELD
210 REM
220 J = 0
230 FOR I = 1 TO NUMCRD
240    FOR K = 1 TO CRDFLD(I)
250        PRINT " CARD #";I;"FIELD #";K;" START=";FILDES(1,K,I);"LENGTH=";
           FILDES(2,K,I);"MIN=";FILDES(3,K,I);"MAX=";FILDES(4,K,I)
253        J = J+1
260        PRINT "GIVE A FOUR LINE DESCRIPTION OF THIS FIELD":GET #3
262        FOR II = 1 TO 4: LN$(II) = LNHLP$(II): PRINT LN$(II):NEXT II
263        INPUT "Is this help message sufficient",ANS$
264        IF ANS$ = "y" OR ANS$ = "Y" THEN 290
265        FOR II = 1 TO 4
266          PRINT LN$(II);:INPUT "ok?";ANS$:IF ANS$="" OR ANS$="y" THEN 269
268          LINE INPUT LN$(II)
269        NEXT II
270       FOR II = 1 TO 4
272         LSET LNHLP$(II)=LN$(II)
274       NEXT II
280      PUT #3, J
290    NEXT K
300 NEXT I
310 REM
320 REM          NEED TO LEAVE THE SYSTEM
330 REM
340 CLOSE
350 END
```

```
10 DIM SCHEDL$(100), LIN$(25),FIL$(100)
20 UPAROW$ = CHR$(30):MNSIGN$ = CHR$(45)
30 DNAROW$ = CHR$(31):PLSIGN$ = CHR$(43)
40 ESC$ = CHR$(3):CNTLC$ = CHR$(27)
50 CR$ = CHR$(13): HOME$ = CHR$(11)
55 HELP$=CHR$(1)
60 REM
65 ISTM = 0
67 CLS
70 REM    SINCE BASCOM ERASES THE SCREEN, RE-WRITE IT
80 REM
100 OPEN "I",#3,"GANTT.OUT"
110 NUMPAG = 24
120 NUMLIN = 0
130 IF ISTM=-1 THEN LIN$(1) = NOWLIN$:NUMLIN = 1:CLS:LOCATE 1,1:PRINT LIN$(1)
140 ISTM = -1
150 FOR ILIN = 1 TO NUMPAG
160    IF EOF(3) THEN IEND = -1: GOTO 290
170    LINE INPUT # 3 , NOWLIN$
180 IF NOWLIN$ = " " THEN 240
190    IF LEFT$(NOWLIN$,1) = "1" AND ILIN <> 1 THEN 245
200    NUMLIN = NUMLIN + 1
210    LIN$(NUMLIN) = NOWLIN$
220    LOCATE NUMLIN,1
230    PRINT LIN$(NUMLIN)
240 NEXT ILIN
245 GOSUB 915
250 REM
260 REM        1ST CHECK TO SEE WHETHER AN EVENT EXPLAINATION IS DESIRED
270 REM
280 A$ = INKEY$ : IF A$ = CR$ THEN 920
290 REM
300 REM            WHAT KEY DID HE PRESS?
310 REM
320 IROW = NUMLIN
330 EVENT$ =  MID$(LIN$(IROW),2,16)
335 COLOR 0,7 : LOCATE IROW,2 : PRINT EVENT$
340 A$ = INKEY$ : IF A$ = "" THEN 340
350   COLOR 7,0
360   LOCATE IROW,2
370   PRINT  EVENT$
```

```
380   IF A$ = UPAROW$ OR A$ = MNSIGN$ THEN IROW = IROW - 1   ELSE
390   IF A$ = DNAROW$ OR A$ = PLSIGN$ THEN IROW = IROW + 1   ELSE
400   IF A$ = ESC$ OR A$ = CNTLC$  THEN 950   ELSE
405   IF  A$ = CR$ THEN 920   ELSE
407   IF  A$ = HOME$ THEN CLOSE:GOTO 60
410   IF A$ = HELP$ THEN 490
420   IF IROW < 3  THEN IROW = 3
430   IF IROW > NUMLIN THEN IROW = NUMLIN
440   EVENT$ =  MID$(LIN$(IROW),2,16)
450   COLOR 0,7
460   LOCATE IROW,2
470   PRINT  EVENT$
480 GOTO 340
490 IF FLAG =  -1 THEN 590
500    FLAG = -1
510    OPEN "I",#1,"EVENTS.INP"
520    FOR IREC = 1 TO 100
530       MAX = IREC
540       INPUT #1,SCHEDL$(IREC),FIL$(IREC)
550       IF EOF(1) GOTO 570
560   NEXT IREC
570   REM
580   CLOSE #1
590 REM
600 REM        SEE IF EVENT$ IS IN THE SCHEDL$
610 REM
620 ISVROW = IROW
630 FOR INDEX = 1 TO MAX
640     ICHEK = INDEX
650     IF EVENT$ = SCHEDL$(ICHEK) THEN 690
660 NEXT INDEX
670 LOCATE IROW,1
680 PRINT " PLEASE RE-ENTER ":GOTO 340
```

```
690 REM
700 REM         CLEAR THE SCREEN, OUTPUT A MESSAGE ON THIS EVENT AND RETURN
710 REM
720 CLS
730 COLOR 7,0
740 ILIN = 0
745 OPEN "I",#2,FIL$(ICHEK)+".SCR"
750 FOR IREC  = 1 TO 24
755    IF EOF(2) THEN 825
760     LINE INPUT #2,REASON$
800     LOCATE IREC,1
810     PRINT REASON$
820 NEXT IREC
825 CLOSE # 2
830 LOCATE 1 ,50 :COLOR 0,7 :PRINT "PRESS RETURN":COLOR 7,0
840 A$ = INKEY$ : IF A$ = "" THEN 840
850 CLS
860 FOR IROW = 1 TO NUMLIN
870    LOCATE IROW,1
880   PRINT LIN$(IROW)
890 NEXT IROW
900 IROW = ISVROW
905 GOSUB 915
910 GOTO 830
915 COLOR 7,0:LOCATE 25,4:PRINT "HELP FOR ASSISTANCE; ENTER FOR NEXT PAGE; HOME
FOR PRIOR PAGES; ESC TO LEAVE";:COLOR 0,7:LOCATE 25,4:PRINT "HELP";:LOCATE 25,25
:PRINT "ENTER";:LOCATE 25,46:PRINT "HOME";:LOCATE 25,68:PRINT "ESC";:COLOR 7,0
917 RETURN
920 REM
930 IF IEND <>-1 THEN 110
950 END
```

```
      SUBROUTINE BACTIM(ISCALE,IDAY81,NUMWEK,IQUATR,CDAY,
     +    CMONTH,CYEAR)
      IMPLICIT CHARACTER (C), LOGICAL (L), DOUBLE PRECISION (D)
      CHARACTER * 3 CMONTH,CQTR(4)
      CHARACTER * 2 CDAY, CYEAR
      DATA CQTR/'JAN','APR','JUL','OCT'/
C
C       THIS ROUTINE BACKTRACKS THE START TIME TO THE BEGINNING
C       OF THE NEXT HIGHEST UNIT OF TIME, FOR A WEEK CHART
C       THE TIME IS STARTED AT THE START OF THE MONTH.
C
      CDAY = '01'
      IF(ISCALE.EQ.1) THEN
        IDAY81 = IDAY81 - NUMWEK + 1
        NUMWEK = 1
      ELSE IF(ISCALE.EQ. 5) THEN
        CALL SINC81(CDAY,CMONTH,CYEAR,IDAY81,IQUATR,NUMWEK)
      ELSE IF(ISCALE.EQ.20) THEN
        CMONTH = CQTR(IQUATR)
        CALL SINC81(CDAY,CMONTH,CYEAR,IDAY81,IQUATR,NUMWEK)
      ELSE
        CMONTH = 'JAN'
        CALL SINC81(CDAY,CMONTH,CYEAR,IDAY81,IQUATR,NUMWEK)
      ENDIF
      RETURN
      END
```

```
C
C       THIS PROGRAM IS THE INTERMEDIATE MODULE BETWEEN PERTCP
C       AND GANTT.  IT LOOKS AT A SORTED LIST OF TO-EVENTS, 1ST
C       START TIME, LAST COMPLETE AND SLACK DAYS
C
      IMPLICIT CHARACTER (C), LOGICAL (L), DOUBLE PRECISION (D)
      CHARACTER * 16 CNEW, CNAME
      DATA IPERT/1/, IGANTT/2/
C
      OPEN(IPERT,FILE='PERTCP.SRT')
      OPEN(IGANTT,FILE='GANTT.INP',STATUS='NEW')
C
      READ(IPERT,200)CNAME,ISTIME,ITMLST,ISLACK
100   CONTINUE
        READ(IPERT,200,END=300)CNEW,NEWIST,NEWLST,NEWSLK
200     FORMAT(A16,3I5)
        IF(CNEW.EQ.CNAME) THEN
          IF(NEWSLK.LT.ISLACK)THEN
             ISTIME = NEWIST
             ITMLST = NEWLST
             ISLACK = NEWSLK
          ENDIF
        ELSE
          WRITE(IGANTT,200)CNAME,ISTIME,ITMLST,ISLACK
          CNAME = CNEW
          ISTIME = NEWIST
          ITMLST = NEWLST
          ISLACK = NEWSLK
        ENDIF
      GO TO 100
300   CONTINUE
      WRITE(IGANTT,200)CNAME,ISTIME,ITMLST,ISLACK
      END
      SUBROUTINE DAY2CH( CDAY,CMNTH,CYEAR,IDAY81,NOWQTR,
     +  NUMWEK )
      IMPLICIT CHARACTER (C),LOGICAL (L),DOUBLE PRECISION (D)
      CHARACTER * 3 CMONTH(12),CMNTH
      CHARACTER * 2 CDAY, CYEAR
      DIMENSION IQTR(4), MONTH(12), CWEEK(7)
      DATA  CMONTH/'JAN','FEB','MAR','APR','MAY','JUN',
     +             'JUL','AUG','SEP','OCT','NOV','DEC'/,
     + MONTH/        31,  59,    90,  120 , 151,  181,
     +              212, 243,   273,  304,  334,  365/,
     +  IWKD81/4/,
     + CWEEK/ 'M','T','W','T','F','S','S'/,I4YEAR/1461/
C
C      THIS ROUTINE IS INPUT THE NUMERIC DAYS SINCE 1 JAN 81
C      (0=1JAN) AND WILL RETURN THE SPECIFIC DAY OF THE WEEK
C      (1-7), THE YEARLY QTR(1-4) AND THE CHARACTER DAYS,
C      MONTH AND YEAR (DDMMMYY)
C
      IF(IDAY81.GE. 0) THEN
C
C     .   FIGURE OUT THE NUMBER OF DAYS, MODULO TO YEARS,QTRS,
C         AND SPECIFIED DAY
C
         NUM4YR = IDAY81/I4YEAR
         IJAN4Y = NUM4YR * I4YEAR
         IDIFF = IDAY81 - IJAN4Y
         IYEAR = IDIFF / 365
         IJANYR = IJAN4Y + IYEAR*365
         IDAYR = IDAY81-IJANYR
C
C       NUM4YR IS THE NUMBER OF FOUR YEAR BLOCKS
C       IJAN4Y IS THE NUMERIC VALUE OF 1 JAN OF THE START OF
C              THE 4 YEAR
C       IYEAR IS THE NUMBER OF COMPLETE YEARS SINCE IJAN4Y
C       IJANYR IS THE NUMERIC VALUE OF 1 JAN OF YEAR OF
C              INTEREST
C       IDAYR IS THE DAY OF THAT YEAR
C
        NUMWEK = MOD(IDAY81+IWKD81-1,7) + 1
C
        IDAY = IDAYR +1
        NOWMNT = 1
        IF(IDAYR.GE.MONTH(1) ) THEN
           ILEAP = 0
           IF(IYEAR.GE.3) ILEAP = 1
           IDAY = 31
           NOWMNT = 12
           IF(IDAYR.EQ.31)THEN
C           FEB 1 OF LEAP YEAR
            IDAY = 1
            NOWMNT = 2
           ENDIF
           DO 100 I = 2,12
              IF( IDAYR .LT. MONTH(I-1)+ILEAP  ) GO TO 100
                  IDAY=IDAYR +1 -MONTH(I-1)
                  IF(I.GE.3)IDAY=IDAY-ILEAP
                  NOWMNT = I
100        CONTINUE
        ENDIF
C
        NOWQTR =  IFIX( FLOAT(NOWMNT)/3. + .67 )
C
        CMNTH = CMONTH(NOWMNT)
C
        CALL YR2CHR(IDAY,C1STDG,C2NDDG)
        WRITE(CDAY,'(2A1)')C1STDG,C2NDDG
C
        NUMYR = 81 + NUM4YR*4 + IYEAR
        CALL YR2CHR(NUMYR,C1STDG,C2NDDG)
        WRITE(CYEAR,'(2A1)')C1STDG,C2NDDG
      ENDIF
C
      RETURN
      END
```

```
C
C       THIS PROGRAM PRODUCES A GANTT CHART, IT USES DATA
C       FROM PERTCP IT LOOKS AT A SORTED LIST OF TO-EVENTS,
C       1ST START TIME, LAST COMPLETE AND SLACK DAYS
C
      IMPLICIT CHARACTER (C),LOGICAL (L),DOUBLE PRECISION (D)
      CHARACTER * 80 CTITLE(2)
      CHARACTER * 40 COMENT,CFILE
      CHARACTER * 16 CEVENT,CNEW
      CHARACTER * 3 CMONTH(12),CMNTH
      CHARACTER * 2 CDAY, CYEAR
      DIMENSION IQTR(4), MONTH(12), ISCALE(4)
     +       ,CLINE(60), CSCALE(4), CWEEK(5)
      DATA IOPT/1/, IGANTT/2/, IPRINT/6/, ISAV/3/,MAX/60/
      DATA CBLANK/' '/,CSLACK/'-'/,CPM/'*'/,CEVENT/'+'/,
     + CBAR/'|'/,
     + ISCALE/1,5,20,65/, ILENTH/24/,CSCALE/'D','W','M','Q'/
C
        OPEN(IOPT,FILE='GANTT.OPT')
        OPEN(ISAV,FILE='GANTT.OUT')
        OPEN(IPRINT,FILE='CON:')
      READ(IOPT,100) CINSCL,ILONG,CDAY,CMNTH,CYEAR,COMENT,
     +   CFILE
100   FORMAT(A1,I7,A2,A3,A2,2A40)
      INSCAL = 1
      DO 150 I = 1,4
        IF(CINSCL.EQ.CSCALE(I)) INSCAL = I
150   CONTINUE
      CALL SINC81(CDAY,CMNTH,CYEAR,IDAY81,IQUATR,NUMWEK)
      IBAK = IDAY81
      CALL BACTIM(ISCALE(INSCAL),IBAK,NUMWEK,IQUATR,CDAY,
     +  CMNTH,CYEAR)
      IF(CINSCL.EQ.'W') MAX = 52
      OPEN(IGANTT,FILE='GANTT.SRT')
C
      ITMPAG = MAX*ISCALE(INSCAL)
      NUMPAG = ILONG/ITMPAG + 1
      DO 900 IPAGE = 1,NUMPAG
         ISTART = (IPAGE-1)*ITMPAG + IBAK
         ILAST = ISTART + ITMPAG - 1
         KNTPAG = 3
         REWIND IGANTT
         CALL TITLE( CTITLE,CDAY,CMNTH,CYEAR,COMENT,
     +                ISCALE(INSCAL),IPAGE,IBAK,NUMWEK,IQUATR)
C
         IF(IPAGE.NE.1) PAUSE
         WRITE(ISAV,200) CTITLE
         WRITE(IPRINT,200) CTITLE
200      FORMAT('1',A79,/,1X,A79,/)
300      CONTINUE
            READ(IGANTT,400,END=900)CNEW,NEWIST,NEWLST,NEWSLK
400         FORMAT(A16,3I5)
```

```
            NEWIST = NEWIST + IDAY81
            NEWLST = NEWLST + IDAY81
            DO 450 I = 1,MAX
450           CLINE(I) = CBLANK
              IF(NEWLST.GE.ISTART.AND.NEWIST.LE.ILAST )THEN
                IST = (NEWIST - ISTART)/ISCALE(INSCAL) + 1
                IF(IST.LT.1) IST = 1
                ILST = (NEWLST - ISTART)/ISCALE(INSCAL) + 1
                IF(ILST.GT.MAX) ILST = MAX
                CHAR = CEVENT
                IF(NEWSLK.EQ.0) THEN
                  CHAR = CPM
                ELSE
                  DO 500 I = IST,ILST
500                CLINE(I)=CSLACK
                   ILST = (NEWLST - NEWSLK - ISTART)/
     +                                      ISCALE(INSCAL)
                   IF( ILST.GT. MAX) ILST = MAX
                   IF( ILST.LE.IST) GO TO 650
                ENDIF
              IF(ILST.LT.IST)ILST = IST
              DO 600 I = IST,ILST
600              CLINE(I) = CHAR
C
650          KNTPAG = KNTPAG + 1
             IF(KNTPAG.GE.ILENTH)THEN
               PAUSE
               KNTPAG = 3
               WRITE( IPRINT,200)CTITLE
               WRITE(ISAV,200) CTITLE
            ENDIF
C
           WRITE( * ,700)CNEW,(CLINE(I),I=1,MAX)
           WRITE( ISAV ,700)CNEW,(CLINE(I),I=1,MAX)
700        FORMAT(1X,A16,3X,60A1)
         ENDIF
        GO TO 300
900   CONTINUE
      END
      IMPLICIT CHARACTER*10 (C), LOGICAL*4 (L), REAL*8 (D)
C
C       This routine will prompt for a new file to be input
C       to PERTCP.
C
      CHARACTER*64 CFILE
      CHARACTER*16 CFROM, CTO
C
      DIMENSION CREASN(8)
      DATA IREAD/0/, IPRINT/0/, IFILE/7/
C
      WRITE( IPRINT,20)
20    FORMAT(' PLEASE INPUT THE NEW FILE NAME => ')
      READ(*,'(A)') CFILE
      OPEN( 7,FILE=CFILE,ACCESS='SEQUENTIAL',STATUS='NEW')
      WRITE(IPRINT,50)
50    FORMAT(' PLEASE INPUT AN EIGHTY CHARACTER EXPLAINATION'
     +      , 'OF THIS FILE',/)
      READ(IREAD,75)CREASN
75    FORMAT(8A10)
      WRITE( IPRINT, 100)
100   FORMAT(' IS THIS DATA FOR A PERT PROBLEM? ')
      CALL YESNO( LANSWR )
      IF( LANSWR ) THEN
        WRITE(IFILE,150)1,CREASN
150     FORMAT(I5,/,8A10)
        WRITE(IPRINT,200)
200     FORMAT(' YOU NEED TO INPUT THE "FROM EVENT",
     +          "TO EVENT",AND ', ' THREE',/,
     +         ' TIMES (MOST LIKELY, PESSIMISTIC, AND ',
     +         ' OPTIMISTIC).',/, ' WHEN DONE,',
     +         'JUST DEPRESS *ENTER*')
        I=0
300     CONTINUE
          WRITE(IPRINT,310)
310       FORMAT(' FROM (16 CHAR)',/,'  TO (16 CHAR)  ',/,
     +           5X,'MOST',
     +           T12,'PESS',T20,'OPTI',/,
     +           T5,'LIKELY',T12,'TIME',T20,'TIME')
          READ( IREAD, 320)CFROM
          READ( IREAD, 320)CTO
          IF(CFROM .EQ. ' ' .OR. CTO .EQ. ' ') GO TO 500
          READ( IREAD, *, ERR= 340) MSTLIK,IPESIS,IOPTIM
320       FORMAT(A16)
330       FORMAT(2A16,3I5)
          WRITE(IFILE,330)CFROM,CTO,MSTLIK,IPESIS,IOPTIM
          GO TO 300
340       CONTINUE
             WRITE( IPRINT,350)
350          FORMAT(' THE *TIMES* MUST BE INTEGERS <',
     +              ' 100,000  DAYS')
         GO TO 300
      ELSE
        WRITE(IFILE,150)0,CREASN
        WRITE(IPRINT,350)
400     CONTINUE
          WRITE(IPRINT,410)
410       FORMAT(' FROM (16 CHAR)',/,' TO (16 CHAR)   ',/,
     +            T5,'TIME')
          READ( IREAD, 320)CFROM
          READ( IREAD, 320)CTO
          IF( CFROM .EQ. ' ' .OR. CTO .EQ. ' ' ) GO TO 500
          READ( IREAD, *,ERR = 440) ITIME
420       FORMAT(2A16,I5)
          WRITE(IFILE,420)CFROM,CTO,ITIME
          GO TO 400
440       CONTINUE
             WRITE( IPRINT,350)
         GO TO 400
      ENDIF
500   CONTINUE
      CLOSE ( IFILE )
      END
      SUBROUTINE YESNO( LANSWR )
      IMPLICIT CHARACTER*10 (C), LOGICAL*4(L), REAL*8(D)
      DATA IREAD/0/, IPRINT/0/
C
      LANSWR = .FALSE.
      READ( IREAD,100,END=500 ) CANS
100   FORMAT(1A1)
      IF( CANS.EQ.'Y'.OR.CANS.EQ.'y') LANSWR = .TRUE.
      RETURN
500   CONTINUE
      RETURN
      END
```

```
C   THE PURPOSE OF THIS PROGRAM IS TO SOLVE CRITICAL PATH
C   PROBLEMS. IT WILL TAKE A PROJECT NETWORK AND DETERMINE
C   THE CRITICAL PATH. THE CRITICAL PATH IN A NETWORK IS THE
C   PATH THAT ALLOWS ALL EVENTS IN THEIR SPECIFIED SEQUENCES
C   TO BE PERFORMED IN THE MINNIMUM AMOUNT OF TIME. THEREFORE
C   IT IS THE LONGEST PATH IN THE NETWORK. WE ALSO WANT TO
C   IDENTIFY THE EVENTS IN THE CRITICAL PATH SO THEY
C   CAN BE MINIMIZED.  THEY ARE CALLED CRITICAL EVENTS.  THIS
C   PROGRAM WILL HANDLE UP TO N EVENTS.  N IS 100 OR LESS.
C   THE FIRST EVENT MUST
C   BE LABLED 1 AND THE LAST ONE MUST BE N.
C   THERE MUST BE A NODE AND NUMBER FROM 1 TO N.
C *********************************************************
C
C   **NOTE**
C
C SINCE PUTTING IN THE ENHANCEMENTS FOR PERT, DATA CARD 1 NOW
C      CONTAINS EITHER A:
C                          0 (THIS IS A CPM PROBLEM)
C                                   - OR -
C                          1 (THIS IS A PERT PROBLEM)
C
C
C   IF THIS IS A PERT PROBLEM IT IS NOW NECESSARY TO PUT IN
C      THREE TIMES, THE MOST LIKELY TIME AND THE PESSIMISTIC
C      TIME, THE OPTIMISTIC TIME.  A WEIGHTED AVERAGE OF THE
C      TIMES WILL THEN BE USED TO COMPUTE THE CRITICAL PATH.
C      THE INPUT FORMAT IS NOW '2A16,3I5'
C      AND THE DATA SHOULD BE ENTERED AS FOLLOWS:
C
C      FROM NODE,TO NODE,MOST LIKELY TIME,PES TIME,OPTIM TIME
C
C
C   THIS DATA WILL BEGIN ON DATA CARD 3, WITH A NEW CARD BEING
C      USED FOR EACH ARC UNTIL ALL ARCS ARE SPECIFIED.  DUMMY
C      ARCS MUST ALSO BE INCLUDED WITH THE TIMES
C      ENTERED AS '0,0,0'.
C
C   DATA CARD 2 NOW CONTAINS THE GENERAL INFORMATION DESCRIBED
C
C
C***********************************************************
C****A PRELIMINARY CARD TELLS THE NUMBER OF JOBS TO BE RUN.
C
C   THEN COMES THE INDIVIDUAL JOB INFORMATION.
C   DATA CARD 1 GIVES GENERAL INFORMATION.  JOB NAME, DATE,
C   CARD 2  STARTS THE NODE TO NODES DATA OF THE NETWORK.
C   CARDS FROM THEN ON GIVE ARC INFORMATION.  NODE IT IS
C   FROM, THE ONE IT IS GOING TO , AND THE TIME INVOLVED.
C   3I5 FORMAT IS USED.  ALL NUMBERS MUST BE RIGHT JUSTIFIED
C
C
```

```
      CHARACTER*16 NAMES(100),NAMFRM,NAMTO
      CHARACTER*7 CDATIN
      REAL VARPTH,FNORML(6),PROB(6)
      INTEGER X,Y,I,N,TOP1,VALUE,P,Q,TIME,FROM,TO,SLACK,TOP2,
     *  K,H,LS,EF,C(20),KO
      INTEGER PRTCPM,OPTTM,PESTM,MLKTM,WTTM
      INTEGER * 2   T(100,100),U(100),V(100),NORMAL(6)
     +   , VARARC(100,100)
      COMMON/NAMES/NAMES,KNTNAM
      COMMON/VARARC/ VARARC
        COMMON/TIMES/T,U,V
        COMMON/STORE/X,N
        DATA IGANTT/2/, IGNTOP/3/,
     +             NORMAL/99,  95,   90,  75,  66,  25/,
     +             FNORML/2.33,1.645,1.28,.67,.44,-.67/
C
C       ***THIS IS A VARIABLE TO DETERMINE WHETHER THIS IS
C           A CPM PROBLEM OR A PERT PROBLEM
C
        OPEN(IGANTT,FILE='GANTT.INP')
        OPEN(5,FILE='PERTCP.INP',STATUS='OLD')
        OPEN(6,FILE='CON:')
        READ(5,25)PRTCPM
1       CONTINUE
C
C   THIS IS A JOB COMMENT .
C
        READ(5,5) (C(I),I=1,10),CDATIN,IGNTDS
 5      FORMAT(10A4,4X,A7,3X,A1)
        WRITE(6,400)
        WRITE(6,6) (C(I),I=1,10)
6       FORMAT(//,10X,10A4)
C
C   THE NUMBER OF NODES IN NETWORK IS DETERMINED FOR THE USER.
C
25      FORMAT(I5)
C
C    SET ALL TIMES IN THE MATRIX TO -1 SO THE UNUSED SLOTS
C    WILL BE IDENTIFIED.
C
         DO 50 P=1,100
              NAMES(P)=' '
              U(P)=-1
              V(P)=-1
              DO 50 Q=1,100
50                 T(P,Q)=-1
      KNTNAM = 0
C
C   READ EXISTING ARCS AND TIMES.
C
      IF(PRTCPM.EQ.0) THEN
C      **THIS IS A CPM PROBLEM
```

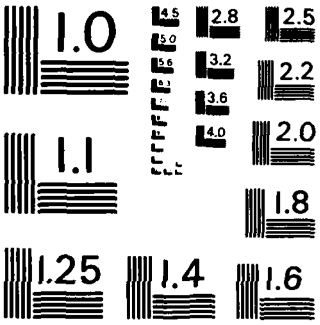
```
       WRITE(6,11)
11     FORMAT(21X,'***THIS IS A CPM PROBLEM')
C
60         READ(5,75,END=100) NAMFRM,NAMTO,TIME
           CALL FNDNAM(NAMFRM,NAMTO,FROM,TO)
           T(FROM,TO)=TIME
75         FORMAT(2A16,I5)
           GO TO 60
      ELSE
C      **THIS IS A PERT PROBLEM
           WRITE(6,12)
12         FORMAT(21X,'***THIS IS A PERT PROBLEM')
C
 70        READ(5,176,END=100) NAMFRM,NAMTO,MLKTM,OPTTM,PESTM
C             **FIND THE WEIGHTED AVERAGE OF THESE TIMES
C
           FX = ((FLOAT(OPTTM+(4*MLKTM)+PESTM))/6.)
           WTTM = INT(FX*10.0)
           CALL FNDNAM(NAMFRM,NAMTO,FROM,TO)
           T(FROM,TO) = WTTM
C
C          ** FIND THE VARIANCE OF THIS ARC (INT*2 TO SAVE SPACE)
C
           IVAR =  10 *INT (   ((FLOAT(OPTTM-PESTM))/6.0)**2)
           IF( IVAR .GT. 32767 ) IVAR = 32767
           VARARC(FROM,TO) = IVAR
176        FORMAT(2A16,3I5)
           GOTO 70
      ENDIF
100        CONTINUE
C
C    THE FIRST TIME IS DESIGNATED AS TIME 0.
           N=KNTNAM
           U(1)=0
           X=N+1
C
C    CALL STORAGE AND SET STORAGE INDEXES TO ZERO.
           CALL ZERO
C
C    CHECK TO MAKE SURE ALL NODES HAVE OUT GOING ARCS.
C
           P=1
           Q=1
105        IF(T(P,Q).GE.0) GO TO 110
           Q=Q+1
           IF(Q.LE.N) GO TO 105
           WRITE(6,400)
           WRITE(6,107) NAMES(P)
107        FORMAT(//,5X,' THERE IS NO EVENT LEAVING ',A16)
           WRITE(6,400)
           GO TO 420
110        P=P+1
```

AD-A161 690 DECISION SUPPORT SYSTEM FOR ASD (AERONATICAL SYSTEMS 3/3
DIVISION) PROGRAM MANAGERS(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF SYST..
UNCLASSIFIED T W BROTHERTON SEP 85 AFIT-GSM/LSY/85S-5 F/G 9/2 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

```
         IF(P.EQ.N) GO TO 115
         Q=1
         GO TO 105
C
C    CHECK ALL NODES FOR INCOMING ARCS.
115      CONTINUE
         P=N
         Q=N
120      IF(T(P,Q).GE.0) GO TO 124
         P=P-1
         IF(P.GT.0) GO TO 120
         WRITE(6,400)
         WRITE(6,123) NAMES( Q )
123      FORMAT(//,5X,'EVENT ',A16,' HAS NO PREDECESSOR ',
     +          'EVENTS')
         WRITE(6,400)
         GO TO 420
124      Q=Q-1
         IF(Q.EQ.1) GO TO 125
         P=N
         GO TO 120
C
C    CALCULATE THE EARLIEST TIMES EACH NODE CAN BE REACHED.
C
C   STORE THE NODE DESIGNATORS SO THEY CAN BE CALLED 1 BY 1
C    AND SOLVED.
C
125      X=X-1
         CALL PUSH1
         IF (X.NE.2) GO TO 125
C
C    ALL NODES TO BE SOLVED ARE STORED. CALL FIRST NODE
C     TO BE SOLVED.
C
         CALL POP 1
C
C    CALL ALGORITHM TO SOLVE FOR U(X) WHICH IS THE EARLIEST
C     TIME NODE X CAN BE REACHED.
C
150      CALL UEARLY
C
C    IF INSUFFICIENT DATA TO SOLVE FOR U(X) STORE IN STACK 2
C     FOR LATER SOLUTION.
         IF(U(X).LT.0)CALL PUSH2
C
C    CALL NEXT U
         CALL POP1
C
C    IF STACK 1 IS NOT EMPTY THEN GO BACK TO STACK 1 ELSE
C      GO ON TO STACK2
         IF(X.NE.0) GO TO 150
C
```

```
C   POP FIRST OF UNSOLVED US OFF STACK 2
         CALL POP2
C
C   IF STACK 2 EMPTY GO ON TO CALCULATE V(X)S, LATEST TIMES
C   A NODE X CAN BE REACHED ,ELSE SOLVE NEXT U.
         IF(X.EQ.0) GO TO 200
175      CALL UEARLY
C
C   IF CAN NOT BE SOLVED STORE ON STACK 1 FOR LATER SOLUTION.
C
         IF(U(X).LT.0) CALL PUSH1
         CALL POP2
C
C   IF STACK 2 IS NOT EMPTY THEN FIGURE NEXT U
C              ELSE CHECK STACK 1.
         IF(X.NE.0) GO TO 175
         CALL POP1
C
C   IF STACK 1 IS NOT EMPTY THEN GO BACK TO SOLVE U  IN IT,
C        ELSE GO 0 ON TO CALCULATE V FOR EACH NODE.
          IF(X.NE.0) GO TO 150
C
200      CONTINUE
C   SOLVE FOR V(X) THE LATEST TIME AN  EVENT CAN BE PERFORMED
C     WITHOUT HOLDING UP THE SCHEDULE OF THE JOB.
         V(N)=U(N)
C
C   SOLVING FOR V IS THE SAME AS FOR SOLVING FOR U.
         X=N
225      X=X-1
         CALL PUSH1
         IF(X.NE.1) GO TO 225
         CALL POP1
250      CALL VLATEST
         IF(V(X).LT.0) CALL PUSH2
         CALL POP1
         IF(X.NE.0) GO TO 250
         CALL POP2
         IF(X.EQ.0) GO TO 280
275      CALL VLATEST
         IF(V(X).LT.0) CALL PUSH1
         CALL POP2
         IF(X.NE.0) GO TO 275
         CALL POP1
         IF(X.NE.0) GO TO 250
280      CONTINUE
C
C   PRINT THE DATA
C
      WRITE(6,400)
         X=0
287      CONTINUE
```

```
         IF(MOD(X,20).EQ.0)WRITE(6,285)
285      FORMAT(//,T8,'EVENT',T28,'EARLIEST',T46,'LATEST ',
     +   T60,'# OF ',/, T26,'FINISH DAY', T43,'FINISH DAY',
     +   T58,'SLACK DAYS')
       X=X+1
         SLACK=V(X)-U(X)
         IF(PRTCPM.EQ.1) THEN
           FU = FLOAT(U(X)/10)
           FV = FLOAT(V(X)/10)
           FSLACK= FLOAT(SLACK/10)
           WRITE(6,291)NAMES( X ),FU,FV,FSLACK
291        FORMAT(2X,A16,T30,F5.1,T46,F5.1,T60,F5.1)
         ELSE
           WRITE(6,290)NAMES(X),U(X),V(X),SLACK
290        FORMAT(2X,A16,T30,I5,T46,I5,T60,I5)
         ENDIF
         IF(SLACK.NE.0) GO TO 287
         WRITE(6,360)
         IF(X.NE.N) GO TO 287
         WRITE(6,400)
       H=0
310    CONTINUE
       IF(MOD(H,20).EQ.0)WRITE(6,300)
300      FORMAT(//,T18,'EVENT',T37,'LENGTH',T45,'FIRST',T53,
     +   'LAST',T59,
     +   'EARLY',T66,'LATEST',T74,'SLACK',/,T45,'START',T52,
     +   'START',T58,'FINISH',T66,'FINISH',T75,'DAYS')
       H=H+1
       IF(H.GT.N) GO TO 405
        J=0
320     J=J+1
        IF(J.GT.N) GO TO 310
        IF(T(H,J).LT.0) GO TO 320
        SLACK=V(J)-(U(H)+T(H,J))
        LS=V(J)-T(H,J)
        EF=U(H)+T(H,J)
        IF(PRTCPM.EQ.1) THEN
          WRITE(IGANTT,352)NAMES (J ),U(H)/10,V(J)/10,SLACK/10
352            FORMAT(A16,3I5)
          FU = FLOAT(U(H))/10.
          FLS = FLOAT(LS)/10.
          FEF = FLOAT(EF)/10.
          FV = FLOAT(V(J))/10.
          FSLACK = FLOAT(SLACK)/10.
          WRITE(6,351)NAMES( H ),NAMES (J ),
     +          FLOAT(T(H,J))/10.,FU,FLS,FEF,FV,FSLACK
351       FORMAT(2X,A16,'=>',A16,
     +           T38,F5.1,T45,F5.1,T52,F5.1,T58,
     *           F5.1,T67,F5.1,T74,F5.1)
        ELSE
          WRITE(6,350)NAMES(H),NAMES(J),
     +                T(H,J),U(H),LS,EF,V(J),SLACK
```

```
350       FORMAT(2X,A16,'=>',A16,
     +       T39,I3,T45,I5,T52,I5,T58,I5,T67,I5,T74,I5)
          WRITE(IGANTT,352)NAMES (J ),U(H),V(J),SLACK
        ENDIF
C
C   PUT A STAR BESIDE THE CRITICAL NODES.
C
        IF(SLACK.NE.0) GO TO 320
          WRITE(6,360)
360       FORMAT(1H+,'*')
          IF(PRTCPM.EQ.1) THEN
            VARPTH = VARPTH + FLOAT(VARARC(H,J))/10.
          ENDIF
        GO TO 320
400     FORMAT(//,10X,
     +  '*********************************************')
405     WRITE(6,410)
410     FORMAT(//,18X,'*    THIS IS ON THE CRITICAL PATH.')
        WRITE(6,400)
        IF(PRTCPM.EQ.1)THEN
            XN = (FLOAT(U(N)))/10.
            ILONG = U(N)/10
            WRITE(6,422) XN,NAMES(1),NAMES(N)
422      FORMAT(//,10X,'THERE ARE ',F8.1,' DAYS ON THE',
     +      ' CRITICAL PATH'
     +  ,/, ' BETWEEN THE FIRST EVENT *',A16,'* AND THE',
     +      ' LAST *' , A16,'*.')
         WRITE(6,416)VARPTH
416      FORMAT(//,10X,
     +   'THE VARIANCE OF THE CRITICAL PATH IS:',2X,F8.2,
     +   /,5X,'THE VARIANCE CAN BE USED TO DETERMINE THE',
     +   ' PROBABILITY OF',
     +   /,5X,'FINISHING A JOB BEFORE A CERTAIN DATE.')
C
C     % CERTAINTY THAT THE NETWORK WILL BE COMPLETED
C    95    90    80    75    50    25
C 1.645   1.38   .84   .67   .0   -.67  * SQRT(VARPTH) + XN
        STDDEV = SQRT(VARPTH)
        DO 4005 II = 1,6
4005       PROB(II) = FNORML(II) * STDDEV + XN
        WRITE(6,414)(NORMAL(II),II=1,6),(PROB(II),II=1,6)
414     FORMAT(//,
     + ' BY ASSUMING THE TIMES ARE OF NORMALLY DISTRIBUTED,'
     + ,' PROBABLE COMPLETION ',/,' DATES CAN BE ESTIMATED:',
     +       /,'    PERCENT PROBABLE',5X,6(I6,2X),
     +       /,' NETWORK COMPLETE BY',5X,6(F6.1,2X))
       ELSE
         ILONG=U(N)
         WRITE(6,415) U(N),NAMES(1),NAMES(N)
415      FORMAT(//,10X,
     +      'THERE ARE ',I7,' DAYS ON THE CRITICAL PATH'
     +   ,/, ' BETWEEN THE FIRST EVENT *',A16,'* AND THE',
     +    ' LAST *', A16,'*.')
       ENDIF
       WRITE(6,400)
C
C   HAVE WE DONE ALL THE JOBS.
C
420   CONTINUE
      OPEN(IGNTOP,FILE='GANTT.OPT')
      WRITE(IGNTOP,520) IGNTDS,ILONG,CDATIN,(C(I),I=1,10)
520   FORMAT(A1,I7,A7,10A4)
      WRITE(6,400)
      STOP
      END
      SUBROUTINE SINC81
     +      ( CDAY,CMNTH,CYEAR,IDAY81,NOWQTR,NUMWEK )
      IMPLICIT CHARACTER (C),LOGICAL (L),DOUBLE PRECISION (D)
      CHARACTER * 7 CDATE
      CHARACTER * 3 CMONTH(12),CMNTH
      CHARACTER * 2 CDAY, CYEAR
      DIMENSION  MONTH(12), CWEEK(7), CDAT7(7)
      EQUIVALENCE (CDATE,CDAT7(1),C1STDY),(CDAT7(2),C2NDDY),
     +            (CDAT7(6),C1STYR)      ,(CDAT7(7),C2NDYR)
      DATA  CMONTH/'JAN','FEB','MAR','APR','MAY','JUN',
     +             'JUL','AUG','SEP','OCT','NOV','DEC'/,
     + MONTH/        0,     31,  59,    90,  120 , 151,
     +              181,   212, 243,   273,  304,  334/,
     +  IWKD81/4/,
     + CWEEK/ 'M','T','W','T','F','S','S'/,I4YEAR/1461/
C
C      THIS ROUTINE IS INPUT THE CHARACTER DATE (DDMMMYY)
C      AND WILL RETURN THE SPECIFIC DAY OF THE WEEK(1-7), THE
C      YEARLY QTR(1-4) AND THE NUMERIC NUMBER OF DAYS SINCE
C      1 JAN 81 (0=1JAN)
C
      WRITE(CDATE,'(A2,A3,A2)') CDAY,CMNTH,CYEAR
      IDAY = 10 * (ICHAR(C1STDY)-48) + ICHAR(C2NDDY)-48
      IYEAR = 10 * (ICHAR(C1STYR)-48) + ICHAR(C2NDYR)-48
      IF(IYEAR.LT.50) IYEAR = IYEAR + 100
      NUMYRS = 0
      IF(IYEAR.GT. 81) NUMYRS = IYEAR - 81
      NUM4YR = NUMYRS / 4
      NOWMTH = 1
      DO 100 I = 1,12
         IF(CMNTH.EQ.CMONTH(I) ) NOWMTH = I
100   CONTINUE
      ISTMTH = MONTH( NOWMTH )
      IDAY81 = NUMYRS * 365 + NUM4YR + ISTMTH + IDAY - 1
      IF((MOD(IYEAR-81,4).EQ.3).AND.(ISTMTH.GT.31))
     +                    IDAY81=IDAY81 + 1
      NUMWEK = MOD(IDAY81+IWKD81-1,7) + 1
      NOWQTR = (NOWMTH-1)/3 + 1
      RETURN
      END
      SUBROUTINE TITLE
     +   (CTITLE,CDAY,CMNTH,CYEAR,COMENT,ISCALE,IPAGE,
     +                                  IBAK,NUMWEK,IQUATR)
      IMPLICIT CHARACTER (C),LOGICAL (L),DOUBLE PRECISION (D)
      CHARACTER * 80 CTITLE(2)
      CHARACTER * 40 COMENT
      CHARACTER * 3 CMONTH(12),CMNTH,CM
      CHARACTER * 2 CDAY, CYEAR,C2YEAR,C2D,CD,CY
      DIMENSION MONTH(12), CWEEK(5), CQTR(5),CLINE(80),
     +   C1MNTH(3,12), C1YEAR(2), C1D(2)
      EQUIVALENCE (CMONTH(1),C1MNTH(1,1)), (C2YEAR,C1YEAR(1))
     +           , (C2D,C1D(1))
      DATA CBLANK/' '/,CBAR/'|'/,
     + CMONTH/'JAN','FEB','MAR','APR','MAY','JUN',
     +        'JUL','AUG','SEP','OCT','NOV','DEC'/,
     + MONTH/  31,  28,    31,    30 ,   31,    30,
     +         31,  31,    30,    31,    30,    31/,NUMDAY/84/
     + CWEEK/ 'M','T','W','T','F'/
C
      C2YEAR = CYEAR
      IF( ISCALE   .EQ. 1) THEN
      DO 95 J=1,12
        DO 95 I = 1,5
 95      CLINE((J-1)*5+I)=CWEEK(I)
        ISTDAT = IBAK + NUMDAY*(IPAGE-1)
        CALL DAY2CH(CD,CM,CY,ISTDAT,IQ,IW)
        WRITE(CTITLE(1),97)COMENT,CD,CM,CY,IPAGE
 97     FORMAT(A40,5X,'12 WEEKS FROM ',A2,A3,A2,4X,'PAGE',I2)
        C2D = CD
        DO 110 I = 1,12
110       IF(CMONTH(I) .EQ. CM) IMONTH = I
        IPOS = 1
        IF(CD .NE. '01' ) THEN
          IDAY = 10*(ICHAR(C1D(1))-48) + ICHAR(C1D(2)) - 48
          IADD = 0
          IREMAN = MONTH(IMONTH) - IDAY + 1
          IWKEND = IREMAN/7
          IWORK = IREMAN - IWKEND * 2
          IF(MOD(IREMAN-1,7)+1.GT.5) IADD = 1
          IPOS = IWORK + IADD
          IMONTH = MOD(IMONTH,12) + 1
        ENDIF
        CLINE(IPOS) = CBLANK
        DO 120 J = 1,3
120        CLINE(IPOS+J) = C1MNTH(J,IMONTH)
        WRITE(CTITLE(2),100)(CLINE(I),I=1,60)
100     FORMAT(7X,'EVENT',7X,60A1)
      ELSE IF(ISCALE.EQ.5) THEN
C
C            GENERATE THE WEEKLY SCALE
C
        DO 200 I = 1,12
```

```
200       IF(CMONTH(I).EQ.CMNTH) NOW = I
        IWKDAY = NUMWEK
        INDEX = 0
        IFULYR =  NOW + 11
        DO 400 NOWMTH = NOW, IFULYR
           I = MOD(NOWMTH-1,12) + 1
           KNTWEK = (MONTH(I) + IWKDAY) / 7
           IWKDAY = MOD(IWKDAY+MONTH(I)-1,7) + 1
           INDEX = INDEX + 1
           CLINE(INDEX) = CBAR
           DO 300 IBUILD = 1,3
300           CLINE(INDEX+IBUILD) = C1MNTH(IBUILD,I)
           INDEX = INDEX + 3
           IF(I .EQ. 1) THEN
C
C             FIGURE OUT THE YEAR
C
              ISUB = 0
              IF(NOW .EQ. 1 ) ISUB = 1
              IYEAR = 10*(ICHAR(C1YEAR(1))-48)
     +                              + ICHAR(C1YEAR(2))-48
     +                              + IPAGE - ISUB
              CALL YR2CHR( IYEAR,C1STDG,C2NDDG)
              CLINE(INDEX) = C2NDDG
              CLINE(INDEX-1) = C1STDG
           ENDIF
           IF(KNTWEK .GT. 4 ) THEN
              INDEX = INDEX + 1
              CLINE(INDEX) = CBLANK
           ENDIF
400     CONTINUE
        WRITE(CTITLE(1),450)
     +       COMENT,'01',CMNTH,IYEAR+ISUB-1,IPAGE
450     FORMAT
     +  (A40,5X,'12 MONTHS FROM ',A2,A3,I2,4X,'PAGE',I2)
        WRITE(CTITLE(2),500)(CLINE(I),I=1,INDEX)
500     FORMAT(7X,'EVENT',7X,60A1)
       ELSE IF( ISCALE .EQ. 20 ) THEN
C
C            GENERATE THE MONTHLY SCALE
C
        ISTYR = 10*(ICHAR(C1YEAR(1))-48)
     +                              + ICHAR(C1YEAR(2))-48
     +                              + (IPAGE-1) * 5
        IQTR = IQUATR
        INDEX = 0
        IFL5YR = IQTR + 19
        DO 600 NOWQTR = IQTR, IFL5YR
           I = MOD(NOWQTR-1,4) + 1
           CLINE(INDEX+1) = CBLANK
           CLINE(INDEX+2) = 'Q'
           CLINE(INDEX+3) = CHAR(I+48)
           INDEX = INDEX + 3
           IF(I .EQ. 1) THEN
C
C             FIGURE OUT THE YEAR
C
              IYEAR = ISTYR + NOWQTR/4
              CALL YR2CHR( IYEAR,C1STDG,C2NDDG)
              CLINE(INDEX) = C2NDDG
              CLINE(INDEX-1) = C1STDG
           ENDIF
600     CONTINUE
        CMNTH=CMONTH((IQTR-1)*3+1)
        WRITE(CTITLE(2),500)(CLINE(J),J=1,INDEX)
        WRITE(CTITLE(1),650)COMENT,'01',CMNTH,ISTYR,IPAGE
650     FORMAT(A40,5X,'5 YEARS FROM ',A2,A3,I2,4X,'PAGE',I2)
       ELSE IF( ISCALE .EQ. 65 ) THEN
C
C            GENERATE THE QUARTERLY SCALE
C
        IYEAR = 10*(ICHAR(C1YEAR(1))-48)
     +                        + ICHAR(C1YEAR(2))-48
     +                        + (IPAGE-1) * 15
        INDEX = 0
        I15YRS = IYEAR + 14
        DO 700 NOWYR = IYEAR, I15YRS
           CLINE(INDEX+1) = CBAR
           CALL YR2CHR( NOWYR, C1STDG, C2NDDG)
           CLINE(INDEX+2) = C1STDG
           CLINE(INDEX+3) = C2NDDG
           CLINE(INDEX+4) = CBLANK
           INDEX = INDEX + 4
700     CONTINUE
        WRITE(CTITLE(1),750)COMENT,'01','JAN',IYEAR,IPAGE
750     FORMAT(A40,5X,'15 YEARS FROM ',A2,A3,I2,4X,'PAGE',I2)
        WRITE(CTITLE(2),500)(CLINE(I),I=1,INDEX)
      ENDIF
      RETURN
      END
      SUBROUTINE YR2CHR(IYEAR,C1STDG,C2NDDG)
      IMPLICIT CHARACTER (C)
C
C           THIS ROUTINE CONVERTS A NUMERIC YEAR  TO
C           2 CHARACTER DIGITS
C
      NEWYR = MOD(IYEAR,100)
      I1STDG = NEWYR / 10
      I2NDDG = NEWYR - 10*I1STDG
      C1STDG = CHAR(I1STDG+48)
      C2NDDG = CHAR(I2NDDG+48)
      RETURN
      END
```

```
C
C
C
C   THESE ARE THE STORAGE STACKS, VALUES ARE STORED HERE
C    UNTIL THERE IS SUFFICIENT INFORMATION TO SOLVE SOME
C    OF THEM.
C
C
        SUBROUTINE ZERO
C
        COMMON/STORE/VALUE,N
        COMMON/PSHPOP/ STACK1,STACK2, TOP1, TOP2
        INTEGER TOP1,TOP2,VALUE,N,STACK1(100),STACK2(100)
C
         TOP1=0
        TOP2=0
        RETURN
        END
C
        SUBROUTINE PUSH 1
C
        COMMON/STORE/VALUE,N
        COMMON/PSHPOP/ STACK1,STACK2, TOP1, TOP2
        INTEGER TOP1,TOP2,VALUE,N,STACK1(100),STACK2(100)
C
        TOP1=TOP1+1
        STACK1(TOP1)=VALUE
        RETURN
        END
C
        SUBROUTINE PUSH 2
C
        COMMON/STORE/VALUE,N
        COMMON/PSHPOP/ STACK1,STACK2, TOP1, TOP2
        INTEGER TOP1,TOP2,VALUE,N,STACK1(100),STACK2(100)
C
        TOP2=TOP2+1
        STACK2(TOP2)=VALUE
        RETURN
        END
C
        SUBROUTINE POP1
C
        COMMON/STORE/VALUE,N
        COMMON/PSHPOP/ STACK1,STACK2, TOP1, TOP2
        INTEGER TOP1,TOP2,VALUE,N,STACK1(100),STACK2(100)
C
        VALUE =0
        IF(TOP1.EQ.0) RETURN
        VALUE=STACK1(TOP1)
        TOP1=TOP1-1
        RETURN
```

```
         END
C
         SUBROUTINE POP2
C
         COMMON/STORE/VALUE,N
         COMMON/PSHPOP/ STACK1,STACK2, TOP1, TOP2
         INTEGER TOP1,TOP2,VALUE,N,STACK1(100),STACK2(100)
C
         VALUE=0
         IF(TOP2.EQ.0) RETURN
         VALUE=STACK2(TOP2)
         TOP2=TOP2-1
         RETURN
         END
C
C
C    THIS SUBROUTINE CALCULATES U,WHICH IS THE EARLIEST TIME
C    A PARTICULAR NODE CAN BE REACHED.
C
         SUBROUTINE UEARLY
C
         COMMON/TIMES/T,U,V
         COMMON/STORE/X,N
         COMMON/MAXMIN/MAXMIN,Y,I
         INTEGER Y,I,X,N,MAXMIN(100)
         INTEGER * 2 T(100,100) ,U(100),V(100)
C
         I=0
         Y=0
450      Y=Y+1
         IF(Y.EQ.N) GO TO 500
         IF(Y.EQ.X) GO TO 450
         IF(T(Y,X).LT.0) GO TO  450
         IF(U(Y).LT.0) GO TO 475
         I=I+1
C
C    CALCULATE THE LENGTH U OF THE ARCS LEADING INTO NODE X.
C
         MAXMIN(I)=U(Y)+T(Y,X)
         GO TO 450
475      IF(I.EQ.0) RETURN
         MAXMIN(I)=0
         I=I-1
         GO TO 475
C
C    DETERMINE THE MAXIMIN U.
C
500      U(X)=MAXMIN(I)
         IF(I.NE.1) GO TO 525
         MAXMIN(I)=0
         RETURN
525      IF(U(X).LT.MAXMIN(I-1)) U(X)=MAXMIN(I-1)
```

```
C
C   SET THE MAXMIN STORAGE ARRAY TO ZERO BEFORE LEAVING
C   THE SUBROUTINE.
C
      MAXMIN(I)=0
      I=I-1
      IF(I.NE.1) GO TO 525
      MAXMIN(I)=0
      RETURN
      END
C
C   THIS CALCULATED THE VALUE FOR V WHICH IS THE LATEST TIME
C   A NODE CAN REACHED WITHOUT HOLDING UP THE SCHEDULE.
C
      SUBROUTINE VLATEST
C
      COMMON/TIMES/T,U,V
      COMMON/STORE/X,N
      COMMON/MAXMIN/MAXMIN,Y,I
      INTEGER Y,I,X,N,MAXMIN(100)
      INTEGER * 2  T(100,100) ,U(100),V(100)
C
      I=0
      Y=N+1
550   Y=Y-1
      IF(Y.EQ.1) GO TO 600
      IF(Y.EQ.X) GO TO 550
      IF(T(X,Y).LT.0) GO TO 550
      IF(V(Y).LT.0) GO TO 575
      I=I+1
C
C   CALCULATE THE LENGTH V OF THE ARCS LEADING OUT OF NIDE X.
C
      MAXMIN(I)=V(Y)-T(X,Y)
      GO TO 550
575   IF(I.EQ.0)RETURN
      MAXMIN(I)=0
      I=I-1
      GO TO 575
C
C   DETERMINE THE MINIMUM V.
C
600   V(X)=MAXMIN(I)
      IF(I.NE.1) GO TO 625
      MAXMIN(I)=0
      RETURN
625   IF(V(X).GT.MAXMIN(I-1)) V(X)=MAXMIN(I-1)
C
C   SET THE MAXMIN STORAGE ARRAY TO ZERO BEFORE LEAVING
C   THE SUBROUTINE.
C
      MAXMIN(I)=0
```

```
         I=I-1
         IF(I.NE.1) GO TO 125
         MAXMIN(I)=0
         RETURN
         END
      SUBROUTINE FNDNAM( NAMFRM, NAMTO, IFROM, ITO )
      COMMON/NAMES/NAMES(100),KNTNAM
      CHARACTER*16 NAMES,NAMFRM,NAMTO
C
C The purpose of this routine is to find the the array
C position of NAMFRM and NAMTO and return that value to
C PERTCP so it can continue to process only numeric values
C This should make PERTCP easier to use.
C
      IFROM=0
      ITO = 0
      IF( KNTNAM .GT. 0) THEN
        DO 100 I = 1, KNTNAM
           IF( IFROM.EQ. 0) THEN
             IF( NAMFRM .EQ. NAMES(I) ) THEN
               IFROM = I
             ENDIF
           ENDIF
           IF( ITO.EQ. 0 ) THEN
             IF(NAMTO .EQ. NAMES(I) ) THEN
               ITO = I
             ENDIF
           ENDIF
100      CONTINUE
         IF( IFROM .EQ. 0 ) THEN
            KNTNAM = KNTNAM + 1
            NAMES( KNTNAM ) = NAMFRM
            IFROM = KNTNAM
         ENDIF
         IF( ITO .EQ. 0 ) THEN
            KNTNAM = KNTNAM + 1
            NAMES(KNTNAM ) = NAMTO
            ITO = KNTNAM
         ENDIF
      ELSE
         KNTNAM = 2
         IFROM = 1
         NAMES( IFROM ) = NAMFRM
         ITO = 2
         NAMES( ITO ) = NAMTO
      ENDIF
      RETURN
      END
```

Appendix F: PMDSS Work-Sheets

The prototype of the PMDSS is designed for the program manager from the SPO of RW. The RW activities are documented via the activity work-sheets. They contain a description of the activity, the OPR, the estimation of the activity duration, related regulation and a lessons learned categories. The work-sheets have been derived from RW but they are generic in nature. An inexperienced program manager could use them as a tutorial instrument.

The work-sheets reside on the PMDSS-USR disk. They can and should be updated to reflect the dynamic interchange of the program. The program manager could use the work-sheets as his CYA file, to explain why the schedule is in a given state.

The RW Generic Program Work-sheets appear in alphabetical order.

3-LTR-RV is the name of the WORK-SHEET:

THREE-LETTER REVIEW (FACE-TO-FACE)

DESCRIPTION: When directed by RW or the responsible three-letter SPO Director/Deputy Director, a three-letter review (internal SRP) will be conducted. The purpose of the review is to determine adequacy/completeness of the Model Contract and Proposal Instructions.

OPR: Program Manager _EVENT DURATION_: 2weeks,3weeks,5 weeks
REFERENCES: RWOI 70-4
REMARKS/LESSONS LEARNED: None

ACQ-MNGT is the name of the WORK-SHEET:

ACQUISITION MANAGEMENT PANEL (AMP)

DESCRIPTION:A standing ASD panel with broad-based membership consisting of the best corporate experience and knowledge available. ASD/CC appoints the chairperson and members. The secretariat is appointed by ASD/AV. Members will be selected as individuals (as opposed to organizational representatives) so as to bring the widest available background and experience to the Panel. For particular meetings, the membership may be augmented by the chairperson to add specific experience. The participation of permanent representatives from AFALD and AFLC/JAG will be solicited by the AMP chairperson. This panel will make recommendations to the program manager but will in no

ACQ-PLAN is the name of the WORK-SHEET:

ACQUISITION PLAN (AP)

DESCRIPTION:Acquisition Planning means the process by which the efforts of all personnel responsible for an acquisition are coordinated and integrated through a comprehensive plan for fulfilling the agency need in a timely manner and at a reasonable cost. It includes developing the overall strategy for managing the acquisition.

OPR: Contracting Officer. _EVENT DURATION_: 30, 45, 120
REFERENCES: RWOI 70-2. For format and content, see FAR Part 7 as supplemented.
REMARKS/LESSONS LEARNED: The AP, BSP, AMP and CSP must all talk to the same acquisition methodology.

AFSCFM56 is the name of the WORK-SHEET:

AFSC FORM 56

DESCRIPTION: The form 56 is actually a short directive from AFSC telling us to start executing a PMD. Without a Form 56, the PMD is virtually useless. (We act on the 56, not the PMD)

OPR: SYSTO at HQ AFSC   _EVENT DURATION_(MIN,AVG,MAX):5,5,7 days

ASDFM117 is the name of the WORK-SHEET:

ASD FORM 117, COORDINATED AND APPROVED

DESCRIPTION: This event is simply the accomplishment of the following: coordinating the PR checklist with all the OPRs identified for the thirty (30) items listed on the checklist, sign-off by the program manager to indicate that the PR package is complete, and signature by the contracting office to indicate acceptance of the PR package. Assignment of the individuals OPRs who are to prepare the items listed on the checklist and establishment of scheduled due dates for each item will have been accomplished by an earlier event. IPR and identification of program schedule. Also, actual preparation of major PR package components (i.e., specification, PR, CDRL, and DD254) is covered by other events and is not considered to be part of this event.

OPR: Program Manager _EVENT DURATION_: One,two,four weeks
REFERENCES: ASD/RW
REMARKS/LESSONS LEARNED: None at this time.

BUS-STR is the name of the WORK-SHEET:

BUSINESS STRATEGY PANEL (BSP)

DESCRIPTION:A Panel with membership tailored to fit the value and complexity of the individual acquisition. The BSP provides assistance to the program team during early planning phase. The panel membership will include membership from program management, contracting, Comptroller representative, manufacturing, product assurance, JAG and representative from other cognizant activities if appropriate. The BSP operates as an advisory body only. No formal direction emanates from the panel.

OPR:Program Management Office & PCO _EVENT DURATION_: 15,45,55
REFERENCES: RWOI 70-4 and AFSCR 70-2.
REMARKS/LESSONS LEARNED: The BSP, AP, AMP and CSP must talk the same acquisitionn methodology. (Woe be unto anyone who doesn't follow the advice of this panel).

CNT-STR is the name of the WORK-SHEET:

CONTRACT STRATEGY PAPER (CSP)

DESCRIPTION:A matrix sheet providing concise overview of program direction, business approach, risk, type contract, schedule, source selection, funds, warranties, special clauses and prior contracts.

OPR: CO _EVENT DURATION_: to AFSC-10,30,40; to ASD-5,10,15
REFERENCES:RWOI 70-4, AFSC FAR Sup 1.601-101 and sups thereto, and ASD/PM/109 letter (1983).
REMARKS/LESSONS LEARNED: The CSP, BSP, AP and AMP must talk the same acquisition methodology.

CNTPKGDN is the name of the WORK-SHEET:

CONTRACT PACKAGE COMPLETED AND TRANSMITTED TO CONTRACTS

DESCRIPTION: This event involves the transmittal to the contracting office, of material acquired with the coordinated PR checklist (a prior event) along with any other information that is required from the program office to allow assembly of a draft contract. This added information may be in the form of Notes to Buyer and would include such information as Source List (or Sole Source Justification), GFE list and sources, delivery schedule, warranty requirements, type of contract (e.g., FFP, FPIF, CPIF, etc), any award or incentive fee information and any special clauses deemed necessary by the program office. Some of these items are coordinated on the PR checklist but have no other prescribed means by which to document them.

OPR: Program Manager _EVENT DURATION_: Two ,five, thirty days
REFERENCES: AFLC/AFSCR 57-7.
REMARKS/LESSONS LEARNED: None at this time.

COSTBASE is *the name of the* WORK-SHEET:

COST BASELINE

DESCRIPTION: The cost baseline is the financial contract between the program manager and the cost baseline approval authority (ASD/CC) and is the best estimate of the cost to accomplish the directed program. Cost baselines are three t,pes: (1) initial - tracks current estimate from most recent President's Budget or other financial point diredted by the ASD Comptroller; (2) revised - new direction, cost growth, or restructure; (3) closeout - program has reached point of technical/financial stability (near Program Management Responsibility Transfer (PMRT) or physical completion.

OPR:Program Manager _EVENT DURATION_(MIN,AVG,MAX): 1,2,4 months

REFERENCES: ASD Cost Baseline Guide

REMARKS/LESSONS LEARNED: Official Cost Baseline files are maintained in Plans and Integration Division (RWPP).

CRISP is the name of the WCRK-SHEET:

COMPUTER RESOURCES INTEGRATED SUPPORT PLAN

DESCRIPTION: The CRISP identifies organizational relationships and responsibilities of the developing, supporting, and using commands for the management, technical and support requirements for mission critical computer resources (Computer resources developed under the 800 series regulations). It should address computer resources (hardware, firmware and software) used as prime mission equipment and support equipment (including software development/maintenance tools). This task involves the development of a coordinated, signed CRISP during the early program formulation to ensure that proper computer resources planning is accomplished and that necessary requirements are documented and incorporated into the SOW, SPEC, CDRL, etc. However, the CRISP is a living document throughout the life of a system and it should be reviewed periodically and coordinated updates made as the program or requirements change.

OPR: Program Manager _EVENT DURATION_: 4/6/8 months

REFERENCES: AFR 800-14, Volumes I & II, AFSC Sup 1 to AFR 800-14, Volume 1, AFLCR 880-21(C1) Attachment 2.

REMARKS/LESSONS LEARNED: Often the CRISP is not accomplished early enough in the program to play a part in the planning exercise, hence the requirements are not incorporated into the proper program documentation, CSBS&WBS is the name of the WORK-SHEET:

ASSESS COST, SCHEDULE, BUSINESS STRATEGY
AND WORK BREAKDOWN STRUCTURE

DESCRIPTION: Prior to the New Start Review, the Program Manager needs to assess the cost, schedule, business strategy and the work breakdown structure (WBS). The Program Manager must keep in mind that this task is in preparation for the New Start Review, and, that the purpose of the NSR is to validate the requirements of the PMD and begin the development of alternatives. In order to have a meaningful NSR much preparation is required so that RW can base decisions on sound information.

OPR: Program Manager _EVENT DURATION_: 3/4/6 weeks.

REFERENCES: RWOI 20-1 (New Start Review)._REMARKS/LESSONS LEARNED_:

CTRT-RIT is the name of the WORK-SHEET:

Contract Writing

DESCRIPTION When prices, terms and conditions are in agreement, the contractual instrument must be written. The buying office forwards a request for writing to ASD/PMA. ASD/PMA writes the contract in accordance with the request and submits to AFLC/JAN for legal review. After JAN review, the buying office reviews for accuracy.

OPR: RWK _EVENT DURATION_: 10/15/20 days
REFERENCES: (1) FAR 15(2) & (3), FAR 15 as supplemented, AF DAR 1-451, ASD DAR 1-403.60(4), and FAR 15.1001 as supplemented.
LESSONS LEARNED: This action can be accomplished concurrently with price negotiation memorandum preparation.

DATAPKG is the name of the WORK-SHEET:

DATA PACKAGE PREPARATION

DESCRIPTION: An effort involving three separte activities - a Data Call, a Data Requirements review Board (DRRB) and a final review of the SOW and CDRL. The data call requests data requirements from all organizations concerned with the aquisition of a specific program to insure correlation between SOW tasks and data to be delivered. The DRRB reviews and validates the data requirements. The final review of the SOW and CDRL insures completeness of requirements for incorporation into the RFP.

OPR: ASD/RWB _EVENT DURATION_(MIN,AVG,MAX):6,8,10 weeks

REFERENCES: AFR 310-1, AFSCR 310-1, RWOI 310-1

REMARKS/LESSONS LEARNED: As Data requirements may represent a significant portion of Program Cost, a through validation of data requirements should be accomplished to avoid unneccessary costs.

DD254 is the name of the WORK-SHEET:

DD254

DESCRIPTION: The DD254 is required for all programs that involve classified data. The DD254 authorizes contractors to have access to classified data.

OPR: Program Manager _EVENT DURATION_: 2 week,4week,8 weeks

REFERENCES: DOD 5200.22M/R, Industrial Security Manual, ASD/SP *preparation instructions.*_REMARKS/LESSONS LEARNED_: None at this time.

DEV-PMP is the name of the WORK-SHEET:

DEVELOP PROGRAM MANAGEMENT PLAN

DESCRIPTION: The PMP is a detailed SPO planning document addressing all aspects of program management, such as business strategy, interfaces with other program participants, system engineering, configuration management, test, logistics and training. PMPs will be published when directed by the PMD. If a PMP is not called for in the PMD, an RW Management Plan will be prepared, following AFSCP 800-3 format. The OMP is a living document and is reviewed at least annually.

OPR: Program Manager _EVENT DURATION_(MIN,AVG,MAX):4,8,12 weeks

REFERENCES: RWOI 800-4, AFR 800-2, AFSCP 800-3

REMARKS/LESSONS LEARNED:

DEV-TEMP is the name of the WORK-SHEET:

DEVELOP TEST AND EVALUATION MASTER PLAN

DESCRIPTION: The TEMP describes how system tests will be conducted and how the results will be used to verify the stated requirements.

OPR: PM _EVENT DURATION_(MIN,AVG,MAX):4,6,8 months

REFERENCES: ASDP 80-14, RWOI 80-3

REMARKS/LESSONS LEARNED: The PM will designate an official test focal point. RWNT provides guidance and policy concerning TEMP's

DRAFTPMD is the name of the WORK-SHEET:

DRAFT PROGRAM MANAGEMENT DIRECTIVE (PMD)

DESCRIPTION: The draft PMD is a coordinated effort between the program element manager (USAF PEM) and the program manager (PM) for the purpose of outlining and initially defining the program that will eventually be officially defined in the PMD. It should also utilize user inputs to identify and specifically define requirements. It should identify source documentation if at all possible.

OPR: PM _EVENT DURATION_: 4/6/8 weeks

REFERENCES: AFR 800-2/AFSC Sup 1, Acquisition Program Management, AFSCR 27-1/ASD Sup 1, Program Direction, AFR 5000.1,2,3

REMARKS/LESSONS LEARNED: Establish close working relationship with PEM and SYSTO. They can provide advice/assistance throughout the program. Usually the PM works most of the effort, but he may receive help from some key functionals, eg., engineering. Insure that both you and the user understand HIS inputs and that the user's inputs are included.

DRAFTRFP is the name of the WORK-SHEET:

DRAFT REQUEST FOR PROPOSAL

DESCRIPTION: A Draft Request for Proposal (DRFP) is a draft version of the intended formal, final RFP. The DRFP is used to get industry feedback in cutting unnecessary requirements and overly complex elements. The DRFP procedure is recommended for all competitive acquisitions and is mandatory for competive acquisitions above $25M unless waived by ASD/PM. DRFP normally will include and may not be limited to proposed SOW, SPECs, Logistic requirements, Instruction to Offerors, direction for a preproposal briefing if appropriate and a model contract.

OPR: RWK _EVENT DURATION_(MIN,AVG,MAX):4,6,8 weeks

REFERENCES: AFSC FAR 15.405-1, ASD DAR 3-550, ASD/PMPP/017 ltr (83)
and ASD/PMPP/048 ltr (83)

REMARKS/LESSONS LEARNED: The report per ASD/DAR 3-550 b (3) normally is a very time consuming effort that can be minimized with a preproposal briefing.

DRFT-SOW is the name of the WORK-SHEET:

DRAFT STATEMENT OF WORK PREPARATION

DESCRIPTION: The Statement of Work (SOW) is a description of all work to be accomplished under the contract. The SOW preparation effort involves the combined efforts of engineering, logistics, training, test & evaluation, financial, configuration, manufacturing, quality assurance and program management.

OPR: Program Manager _EVENT DURATION_: Two/four/six months

REFERENCES: AFSCR 800-XX (DRAFT) SOW Preparation, MIL-STD-881A-Work Breakdown Structure, AFSCP 173-5-Cost/Schedule Control System.

LESSONS LEARNED: SOW paragraph numbering system should be simple and consistent. Cost Performance Reporting is tied directly to the WBS. Include in cost reporting, the levels needed for management cost/schedule visibility in each functional area. Data deliverables should be mentioned in the SOW, but to be deliverable they must be included in the contract data requirements list.

DRFTSPEC is the name of the WORK-SHEET:

DRAFT SPECIFICATION COMPLETE

DESCRIPTION: The technical specification is a statement of performance, physical and functional requirements required of a system or subsystem. The specification will identify development, qualification, test, product assurance and flight test requirements necessary to demonstrate that design requirements have been satisfied. The draft specification is written to define the conceptual performance requirements to allow early coordination with the user and to obtain comments from industry prior to preparation of the final specification.

OPR: RWE _EVENT DURATION_(MIN,AVG,MAX):2,4,6 months

REFERENCES: MIL-STD 490, MIL-STD 483, RWE Operating Procedures

REMARKS/LESSONS LEARNED: If the program is a result of technology transition or is a followon to a laboratory program then proof of concept criteria should be defined and aggreed upon between SPO, Laboratory and Engineering before draft is complete. Successful lab demonstrations should occur and requirements refined before specification is complete.

DVLP-PRM is the name of the WORK-SHEET:

DEVELOP PROGRAM SCHEDULES

DESCRIPTION: Establishment of program strategy involves the preparation of a program schedule and a Purchase Request (PR) schedule. The program schedule allows for a systematic planning and tracking of major program milestones by the program office. The purchase request allows for a complete assemblage of information illustrating the required support and preparation for a contracting action.

OPR: Program Manager _EVENT DURATION_: 1/3/4 weeks
REFERENCES: RWOI 20-1, RWOI 57-2
REMARKS/LESSONS LEARNED: None

ESTB-COM is the name of the WORK-SHEET:

ESTABLISH COMMUNICATION WITH DoD PLAYERS

DESCRIPTION: At this early point in the program, the program manager needs to determine all agencies and organizations that will use the system, interface with it, approve it, influence it or be influenced by it. The program manager must establish a working relationship with these players such that they have a consistent and appropriate influence on the development and acquisition of the system.

OPR: Program Manager _EVENT DURATION_:1/2/2.5 weeks_REFERENCES_:

REMARKS/LESSONS LEARNED: Agencies like DIA, NSA, and DCA may need to be involved in the development of many systems. If these agencies have a change in accreditation policy, a system that is delayed may be unable to fit a "grandfather clause", resulting in mandatory modifications to meet accreditation standards.

FINALCON is the name of the WORK-SHEET:

CONTRACT FINALIZATION/AWARD

DESCRIPTION: This is the process of writing, reviewing, approving, and awarding the contract.

OPR: RWK _EVENT DURATION_: 55/80/100 days b

REFERENCES: (1) FAR 15(2) & (3), FAR 15 as supplemented, AF DAR 1-451, ASD DAR 1-403.60(4), and FAR 15.1001 as supplemented.

LESSONS LEARNED: (1) Contract Writing: This action can be accomplished concurrently with price negotiation memorandum preparation. (2) Preliminary Review: Nothing. (3) Final Review & Award: Nothing. (4) Pre/Post Award Notice: Don't work yourself into a Pre-award Notice situation if you can help it. It will give the recipient a chance to protest and delay your program.

FINLSPEC is the name of the WORK-SHEET:

FINAL SPECIFICATION COMPLETE

DESCRIPTION: The technical specification is a statement of performance, physical and functional requirements required of a system of subsystem. The specificaton will identify development, qualification, test, product assurance and flight test requirements necessary to demonstrate that design requirements have been satisfied. The final Specification is required prior to the form 117 being completed. Concept Demonstration should be completed and technology ready for full scale development.

OPR: RWE _EVENT DURATION_(MIN,AVG,MAX):3,6,10 months

REFERENCES: MIL-STD-490, MIL-STD-483, RWE Operating Procedures

REMARKS/LESSONS LEARNED: If the program is a result of technology transition or is a followon to a lab program then proof of concept criteria should be defined and agreed upon between the SPO, Laboratory and Engineering before draft is complete. Successful demonstration should occur and requirements refined before specification is complete.

FNL-SOW is the name of the WORK-SHEET:

FINAL STATEMENT OF WORK PREPARATION

DESCRIPTION: The Statement of Work (SOW) is a generalized description of all work to be accomplished under the contract. The SOW preparation effort involves the combined efforts of engineering, logistics, training, test & evaluation, financial, configuration, manufacturing, quality assurance and program management.

OPR: Program Manager _EVENT DURATION_: One, two, three months
REFERENCES: AFSCR 800-XX (DRAFT) SOW Preparation, MIL-STD-881A - Work Breakdown Structure, AFSCP 173-5 - Cost/Schedule Control System.
LESSONS LEARNED: SOW paragraph numbering system should be simple and consistent. Cost Performance Reporting is tied directly to the WBS. Include in cost reporting, the levels needed for management cost/schedule visibility in each functional area. Data deliverables should be mentioned in the SOW, but to be deliverable they must be included in the contract data requirements list.

ILSP is the name of the WORK-SHEET:

INTEGRATED LOGISTICS SUPPORT PLAN (ILSP)

DESCRIPTION: The ILSP is a dynamic functional tool for developing and implementing a logistics support capability for new system/equipment acquisitions. This includes the horizontal integration of the ILS elements as well as their vertical integration into various functional aspects of the program. On non-major programs, section 9 of the PMP may replace the ILSP to reflect all ILS considerations.

OPR: PM/DPML/ILSM _EVENT DURATION_(MIN,AVG,MAX):2,3,4 months

REFERENCES: AFR 800-8, AFLC/AFSCR 800-34, RWL ILSP Preparation Guide

REMARKS/LESSONS LEARNED: The ILSp is a living document that is updated as necessary. The ILSp is used to determine inputs to the draft spec and draft SOW. For lessons learned, contact Air Force Lessons Learned Data Bank, AFALC/PTLL, ext 5-7236.

INST2OFR is the name of the WORK-SHEET:

INSTRUCTIONS TO OFFERORS

DESCRIPTION:Instructions to potential offerors to be incorporated into the RFP regarding criteria to be addressed in the technical, logistical, cost and price proposals.

OPR: The Contracting Officer. _EVENT DURATION_: 30,45,60 days
REFERENCES: AFR 70-15 and AFSCR 80-15 as supplemented.
REMARKS/LESSONS LEARNED:Evaluation Criteria constitutes the heart of the evaluation and selection process. Great emphasis should be placed in this area.

IPR is the name of the WORK-SHEET:

INITIAL PROGRAM REVIEW

DESCRIPTION: An Initial Program Review (IPR) is a briefing to present the refined program strategy developed after an approved New Start Review. This strategy includes preliminary baselines and thresholds for technical performance, schedule, and cost anticipated to complete the program.

OPR: Program Office _EVENT DURATION_(MIN,AVG,MAX):2,4,6 weeks

REFERENC S: RWOI 20-1

REMARKS/LESSONS LEARNED: NONE

IPR-PREP is the name of the WORK-SHEET:

IPR PREPARATION

DESCRIPTION: The Initial Program Review (IPR) requires much preparation. Out of the IPR will come a decision as to how to proceed with the program. The IPR is a decision review so the team must develop sound alternatives on which RW can base these decisions.

OPR: Program Manager. _EVENT DURATION_: 6/8/12 weeks.

REFERENCES: RWOI 20-1_REMARKS/LESSONS LEARNED_:

J&A-APRV is the name of the WORK-SHEET:

J&A PACKAGE APPROVAL

DESCRIPTION: The J&A package will be processed for approval based on the dollar values should below under the "Coord/Approval" block.

OPR: RWK _EVENT DURATION_: 2/7/17 weeks

REFERENCES: FAR 6 and supplements.

LESSONS LEARNED: When in doubt, seek help from PM-1.
THIS IS A TEST TO CHANGE THE NUMBER OF DAYS

JUST&APR is the name of the WORK-SHEET:

JUSTIFICATION AND APPROVAL (J&A) PREPARATION PACKAGES

DESCRIPTION: A J&A is required to be prepared when negotiations are to be conducted with contractor(s) in other than full and open competition. Following are examples of other than full and open competition: (1) Only one responsible source. (2) Industrial mobilization or experimental, developmental or research work. (3) Authorized or required by statute. (4) Unusual and compelling urgency. (5) International Agreement. (6) National Security Compromization. (7) Not in the Public Interest.

OPR: _EVENT DURATION_:
REFERENCES:
REMARKS/LESSONS LEARNED:

MOA-MOU is the name of the WORK-SHEET:

MOA/MOU

DESCRIPTION: MOAs/MOUs spell out specific relationships and responsibilities between two or more organizations. They may be needed for use of facilities, equipment responsibilities, support equipment, shared responsibilities, and working arrangement. MOAs/MOUs may be needed between different product divisions, DCAS organizations, and other DoD agencies.

OPR: Program Manager _EVENT DURATION_: 2/4/6 months
REFERENCES: RWOI 11-3, AFR 11-4, AFSC Form 216, AFSC *Sup to FAR*, Section 20, Part 7.
REMARKS/LESSONS LEARNED:Note: The above allotted time is required providing your counterpart is working with you and obtaining his functional coordinations as you are obtaining your functional coordinations.

NOT-YET is the name of the WORK-SHEET:

INFO NOT AVAILABLE YET

DESCRIPTION:

OPR: _EVENT DURATION_(MIN,AVG,MAX):

REFERENCES:

REMARKS/LESSONS LEARNED:

NSR   is the name of the WORK-SHEET:

NEW START REVIEW

DESCRIPTION: The NSR assesses the applicability of the new work effort to the overall mission of the SPO. A team is formally established, consisting of the PM and all functionals. This team assesses the new project and determines how much manpower will be needed during different phases of the program. This study is presented to the CRG by the SPO gaining the new work effort. Totla project needs and resources are validated at the NSR. The result of this NSR is an RW decision to proceed with the new work effort.

OPR: 3 letter SPO   _EVENT DURATION_(MIN,AVG,MAX):1,2,3 weeks

REFERENCES: RWOI 20-1_REMARKS/LESSONS LEARNED_:

PAD   is the name of the WORK-SHEET:

AFLC PROGRAM MANAGEMENT DIRECTIVE (PAD)

DESCRIPTION:The PAD is a guide and direction to AFLC organizations (SPO Logistics, AFALC, or ALC) for doing their responsibilities in implementing and supporting the Air Force PMD. The PAD is oriented to all logistics activities; similarly, directions in the PMD are not repeated in PAD. AFLC PAD guidance and direction varies with the program and its current status.

OPR: HQ AFLC/DCS/Acquistion Logistics (AQ)._EVENT DURATION_: 15 days

REFERENCES: AFLCR 400-1, AFR 800-2, AND AFLCP/AFSCP 800-34.

REMARKS/LESSONS LEARNED:Specific directives are cited only for empahsis within the PMD; omission of a directive reference does not relieve an organization from compliance.

PMD   is the name of the WORK-SHEET:

PROGRAM MANAGEMENT DIRECTIVE

DESCRIPTION: The Program Management Directive (PMD) outlines the management of the program and assigns responsibilities to the implementing, participating, supporting and operating commands. The PMD states review and approval levels, funding and operational constraints, and technical performance.

OPR: PEM at HQ USAF   _EVENT DURATION_: 4/26/52 weeks

REFERENCES: AFR 800-2/AFSC Sup 1, Acquisition Program Management
AFSCR 27-1/ASD Sup 1, Program Direction   AFR 5000.1,2,3

REMARKS/LESSONS LEARNED: Establish close working relationships with PEM and SYSTO. They can provide advice/assistance throughout the program. Usually the PM works most of the effort but he may receive help from some key functionals, eg., engineering. Insure that both you and the user understand HIS inputs and that the user's inputs are included.

POST-IPR is the name of the WORK-SHEET:

POST IPR COST ESTIMATES

DESCRIPTION: The post IPR cost estimate is a refinement of the yearly estimate generated by RWPE for each directed program in RW (excluding studies, FMS and level-of-effort tasks).

OPR:Program Eval Div (RWPE) _EVENT DURATION_(MIN,AVG,MAX):6,7,8 weeks

REFERENCES: ASDR 173-1, Cost Analysis Program
RW OI 20-1, Reviews (which reference the CERP)

REMARKS/LESSONS LEARNED: Essential that all data be available at initiation of estimating process to perform project in 6-8 weeks.

PR-PKG is the name of the WORK-SHEET:

PR PACKAGE ACCEPTANCE/CONTRACT PREPARATION

DESCRIPTION: A PR package is required to be prepared by the Program Management Office, reviewed and accepted by the Contracting Office. Upon acceptance of the PR package, the model or informal contract preparation commences.

OPR: Contracting Officer and RWK. _EVENT DURATION_: 2/3/4 weeks
REFERENCES: AFLCR/AFSCR 57-7 and FAR 15 as supplemented and RWOI 70-3
REMARKS/LESSONS LEARNED: The Contracting Officer/Buyer should use the checklist in ASDP 70-2 for accepting the PR package.

PRE-AWAR is the name of the WORK-SHEET:

Pre/Post Award Notice

DESCRIPTION: The unsuccessful competitors for the requirements must be notified of their unsuccessful effort by a formal Contracting Officer letter. The Contracting Officer must notify each offeror whose proposal is determined to be unacceptable or whose offer is not selected for award, unless disclosure might prejudice the Government's interest.

OPR: RWK _EVENT DURATION_: 1/1/3 days
REFERENCES: (1) FAR 15(2) & (3), FAR 15 as supplemented, AF DAR 1-451, ASD DAR 1-403.60(4), and FAR 15.1001 as supplemented.
LESSONS LEARNED: Don't work yourself into a Pre-award Notice situation if you can help it. It will give the chance to protest and delay your program.

PREL-REV is the name of the WORK-SHEET:

Preliminary Review

DESCRIPTION The contractual document and its related supporting file must be reviewed and any necessary corrections accomplished prior to submission to the prospective contractor for signature. The buying office prepares the contractual document and file and forwards same through appropriate channels for preliminary review. The review authority is directly related to the dollar threshold. This may be ASD/RWKO, ASD/PMC or AFSC/PMC.

OPR: RWK _EVENT DURATION_: 30/45/55 days
REFERENCES: (1) FAR 15(2) & (3), FAR 15 as supplemented, AF DAR 1-451, ASD DAR 1-403.60(4), and FAR 15.1001 as supplemented.
LESSONS LEARNED: Nothing.

PRGMBASE is the name of the WORK-SHEET:

PROGRAM BASELINE

DESCRIPTION: Directed in the Program Management Directive for FSD and production programs. The baseline, which os written by the program manager, is a formal agreement between the participating commands listed in the PMD. Part I outlines requirements of the program (defined in Statement of Need, Required Operational Capability, System Operational Concept); Part II describes the content of the program in terms of system readiness, operations concept, training, logistics support, test and evaluation, acquisition strategy, the latest President's Budget (PB) funding (if not available, BES figures are used and figures later updated to PB).

OPR: Program Manager _EVENT DURATION_(MIN,AVG,MAX):3,4,12 months

REFERENCES: AFR 800-25

REMARKS/LESSONS LEARNED: 1. Baselines are updated annually to conform to PB. An out-of-cycle change occurs when the baseline is determined not to be executable. 2. The schedule for preparing the baseline, releasing for coord. and review and forwarding for signature, is jointly done by the RWP focal point and the 3-ltr SPO diretor. 3.Office Program Baseline files are maintained in Plans and Integration Division, RWPP.

RFP-IFB is the name of the WORK-SHEET:

FINAL REQUEST FOR PROPOSAL (RFP)/IFB

DESCRIPTION: This task includes the incorporation of final changes to acquisition documents, writing the final RFP, reviewing the final RFP and transmitting the RFP to interested bidders. This type of solicitation is used for both sole source and competitive acquisition.

OPR: RWK _EVENT DURATION_: one week,three weeks,five weeks
REFERENCES: Far 52 and supplements
REMARKS/LESSONS LEARNED: None

SAFTYREQ is the name of the WORK-SHEET:

SAFETY REQUIREMENTS

DESCRIPTION: Saftey is divided up into four basic areas; graound safety, system safety, explosive safety, and flight safty. Ground Safety includes production techniques, facility safety, and ground laser activition safety. System safety involves monitoring and eliminating hazards associated with the system itself, and any type of testing safety. If a system contains any type of explosive of flammable material, it must pass explosive safety. If the system will interface in any way with nuclear weapons, then it must pass nuclear explosive safety. Finally, before the equipment can be test flown, the system must be proven to be flight safe.

OPR: RWS   _EVENT DURATION_(MIN,AVG,MAX):2,4,6 weeks

REFERENCES: System Safety:MIL-STD 882B, DESIGN HANDBOOK DH1-6 & 1-X
AFR 800-16, RWOI 800-11
Ground Safety: AFOSH STDS 127 & 161
Flight Safety: AFR 127-2 & AFSCP 127-2

REMARKS/LESSONS LEARNED:

SEC-CLAS is the name of the WORK-SHEET:

SECURITY CLASSIFICATION GUIDE

DESCRIPTION: This event involves both preparation and approval of guidance to program participants, both Government and contractor, who might generate classified data, software or hardware. The guidance describes the various categories of such material which might be encountered (generated) in conducting the program and it indicates the circumstances which would cause this material to require classification and the level(s) of such classification.

OPR: Program Manager and Engineer _EVENT DURATION_:1,3,6 months

REFERENCES: AFR 205-37

REMARKS/LESSONS LEARNED: Check within ASD/RW and AFWAL/AA for similar programs which might have usable SCG or one which can form the general basis for a new guide. ASD/SPI has copies of all ASD SCG's and may also assist in this step. The initial formulative period of a program is particularly susceptible to inadvertent disclosure of material which should be protected, so it is imperative to develop good interim guidance to use while the SCB is being formalized.

SOW-PREP is the name of the WORK-SHEET:

FINAL STATEMENT OF WORK PREPARATION

DESCRIPTION: The Statement of Work (SOW) is a generalized description of all work to be accomplished under the contract. The SOW preparation effort involves the combined efforts of engineering, logistics, training, test & evaluation, financial, configuration, manufacturing, quality assurance and program management.

OPR: Program Manager _EVENT DURATION_: One, two, three months

REFERENCES: AFSCR 800-XX (DRAFT) SOW Preparation, MIL-STD-881A - Work Breakdown Structure, AFSCP 173-5 - Cost/Schedule Control System.

LESSONS LEARNED: SOW paragraph numbering system should be simple and consistent. Cost Performance Reporting is tied directly to the WBS. Include in cost reporting, the levels needed for management cost/schedule visibility in each functional area. Data deliverables should be mentioned in the SOW, but to be deliverable they must be included in the contract data requirements list.

SS-ACT is the name of the WORK-SHEET:

SOURCE SELECTION ACTIVITIES

DESCRIPTION: The objective of the source selection process is to select the source (contractor) whose proposal has the highest degree of credibility and whose performance is expected to meet government requirements at an affordable cost. The source selection should be conducted in such a manner as to provide impartial, comprehensive evaluations of the competitors' proposals. During this period, several milestones must be met, including the Quick Look Briefing, Mid-term Briefing, Decision Briefing, and Source selection Authority (SSA) decision.

OPR: PM and PCO _EVENT DURATION_(MIN,AVG,MAX):4,6,12 months

REFERENCES: AFR 70-15, ASD Pamphlet 800-7, "Source Selection Handbook"

REMARKS/LESSONS LEARNED: Alert functionals in time for scheduling their personnel for the proposed source selection.

SS-PLAN is the name of the WORK-SHEET:

SOURCE SELETION PLAN

DESCRIPTION: The Source SelectionPlan is a key document for initiating and conducting the source selection. The Plan should include a system descripton, an organization breakdown, and evalualtion criteria for the source selection.

OPR: PM and PCO   _EVENT DURATION_(MIN,AVG,MAX):2,3,4 months

REFERENCES: AFR 70-15, ASD Pamphlete 800-7, ASD Supplement 1 to AFR 70-15.

REMARKS/LESSONS LEARNED:

SS-STDS is the name of the WORK-SHEET:

SOURCE SELECTION DOCUMENTATION/STANDARDS

DESCRI"TION: Preparation for Source Selection required definition of standards fpr evaluation and procedured for conducting the review and comment on contractor proposals. The program manager must also reference the Source Selection Handbook before the selection of area chiefs and item managers. Procedures and standurds will facilitate the Source Selection team`s review of each contractor involved in the selection process.

OPR: PM and PCO _EVENT DURATION_(MIN,AVG,MAX):1,2,3 months

REFERENCES: AFR 70-15, Acquisition Plan, ASD Pamphlet 800-7

REMARKS/LESSONS LEARNED:

SYNOPSES is the name of the WORK-SHEET:

SYNOPSES

DESCRIPTION: Contracting Officers shall publicize Contract actions offering competitive opportunities for contractors and subcontractors in order to increase competition, broaden industry participation in meeting Government requirements, assist small business concerns and labor surplus area concerns in obtaining contracts and subcontracts. Contracting Officers are required to synopsize pre-solicitation notices, normally associated with production efforts. An advance notice of an R&D effort seeking new sources is normally used in R&D activities. If a pre-solicitation notice is used where the cognizant acquisition activity is contemplating a sole source, special verbage is required for the notice. Response to a sole source synopses and their evaluation are required to be forwarded to AFSC with the sol source justification.

OPR: Contracting Officer _EVENT DURATION_: 35/40/45 days
REFERENCES: FAR 5 and supplements
REMARKS/LESSONS LEARNED: ASD/PM-1 has expressed a concern that closer attention will be required for synopses contemplating sole source contractors as they relate to PL 98-369, Competition in Contracting Act.

THREATAS is the name of the WORK-SHEET:

THREAT ASSESSMENT

DESCRIPTION: Threat assessments are long range estimates covering the expected life cycle of the proposed system and are produced to support the weapon system acquisition planning, programming and budgeting process. Typical threat products include studies, descriptions of foreign technology capabilities, threat scenarios, threat trend projections, system threat assessment reports (STARs), threat assessment reports (TARs), threat planning documents and threat environmental descriptions (TEDs). Limited threat information is contained in both the Using Command's Statement of Operational Need (SON) and the PMD. However, the information contained in these documents is not generally in sufficient detail to define specification requirements. This detailed information must be obtained from the Foreign Technology Division (FTD).

OPR: Program Manager _EVENT DURATION_: 14/30/90 days

REFERENCES:_REMARKS/LESSONS LEARNED_:

Bibliography

1. -----. "ICAM Decision Support System, IDSS BUILD 1, Final Technical Report, June 1982-February 1985," Prepared for: Materials Laboratory AFWAL, AFSC WPAFB, Pritsker & Associates, Inc.: (November 1984).

2. -----. ASD/ACPA Letter Titled "Staff Zenith Justification" to ASD/ADL (6 April 1984).

3. -----. Materials Laboratory Computer Data Tape from Contract F33615-82-C5080, ICAM Program Office, AFWAL/MLTC, WPAFB , (January 1985).

4. Air Force Operational Test and Evaluation Center. Volume IV Software Operational Test & Evaluation Guidelines (Software Operator-Machine Interface), -Evaluators Guide-, AFTECP 800-2. (10 November 1982).

5. Alavi, Maryam. "An Assessment of the Prototyping Approach to Information Systems Development," Communications of the ACM, 27-6: 556-563 (June 1984).

6. Alter, Steven. Decision Support Systems: Current Practice and Continuing Challenges Addison-Wesley, 1980.

7. Anderson, Fredrik. "Memorandum for Col McCauley Concerning Microcomputer Introduction Plan " Assistant Deputy for Aeronatical Equipment (8 March 1984).

8. Bailey, James E. and Pearson, Sammy W. "Development of a Tool for Measuring and Analyzing Computer User Satisfaction," Management Science, 29-5: 530-545 (May 1983).

9. Bally, Laurent and Brittan, John. "A Prototype Approach to Information System Design and Development," Information & Management 1. 21-26. North-Holland Publishing Co. (1977)

10. Baumgartner, Stanley. Systems Management, Bureau of National Affairs, INC. 1979.

11. Blanning, Robert W. "What Is Happening in DSS?," Interfaces, 13: 71-80 (5 October 1983)

12. Bobcock, J. and Worthy, R. M. "Affect Information Systems in Future Military Decision Making," Signal,38: 23-25 (October 1983).

13. Bonczek R, H. Foundations of Decision Support Systems, New York: Academic Press, 1981.

14. Buede, Dennis and Choisser, Robert. "An Aid for Evaluators of Systems Design Alternatives," Defense Management Journal, 20: (Second Quarter 1984).

15. Buede, Dennis M., Yates, Gerald and Weaver, Carl. "Concept Design of a Program Manager's Decision Support System," IEEE Transactions on Systems, Man, and Cybernetics, SMC-15-4: 457-468 (July/August 1985).

16. Calamaras, P. "The Microcomputer as a Decision Making Aid," Byte,9: 122-124 (May 1984).

17. Dauphinais, Bill and Darnell, Leonard. "Project Management: One Step at a Time," PC World: 240-250 (September 1984).

18. Davis, Gordon B. and Olson Margrethe H. Management Information Systems: Conceptual Foundations, Structure, and Development: New York, McGraw Hill, 1985.

19. Dunne, Edward J. and Klementowski, Lawrence J. "An Investigation of the Use of Network Techniques in Research and Development Management," IEEE Transactions on Engineering Mangagement, EM-29-3: 74-77 (August 1982).

20. Holsapple, Clyde and Whinston, Andrew. Integrated DSS Development Tools for Micro Computers, Purdue University, 31 Jan 84 (AD-P000 024/7).

21. Howell E.A., Yancey D.P., Miner R.J., Armstrong F.B., Stark D.W., Tobin C.D. and Standridge C.R. "IDSS Build 1 System Requirements Document," Contract # F33615-82-C-5080, Pritsker And Associates (October 1982).

22. Huber, George P. "Cognitive Style As A Basis For MIS and DSS Designs: Much Ado About Nothing?" Management Science, 29-5 567-579 (May 1983).

23. Ives, Blake, Olson, Margrethe H. and Baroudi, Jack J. "The Measurement of User Information Satisfaction," Communications of the ACM, 26-10: 785-793 (October 1983).

24. Ives, Blake and Olson, Margrethe H. "User Involvement and MIS Success: A Review of Research," Management Science, 30-5: 586-603 (May 1984).

25. Keen, P.G. and Wagner G.R. "DSS: An Executive Mind-Support System," Datamation,25-12: 117-122 (November 1979).

26. Keen, Peter G.W. "Decision Support Systems: Translating Analytic Techniques into Useful Tools," Sloan Management Review: 33-44 (Spring 1980).

27. Keen, Peter G.W. and Scott Morton, Michael S. Decision Support Systems: An Organizational Perspective, Reading MA: Addison-Wesley, 1978.

28. King, William R. and Rodriguez, Jaime I. "Evaluating Management Information Systems," MIS Quarterly: 43-51 (September 1978).

29. Kling, Rob. "The Organizational Context of User-Centered Software Designs," MIS Quarterly: 41-52 (December 1977).

30. Koble, Roger. Applications of Computers as an Aid to Decision-Making in Air Force Systems Program Offices, Unpublished MS Thesis GSM/LS/85S-20. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1985.

31. Korb, Lawrence. "Designs for Man and Machines," Journal of Electronic Defense, 7: 28-30 (March 1984).

32. Liberatore, Matthew and Titus G. "The Practice of Management Science in R&D Project Management," Management Science, 29-8: 29-38 (August 1983)

33. Light, J. "Helping Managers Manage [DSS]," Computer Electronics, 22: 52-55 (July 1984).

34. Lockett, Geoff. "Implemetation Exchange: Self-Starters in MS/OR - A Growth Area," Interfaces, 14: 59-61 (May-June 1984).

35. McCall, James A. "An Introduction to Software Quality Metrics," Software Quality Management, New York: Petrocelli Books Inc (1979).

36. Meador, C., Lawrence, Guyote, Martin J., Keen and Peter G.W. "Setting Priorities for DSS Development," MIS Quarterly: 117-129 (June 1984).

37. Meyrowitz, A. "Military Applications of Artificial Intelligence," Signal, 38: 45-48 (March 1984).

38. Mintzberg, H. "The Structure of Unstructured Decision Processes," Administrative Science Quarterly,21: 246-275 (June 1976).

39. Mitrione, Michael, Lt Col. "Integration of Decision Support Systems," Military Review, 63 52-59 (April 1983).

40. Morland, D. Verne. "Human Factors Guidelines for Terminal Interface Design," Communications of the ACM, 26-7: 484-494 (July 1983).

41. Morlan, Bruce. A Computer-Based Decision Analysis Support System MS Thesis . School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1979.

42. Morton, Michael Scott. Management Decision Systems Computer Based Support for Decision Making, Boston,Mass: Div of Research, Graduate School of Business Administratioin, Harvard University 1971.

43. Naylor, Thomas H. "Decision Support Systems or Whatever Happened to M.I.S.," Interfaces,13: 81-88 (5 October 1983).

44. Norman, Donald A. "Design Rules Based on Analyses of Human Error," Communications of the ACM, 26-4: 254-8 (April 1983).

45. Papageorgiou, John C. "Decision-making in the Year 2000," Interfaces,13: 77-86 (2 April 1983).

46. Poston, Robert M. "Software Quality Assurance Implemention," The IEEE Computer Society's Sixth Internations Computer Software and Applications Conference. 356-357. IEEE, Inc., New York 1982.

47. Prentice, Carolyn and Schutt, Harold. The Program Manager's Support System (PMSS); An Executive Overview; PMSS Directorate, Defence System Management College (1 May 1984).

48. Reimann, Bernard C. and Waren, Allan D. "User-Orientated Criteria For The Selection of DSS Software," Communications of the ACM, 28-2: 166-179 (February 1985).

49. Robey, Daniel. "Cognitive Style And DSS Design: A Comment on Huber's Paper," Management Science, 29-5:580 (May 1983).

50. Robey, Daniel and Markus M. Lynne. "Rituals in Information System Design," MIS Quarterly: 5-15 (March 1984).

51. Schutt, Harold. "Explorting Knowledge and Technology for Better Management: Disciplines for Decision Support Systems and the Program Manager's Support System," Project Manager (Jan-Feb 1983).

52. Sprague, Ralph H and Carleson Eric D. Building Effective Decision Support Systems, Englewood Cliffs NJ, Prentice Hall, 1982.

53. Sprauge, Ralph H. "A Framework for the Development of Decision Support Systems," MIS Quarterly: 1-26 (December 1980).

54. Teates, H. Bennett. "The Role of Decision Support Systems in Command and Control," Signal,37: 45-49 (September 1982).

55. Wanner, E. "The Application of Decision Aid Technology to Command and Control," Signal,38: 43-46 (March 1984).

56. Watson, Hugh J. "Decision Support Systems or What Didn't Happen with MIS," Interfaces,13: 81-88 (5 October 1983).

57. Webster, Francis. "Tools For Managing Projects," Project Management Quarterly: 46-58 (June 1982).

58. Wynne, Bayard E. "A Domination Sequence - MS/OR; DSS and the Fifth Generation," Interfaces,14: 51-58 (May-June 1984).

59. Yoder, Mark. "AMS Phase II Devices," Fm 380 to ASD/ACP (6 April 1984).

VITA

Captain Terrence W. Brotherton was born on 8 November 1954 in Seattle Washington. He received an Associates of Arts degree from the University of Florida in June 1974. That year he transferred to Florida State University, and received the degree of Bachelor of Science in Management with emphasis in Information Systems in June 1976. Upon graduation, he recieve a commission in the USAF through the ROTC program. He continued teaching an undergraduate computer course until called to active duty in September 1976.

Captain Brotherton's initial duty assignment was to the the Alternate Space Computational Center (ASDC) at Eglin AFB where he later became Chief of ASDC Software. In May 1980, he made a detour to SOS before his transfer to the Tactical Fighter Weapons Center/Studies and Analysis as a large scale war-gaming computer programmer. He accompanied this organization to Langley AFB when it was renamed the Joint Studies Group (JSG) and moved. He became the Chief of JSG Computer Programming and worked on studies ranging from the TAF future force structure to the placement of the Ground Attack Control Center in Europe. In May 1984 he entered the School of Systems and Logistics in pursuit of a Masters of Science in Systems Management.

Permanent Address: 334 Southwalk Pl
Pensacola, Fl. 32506

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS
AD-A161 690

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GSM/LSY/85S-5

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of
Systems and Logistics

6b. OFFICE SYMBOL *(If applicable)*
AFIT/LSY

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State and ZIP Code)*
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

7b. ADDRESS *(City, State and ZIP Code)*

8a. NAME OF FUNDING/SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State and ZIP Code)*

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

11. TITLE *(Include Security Classification)*
See Box 19

12. PERSONAL AUTHOR(S)
Terrence W. Brotherton, Captain, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Yr., Mo., Day)*
1985 September

15. PAGE COUNT
265

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.
09	02	
09	04	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Automated Data Processing, Software Engineering, Matematical Models, Information Systems, Systems Management, Management Information Systems

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Title: DECISION SUPPORT SYTEMS FOR ASD PROGRAM MANAGERS

Thesis Advisor: Thomas Triscari, Jr., Captain, USAF
Assistant Professor of Systems Management

Approved for public release: IAW AFR 190-1.
Lynn W [illegible] 11 Sept 85
[illegible] Development
[illegible] (ATC)
[illegible]

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. □ DTIC USERS □

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Terrence W. Brotherton, Captain, USAF

22b. TELEPHONE NUMBER *(Include Area Code)*
(904) 456-5434

22c. OFFICE SYMBOL
AFIT/LSY

EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

This thesis effort identifies the program management tasks most amenable to computerization, researches existing implementation of the identified tasks, and incorporated selected implementations with a user friendly interface on a Zenith Z-100 computer.

The thesis is a combination of reviewed literature and the demonstration of the prototype concept. The literature review concentrated on the program management environment, the application of a Decision Support System (DSS) to that environment, Information System design factors related to development of a DSS and the evaluation of Information systems. A prototyping effort ensued to insure that the system would meet the requirements of the prototype user.

The DSS prototype was demonstrated to two sub-groups of generic program managers at ASD and AFIT. Using a developed evaluation instument, they evaluated eleven qualities of the DSS. The evaluation was composed of the three subcategories of system worth, system quality, and user propensity to use the system. The DSS was favorably received by both groups of prospective users. Key words: