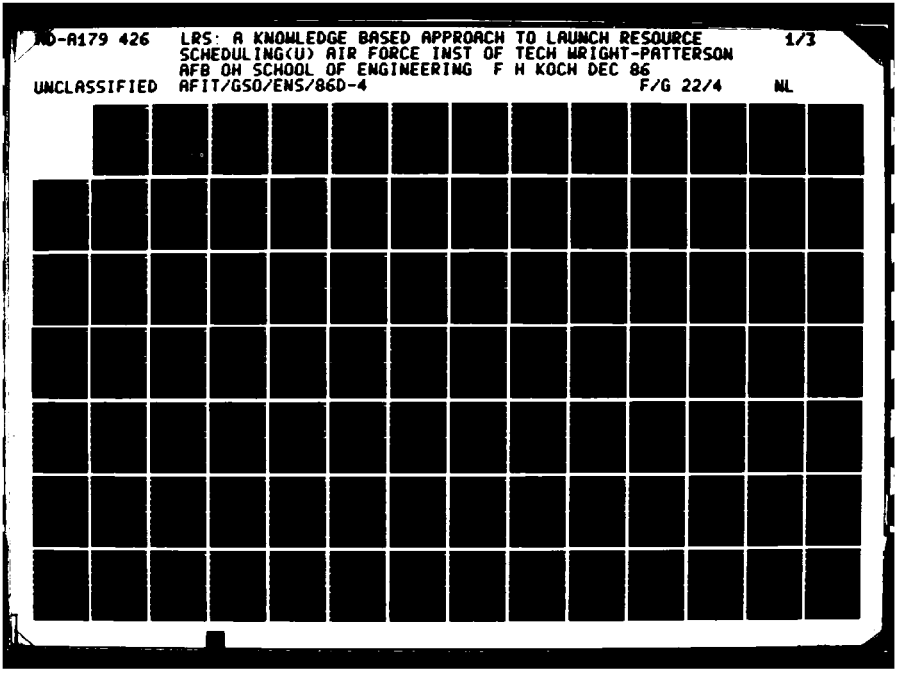
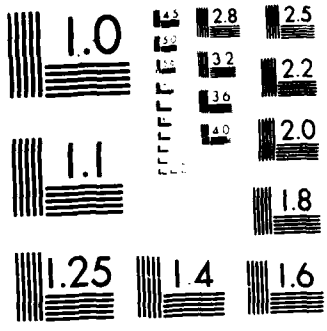
AD-A179 426 LRS: A KNOWLEDGE BASED APPROACH TO LAUNCH RESOURCE SCHEDULING(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING F H KOCH DEC 86 1/3

UNCLASSIFIED AFIT/GSO/ENS/86D-4 F/G 22/4 NL





LRS: A KNOWLEDGE BASED APPROACH TO LAUNCH RESOURCE SCHEDULING

THESIS

Fred H. Koch
Major, USAF

AFIT/GSO/ENS/86D-4

AFIT/GSO/ENS/86D-4

LRS: A KNOWLEDGE BASED APPROACH TO LAUNCH RESOURCE SCHEDULING

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Space Operations

Fred H. Koch, B.A.
Major, USAF

December 1986

Approved for public release; distribution unlimited

Table of Contents

List of Figures

List of Tables

Abstract

This work describes a knowledge based prototype launch resource scheduling tool called LRS. The tool was requested by US Space Command (J3X) for use by planners to evaluate future space launch resource requirements.

A knowledge based approach to the launch resource scheduling problem is used due to its knowledge intensive nature. The LRS prototype is implemented using a knowledge system tool called Insight 2+ on an IBM PC compatible microcomputer. LRS uses menus and explanation screens to make it simple to operate.

The code is written in IF THEN ELSE production rule format making it easy to understand and easy to update as policy and needs change. LRS uses dBase II format files for storing the required information on launch vehicle and launch pad resources, and satellite launch requirements.

LRS matches launch resources to launch requirements. A launch schedule is produced which shows how the resources meet the launch requirements. At completion of the matching process LRS provides a list of unsatisfied requirements and available resources. This list may be used by the planner to determine how well the planned launch resources meet the estimated launch requirements.

The necessary enhancements and improvements to convert the LRS prototype into an operational system are identified.

LRS: A KNOWLEDGE BASED APPROACH TO LAUNCH RESOURCE SCHEDULING

I. Introduction

Overview

This chapter provides a brief description of the problem addressed, who the user is, how the system approaches the problem, and why a knowledge based approach was selected. Greater detail on the problem and system characteristics along with how the system operates is presented in later chapters.

The Problem Addressed

The launch support required to maintain our satellite constellations is increasing every year. As the need increases so does the difficulty of determining how to meet that need. There are several new constellations planned for the near future such as NAVSTAR (Global Positioning System 29:26), MILSTAR (Military Strategic, Tactical and Relay 29:40) and various Strategic Defense Initiative options. With each new constellation comes the need to maintain that constellation. Part of maintaining the constellations is the added launch support required to replace satellites as they fail in order to maintain proper operation of the entire constellation.

U.S. Space Command's Operations section (J3), has among its responsibilities the task of looking at the future launch support required to maintain all planned satellite constellations. They need to determine if there will be sufficient launch capability to maintain the planned constellations at a reasonable performance level. Their task also encompasses looking at the feasibility of maintaining new constellations of up to 60 satellites in addition to the already planned constellations.

U.S. Space Command's Operations Plans section (J3X), requested a computer program be developed to provide an estimate of the launch support required to maintain any number of satellite constellations at a given level of performance. They also requested an examination of the feasibility of maintaining constellations composed of 30 to 60 satellites given our planned launch resources for the next 20 years (1).

This research provides a tool with which a planner can examine how launch resources can be matched with launch support needs. This tool will use the given resource capabilities and restrictions to provide a possible schedule which will use the available resources to meet the stated launch requirements. By changing the rules which govern how resources may be used, the planner can examine how these changes affect the way launch requirements are met.

A knowledge based approach is used to determine if the planned launch resources can meet the forecast launch support requirements. The system requests the user to enter known launch requirements and launch resource capabilities and restrictions. Using this information, the system provides a match of requirements to resources. This matching allows the user to determine if there are sufficient resources to meet the need. It will also allow the user to examine what affect various changes in planning decisions could have on meeting launch support requirements.

The User

The expected user will primarily be the planners at U.S. Space Command Operations Plans (J3X). They can use the tool to examine how launch requirements may be met using existing and planned launch resources. This can help them determine what type and how many launch support resources are required to meet the planned need. This tool should allow them to more easily make what if comparisons, and to examine the affect different launch systems have on meeting the planned launch requirements.

How the System Approaches the Problem

The method of matching resources to needs must consider many factors. These factors include launch vehicle boost capacity, payload/vehicle compatibility, orbit requirements, payload power requirements, payload priority, and many more.

Each factor must be considered when matching a requirement to a launch resource. Since most of these factors involve little or no computation but a matching of restrictions and characteristics, a question and answer approach can be used. This is the approach taken.

The system asks the user the appropriate questions to obtain the information required to match launch requirements to launch resources. Once all required information is obtained, a schedule which matches resources to needs is presented. This is only one possible schedule which will meet the needs. If all needs cannot be met those requirements which are unfulfilled are listed. If there is extra launch support available it is identified.

Why a Knowledge Based Approach ?

As previously mentioned much of the information required to match launch resources to requirements involves matching of known facts with little or no computation. Human schedulers tend to get better at scheduling as they learn more about the scheduling situation. This acquiring of knowledge and being able to apply the knowledge properly makes an experienced scheduler considerably better at scheduling than the average person. Thus it can be said there are experts in the scheduling area, although the range of experts is much broader than in other areas. Since the problem solution largely depends on the problem solver

knowing specific facts about the problem and how to relate these facts, a knowledge based approach is suggested.

Much work is being done to apply artificial intelligence (AI) techniques to scheduling problems. The possible solutions to a scheduling problem quickly expands as the number of items to be scheduled and the possible ways they may be scheduled increases. "The sequencing of 10 orders through 5 operations has $(10!)^5$ or more than 10^{30} possible schedules (without gaps or alternative routings) (8:3)". A computational approach which examines each possible solution to find the best solution is not feasible with such a large number of possibilities. A method of reducing the number of possible solutions is needed. AI techniques can provide this required solution pruning action. This is why AI techniques are being used to explore finding solutions to the scheduling problem.

Mark S. Fox has done extensive research in the area of applying AI techniques to scheduling problems, with emphasis on manufacturing scheduling (8; 9; 11; 12; 13). His approach is to use the satisfaction of constraints to provide the scheduling problem solution. During his research he discovered "Much of the scheduler's time (80%) is spent gathering these constraints, and then constructing a schedule that satisfies as many of the constraints as possible. Hence, the problem of scheduling can be viewed as the problem of constraint satisfaction. (9:25)"

A major consideration is how the facts and constraints are obtained by the scheduler. If the information must be obtained from scheduler sensory observation, the problem is not easily solved by a Knowledge Based approach. In the launch resource case the required information does not require sensory observation by the scheduler, but knowledge of certain facts and constraints. This knowledge can be obtained by a conversation with another individual who can provide the required information. This is referred to as the telephone test. If the information required to solve a problem can be obtained by a telephone conversation, which provides no sensory information, the problem is a candidate for solution by a knowledge based system. (18:179)

Another key factor is the type of knowledge to be applied. Common sense knowledge is very hard to represent in a computer. On the other hand factual knowledge is easily represented. Information about the size and weight capacity of many satellites is easily handled by a computer. However, dealing with information which is not clearly defined is not as easily represented. An example of such information is, people do not work well in a hot room, where the definition of what classifies a room as hot is not clearly understood. For a knowledge based approach to be applicable the information to be manipulated must be able to be represented effectively in the computer.

Since the main concern in this problem is proper application of factual knowledge, which may be obtained from description, not observation, a Knowledge Base approach is appropriate.

Summary

The problem to be addressed is one of matching launch requirements to launch resources. The anticipated user will be the operation planners at U.S. Space Command J3. The solution deals primarily with considering known facts and restrictions during the matching process. Since the problem could be solved by a conversation, not requiring observation by the problem solver, and deals primarily with facts not complex computations, a Knowledge Based approach is suggested.

II. System Requirements

Overview

This chapter defines the system requirements and presents the conclusions from the literature search. The elements which are considered essential to the computer program are provided with reasons why each element is important. Relevant points from the literature search are presented with reasons why they are important.

Definition of Terms

Some of the terms used throughout this document are defined below to provide a common understanding for their use.

Scheduling is the process of providing a time sequencing of events required to meet a desired goal. In the launch resource scheduling problem, scheduling means providing a list of which launch vehicles and launch pads are required, and when they are required to launch a given list of satellites.

Resources refers to the physical materials required to meet a desired goal. The resources considered here are the launch vehicles and launch pads available to launch the satellites.

Requirements are what must be accomplished to achieve the desired goal. Launching the satellites is a general requirement. The scheduling problem requirement is to

launch each satellite between it's desired launch time and the not later than launch time. Another requirement example is the satellite orbit. A satellite must be placed its desired orbit or it will not be able to meet the mission goals. A requirement must be met for the goal to be achieved.

A constraint is very similar to a requirement. In many places the two will be considered synonymous. The difference between a requirement and a constraint is a constraint may be relaxed, in some cases, and the goal still be accomplished, where a requirement cannot be relaxed and the goal still be met. In the launch resource scheduling problem there are few constraints which are not actually requirements. An example of a constraint which is not a requirement is the desired launch date. The satellite may be launched earlier or later than the desired launch date and still be launched. However, the not later than launch date is a requirement since if the satellite is not launched by this date it cannot be launched at all.

A restriction is similar to both a requirement and a constraint but refers to something which cannot be done rather than a condition on what can be done. The limiting orbit altitude of the shuttle launch vehicle is a restriction. The vehicle cannot reach a higher orbit.

An objective is something which is to be achieved to move toward the goal. Finding a suitable launch vehicle

which is available when needed is an objective of the launch resource problem. An objective is a sub part of the goal.

The optimal solution to the launch resource scheduling problem would be the best possible solution. If the optimal solution was found, no better solution is possible. Other solutions may be as good but none would be better.

A feasible solution on the other hand is an acceptable solution, but not necessarily the best or optimal solution. A feasible solution is a possible solution to the problem, but other solutions may be found which are better solutions and could provide a better use of the launch resources. Launching every possible satellite is a feasible solution to the launch resource scheduling problem. However, there may be a different schedule which would launch all possible satellites using one less launch vehicle. This could be a better solution to the problem. Both solutions are feasible but neither may be the optimal or best solution.

User Requirements

As mentioned in Chapter I the planners at US Space Command would like a computer program which would allow them to easily examine the effect different resources and different resource allocation policies have on meeting the planned launch requirements. Such a program would assist them in determining what type launch resources and how many of each type are needed. The following discussion will

examine the characteristics of such a program. Table I provides a comparison of the features of an operational system versus the prototype system.

The first, and perhaps one of the most important features, is ease of use. If the user cannot easily enter the launch resources and launch requirements, and obtain an easily usable output, the program will never be used. Program operation must be quick and easy to learn. This is often accomplished by using prompts during program operation to assist the user in knowing what information must be entered and when to enter it. Help instructions which may be called upon to explain the purpose of a particular function are often employed to make a program easier to operate.

The user entered information must be clearly defined. If the user is not sure what information is required to operate the program it is of little value. If the wrong information is entered the program may not operate properly or worse, may provide a resource allocation summary which is in error, but with no indication that an error has occurred.

The information required by the program must be accessible to the user. If the required launch resource or launch requirement information is not readily available, the user will seek other means to solve the problem.

The program must be easily accessible to the user. It should be available at the user's work area and be able

TABLE I

Operational vs Prototype System Capabilities

Feature	Operational System	Prototype System
Instructions	Self Contained, ask for help at any time	Program Instructions at beginning of the program, limited help during execution
Parameter knowledge	Known real life parameters stored in data base accessible by program	Hypothetical parameters stored in data base, entered by user
Parameter changes	Change any parameter at any time	Change parameters at beginning of program
Parameter saving	Save any or all parameters	Resource/Requirement data base files saved
Parameter loading	Load any or all parameters	Same data base files used every time
Assumptions	List program assumptions at any time	List program assumptions from main menu only
Resource scheduling	Start solving a problem at any stage	Must start at the beginning of problem
Solution saving	Save solution at any stage	Must be done from external program.
Schedule changes	Make schedule change and remaining schedule will be solved	No user schedule changes
Output	Print solution and all assumptions and parameters	Print solution and all assumptions and parameters
Computations	Deterministic or probabilistic	Deterministic only
Bottle necks	Locate resource bottle necks	Shows limiting resource

TABLE II

Operational vs Prototype System Capabilities (cont)

Feature	Operational System	Prototype System
Resource requirements	Establish number of each resource required to meet given requirements	No requirement estimation
Constellations	Generate launch requirements to establish/maintain constellation	Launch requirements entered
Resource assignment	Make optimal resource assignments	Feasible resource assignments
Graphics	Represent resource utilization graphically	No graphics
Resource excess	State when and what additional capability exists	States excess only at end of schedule
Resource shortage	Show limiting resources and status of remaining resources	Show limiting resource for each missed launch
Data entry	Query user for correct data in proper format	Limited user query DBase II format used

to be quickly accessed when needed. If the program requires special equipment which is not available to the user during his normal work day, it will not be used as often as one which is easily accessible. To be easily accessible the program must be quick to get operating. A program which requires a long load or set-up time is not easily accessible even if it operates on the computer which the user has at his desk.

For the program to be really helpful it must be able to provide information in a reasonable period of time. This period of time may vary depending on how much detail the planner requires. For a quick look capability the planner does not want to wait for several hours to get the information. Planners are generally not concerned with every little detail of how the needs can be met, but rather if they can or cannot be met. The program must provide a resource utilization schedule to demonstrate how the resources are meeting the need, or why they are not able to meet the need.

The program output should list the program assumptions as well as the resources and requirements used to generate the solution. This gives the planner easy access to the information used to derive the solution. It also helps keep a decision from being made using the wrong assumptions or a mismatch of resources and requirements.

Problems encountered during the resource and requirement matching process should be identified by the program. If a resource shortage exists, the shortage should be identified. If a particular group of requirements are consistently not being met, this too should be identified by the program. Such identification would allow the planner to concentrate his efforts on solving the most serious problems. Time critical shortages should be identified. That is, is there a period of time when a shortage of a

particular resource exists, even though over all the resource supply may be sufficient.

If there is an excess of a particular resource, this too should be identified. This may allow the planner to divert resources to other purposes or change the way the resources are being used to allow better allocation of the existing resources.

The launch resource and requirement information should be easy to enter. This includes initial entry as well as changes which might need to be made. The program should not require all the necessary information to be reentered every time the program is used. The common information which is used for every session should be savable for later recall with as little effort as possible. If it is necessary to change some information, it should not be necessary to reenter all the information used by the program. As little time as possible should be required to update information. It should also be easy to change information which would allow what if comparisons. If such information can be easily and quickly changed the planner can examine many more possibilities and provide a better determination of which resource combinations best meet the needs.

All the above is of little use if the planner cannot easily understand the output. The output must be able to be understood by anyone who understands the resource allocation problem. It must be able to stand alone with no

interpretation required by the person who entered the resource and requirement information.

The Literature Search

As mentioned in Chapter I, the nature of the launch resource scheduling problem and the information it deals with, lends it self to using an Artificial Intelligence (AI) approach. A literature search was conducted to find a system which would meet the program requirements listed above.

The literature search revealed the scheduling problem covers a very broad area of problems. It ranges from determining which operation to perform first in a multiple operation task, to autonomous vehicle operations. Since launch resource scheduling involves mostly resource allocation, this is the area concentrated on during the literature search.

The Job Shop Problem. The majority of AI scheduling work reported deals with manufacturing scheduling, commonly called job shop scheduling. In job shop scheduling, the scheduler must determine when each of several jobs should be started and which resources should be used to complete each job.

The job shop problem has many levels of difficulty. The simplest level is one job requiring multiple operations on one machine. The most complex problem is one with many jobs and many machines. Each job may require multiple

operations and each machine may be able to complete several different operations.

The job shop problem is solved by satisfying the requirements of each job, within the given constraints. The amount of information to be considered in the more complex problems makes the scheduling task difficult for even the skilled scheduler. Since a computer has little problem remembering to consider many different constraints, using a computer to solve the complex job shop problems is a natural consideration.

Job Shop and Launch Resource Scheduling Similarities.

The launch resource scheduling problem is similar to the job shop scheduling problem. The satellites to be launched are the jobs to be accomplished. The launch resources, vehicles and pads, are the machines to accomplish the jobs. Each launch vehicle has certain limitations as to which orbits it can achieve and what its capacities are. Each pad can only launch certain vehicles. These are resource restrictions. Each satellite can only be launched on certain vehicles. This is a job restriction. The time limits for launch vehicle and pad turn times and the satellite launch dates are the scheduling restrictions.

Systems Available. Most of the AI scheduling systems examined did not provide optimal schedules. They were content to provide a feasible solution to the problem. Only a few of the systems are operational systems, most are

prototypes which may later be developed into operational systems. A summary of the systems examined is provided in Table III and IV.

As mentioned in Chapter I, Mark S. Fox is one of the leaders in applying AI techniques to the scheduling problem as demonstrated with the ISIS system (8; 9; 11; 12; 13). ISIS is the most complex system examined during the literature search for systems which might be used to solve the launch resource scheduling problem. It provides an optimal schedule based on all the constraints known by the scheduler, and the time requirements of the jobs to be completed.

The ISIS system, like many of the job shop schedulers, could possibly be modified to provide a launch resource scheduler. However, a system which provides a launch resource schedule for planning purposes was not found. No launch resource schedulers were found, nor plans to modify an existing tool to provide such a capability.

Hardware and Software. Not all systems examined provided information on what hardware and software were used. Of the systems which did provide information, mainframe computers and Lisp machines were used the most. This appeared to be mainly due to the easy accessibility of this type equipment, the large amount of storage space and fast operation speed provided.

The earlier systems used custom software written in an AI language such as Lisp. Since these languages were only available on the large machines this contributed to the use of mainframes and Lisp machines. Later systems made use of AI tools which are much more available for the mainframe systems and Lisp machines than microcomputer systems. Many of these tools are the result of previous work completed on a mainframe or Lisp machine system which was generalized to be used as a programming tool to create new systems.

All the systems which listed the type inference engine used, employed forward chaining inference engines. As mentioned in Chapter I this is due to the large number of possible solutions in the scheduling problem. A forward chaining system works from the known facts toward a solution which is consistent with the problem constraints. In this way the number of possible solutions has little effect on the solution process. With a backward chaining system each possible solution is examined and the required supporting facts are searched for to prove the solution. In this case having a large number of possible solutions can significantly affect the solution process.

Other Possibilities. There may well be systems under development which address the launch resource scheduling problem but have not been presented in the literature. Since the area of scheduling is receiving increased emphasis in the AI research, this is very likely the case.

Summary

This chapter outlined the essential elements to be included in a computer program which would provide the capability sought by US Space Command J3X. Some of the major considerations include ease of use, program accessibility and speed of execution. After examining the available literature, no system was found which addressed the launch resource scheduling problem. The literature search revealed a trend to use mainframe and Lisp machines for scheduling problems. The more recent systems used available AI tools to reduce the development time for the prototype system. Where information on the type inference engine employed was provided a forward chaining inference engine was used in every case.

Table III

AI Scheduling Systems Examined

System Name	Researcher	Hardware	Software	Ref
RPMS	Boarnet	Lisp Mach	ZetaLisp	4
Allocate	Nygard	VAX 11/780	Turbo Pascal	30
Battle	Kaste	---	---	22
ESP	Levine	IBM/MVS	Custom	26
GENSCHED	Semeco	---	ZetaList	37
---	Ho	VAX 11/780	Franz Lisp	19
KAOS	Nachtsheim	VAX 11/780	MRS	27
---	Newman	---	---	28
---	Fox	---	---	10
---	Stockbridge	VAX 11/780	ROSS	38
ISIS	Fox	VAX 11/780	SRL	8
EXPLAIN	Cheeseman	---	---	6
---	Kohn	LMI Plus	ZetaLisp	29
TREK	Gilmore	Symbolic 3600	Lisp	16
TIMM/Tuner	Kornell	VAX	TIMM	25
---	Gadsden	VAX	SAGE	14
RPA	Rockmore	VAX 11/780	InterLisp	35
CTA	Gates	IBM 4331	VS APL	15
NONLIN	Tate	VAX 11/780	PDP-11	39
---	Arbabi	IBM 370	APL	3
---	Keirsey	Symbolics	Lisp	23
DOSS	Whalen	Mainframe	CONFIG	40
IMS	Fox	VAX 11/780	Franz Lisp	9

Table IV

AI Scheduling Systems Examined (cont)

System Name	Researcher	Hardware	Software	Ref
SPOT	Robinson	---	SIPE	34
ISA	Orciuch	VAX 11/780	OPS-5	31
NUDGE	Goldstein	---	FRL-0	17
ACS.1	Pease	PDP-11	InterLisp	32

III. Prototype System Design

Overview

This chapter presents the system design used in LRS. The system trade-off study is presented first with general considerations and specific factors used to choose the computer hardware and Knowledge System tool. The system organization and structure is presented with a discussion of how the system operates. The chapter ends with a summary of the prototype system limitations.

System Trade-off Study

The literature search revealed several trends in applying AI to scheduling problems. These included the use of mainframe or Lisp machines and later systems using knowledge base programming tools for prototype development. A few authors mentioned plans to reimplement their system on a microcomputer once the operational system was fully developed.

SLAM Prototype. The first prototype developed used the SLAM simulation language (33). Simulation provided support for the probabilistic nature of actual launch operations, but provided no special help for the scheduling problem. SLAM provided the ability to produce great detail in how the launch system operates, but the scheduling code had to be imbedded within the simulation code making updating and

changes difficult.

The amount of detail and complexity of the representation also made changing the simulation difficult. The probabilistic nature of the simulation required output analysis be performed before a decision on the ability of the launch resources to meet the launch requirements could be made. This provided much more detail and required much more time than the future planning goal required.

The simulation approach was abandoned as too complex and too time consuming to meet the goals of the planning tool desired.

OPS-5 Prototype. The decision to use a knowledge based approach was made based on the information previously presented in Chapter I. The initial plan was to implement the LRS prototype on a VAX 11/780 using OPS-5. The use of a VAX based system was discussed with Major Aderhold at US Space Command J3X (1). He stated a VAX computer was available and microcomputers were available in every office.

OPS-5 was used to solve a scheduling problem by Orciuch in creating ISA (31). OPS-5 is written in Franz Lisp. It incorporates a forward chaining inference engine and a production rule representation of knowledge (7, 5). It is a programming language rather than a tool. This allows greater flexibility in Knowledge Base design, but also requires more work by the knowledge engineer.

The version of OPS-5 available on the AFIT VAX 11/780

was developed under a government contract (7), and is available to any government agency. This makes the program available to US Space Command J3X if it is not already installed on their VAX computer.

A prototype was developed using OPS-5 with a simplified rule base. The prototype proved to operate satisfactorily. It solved the simple planning problem and provided a usable schedule. Since OPS-5 is a programming language, any system design could be implemented and any desired helps provided. However, there were several drawbacks to using OPS-5.

The OPS-5 production rules are written in Lisp format. This makes the rules difficult to understand without Lisp experience. Even with comments explaining what each rule accomplishes, rule base changes are not easy.

OPS-5 uses a last first priority in its inference engine which causes search of the newest solution path first providing a shorter solution in many cases. However, this causes problems if a sequence of rules is desired to be executed. In launch resource scheduling this occurs when a satellite is found, then a launch vehicle must be found, then a launch pad, etc. This type of sequencing proved difficult to achieve with the OPS-5 inference engine.

Since OPS-5 is a programming language not a knowledge base system tool, it requires more programming time than a knowledge base tool. This was a significant factor in

deciding which software to use for the LRS prototype system. A knowledge base tool could significantly reduce the programming work load for prototype system development.

Hardware Selection. Choosing the hardware for LRS system development was the next consideration. A VAX 11/780 is available at US Space Command and at AFIT for the development work. The only drawback to the VAX was the software available. OPS-5 is the only forward chaining language available on the system. A knowledge system tool is desirable for the reasons mentioned above.

A few of the systems reviewed during the literature search considered implementing the operational systems on microcomputers. With the increased capacity and speed of micros, more software is becoming available for them. Microcomputers are readily available at US Space Command, therefore a microcomputer based system is considered for LRS. Major Aderhold prefers the use of a microcomputer based system since planners have easier access to micros (2). There are several different micro systems available at AFIT which are compatible with the equipment available at US Space Command. If a microcomputer is to be used a suitable tool would needs to be found.

Software Selection. After examining the literature on several AI tools available for microcomputers, Insight 2+ by Level Five Research, Inc., was chosen for further consideration. It is advertised as having forward and

backward chaining capability (21). It is designed for use on the IBM PC, AT, and XT or compatible computers and operates with two floppy disk drives or a hard disk. It can be operated on a 360K system but Level Five Research recommends use of a 512K system for development work.

Making the software available to US Space Command was a remaining factor. The cost of the tool became a consideration at this point. The full development system costs $495.00 and a run time only system costs $95.00, August 1986 prices (21). The program cost would not be prohibitive if the prototype proved useful.

The System Chosen. The final decision on which hardware and software to use for the LRS prototype considered all the factors previously mentioned. A knowledge system tool was desired to reduce development time. All authors which indicated the type inference engine used in their system made use of forward chaining inference engines, therefore the tool used should provide a forward chaining inference engine. The completed prototype should be easy to use and update. The hardware has to be available both at AFIT for development work and at US Space Command for operational implementation if the system proves useful.

A microcomputer system running Insight 2+ was selected for the following reasons. US Space Command has several Zenith Z-150 computers with hard disk drives and at least 512K of memory. AFIT has several micro systems which could

support Insight 2+. The two most accessible were a Zenith Z-150 with 640K memory and two 720K floppy disk drives, and several Gemini enhanced Z-100s, each equipped with a 720K floppy disk drive and a hard disk drive. A microcomputer system is preferred by US Space Command for accessibility reasons. Insight 2+ is a knowledge system tool which operates on an available micro system. It provides the forward chaining capability desired and is the only system examined which advertised a significant forward chaining capability.

The Zenith Z-150 with two 720K floppy disk drives and 640K memory was the initial system used for development work using Insight 2+ version 1.2. As the knowledge base increased in size, the development work was moved to the Gemini enhanced Z-100 system with a hard disk drive. This significantly reduced the development time due to reduced load and compile times. The LRS system also executes significantly faster on the hard disk system due to extensive disk access required by the external program calls and data base access.

Insight 2+ Capabilities

A brief description of the capabilities of Insight 2+ is provided here, with more detail available in Appendix A.

Insight 2+ provides access to data base files using dBase II or dBase III formats. This feature allows access to large amounts of externally stored information. It also

provides a data base file creator/editor for use if dBase II or dBase III programs are not available.

It has easy screen formatting commands for help screens and data presentation. This feature makes programming help screens and easily understood output an easy task. Since it supports communication with any external program written in an executable code, routines not available under Insight 2+ can be accessed by a call to an external program, giving added flexibility.

The knowledge representation used in Insight 2+ is production rules using Level Five's own Production Rule Language (PRL) (20). PRL is based on IF THEN ELSE constructs. The rules are satisfied by comparisons of variable values. This allows descriptive names to be used for variables as long as the 60 character limit is not exceeded. Using descriptive variable names requires little other explanation to describe a rule's function. A rule from LRS will illustrate the point.

```
RULE Main Menu: Load Saved Parameters
IF Step = 1
  AND Program Option IS Load Saved Parameters
THEN Option IS Load Saved Parameters
  AND DISPLAY Load Saved Parameters Display
  AND FORGET Program Option
  AND CYCLE
```

The rules can get a little more complicated when some of the system specific commands, identified by all capitals, are used. However, these commands are designed to represent the actions to be performed and with a little exposure to

the system the commands are easily remembered and present no problem to system understanding.

This simplified rule representation is a benefit of Insight 2+. Such a representation makes system updating much easier. With judicious selection of variable names the rules can be self documenting. The clear rule representation makes adding rules at a later time much easier than with the OPS-5 program. Since ease of updating is a consideration for LRS, Insight 2+ has a clear advantage over OPS-5 in this area.

LRS Organization

The information used during the scheduling process is stored in two different places. The rules used to generate the schedule are stored within the LRS rule base and are an integral part of the LRS program code. The launch resource and launch requirement information is stored in external data base files.

The rule base information contains two categories of rules which are coded using the Insight 2+ Production Rule Language (PRL). The first category is the logic to be used to execute the program and generate a schedule. The rule shown above is such a rule. The other category includes the restrictions and constraints to be considered during the scheduling process. Such a rule is demonstrated below.

```
RULE Find Satellite Available
IF Status IS Find Satellite
```

```
    AND Sat Launched = n
    AND Sat Lch Priority <= Current Priority !(1 high)
    AND Sat Desire Lch <= Current Schedule Day
    AND Sat NLT Lch >= Current Schedule Day
  THEN All Requirements Scheduled
    AND FORGET All Requirements Scheduled
    AND Status IS Find Vehicle
    AND Schedule Day Sat Available := Current Schedule Day
    AND CYCLE
```

To change the logic used in LRS, new rules could be added or the existing rules modified. This is an uncomplicated process, but a basic knowledge of how Insight 2+ operates is required.

The help screens are called from the rules contained within the LRS knowledge base. The help screens are placed in a separate section of the knowledge base structure. This makes them easy to find and update as changes are made to program operation.

All the launch resource and launch requirement information is stored in data base files. Since Insight 2+ does not handle creation of new variables during program execution, all possible variables must be provided during programming. Rather than limit LRS to a maximum number of launch resources and requirements, the alternative is to use the data base file interface. By using the data base files to store the resources and requirements, the only limit is imposed by available disk space.

The launch resource and requirement data base files must be created before entering LRS. The files use dbase II format, which allows using the dbase II or dbase III

programs to create, edit, or add information. Insight 2+ also provides a data base creator/editor program which may be used to provide the required information.

All data base access is accomplished through external programs. These programs can be written in any executable code. Insight 2+ provides a modified Pascal language called DBPAS which has specific commands to allow easier access to the dbase II and dbase III format files. Several programs were written using DBPAS to access the data base files.

LRS Operation

One of the major requirements for the launch resource scheduling problem is to be easy to use. As mentioned in Chapter 2 menus and help screens are one way this is achieved. This is the method used in LRS. Each section of the program has a menu from which the user can choose the desired action. Help screens are provided to explain the purpose of each choice.

LRS opens with a title screen. Six program explanation screens may be viewed, or the program maybe started immediately. The program explanation screens provide a brief introduction to the purpose of the program, program operation, and the type information required. When the program is started a Main Menu is presented listing the major program actions.

The Main Menu choices are presented in Figure 1. Each selection presents a sub menu from which the desired action

is chosen. To illustrate, examine the View Information choice from the Main Menu Figure 1. When this is chosen the View Information Menu is displayed Figure 2. From this menu the user may chose to view the program assumptions, view all data base files, view any individual data base file, or return to the Main Menu. Each menu choice has similar sub menu choices to guide the user.

The menu item selection is accomplished by positioning a pointer in front of the desired choice using the arrow keys or the space bar. This feature is part of the Insight 2+ program design to choose values for object attributes.

No scheduling information is requested until the user chooses to generate a schedule. At this point LRS requests some user specific information. This provides identifying information on the schedule output to include name, date, and a line of comment. The start and end schedule day are requested after the user data is obtained. Once all this information is entered the scheduling process of matching launch requirements to launch resources begins.

The user is not asked to provide any resource or requirement information during LRS execution. All this information is stored in the external data base files, and must be set-up prior to entering LRS.

The user can specify where the output should be displayed. The schedule generated can be sent to the screen or sent to a printer. In either case the schedule

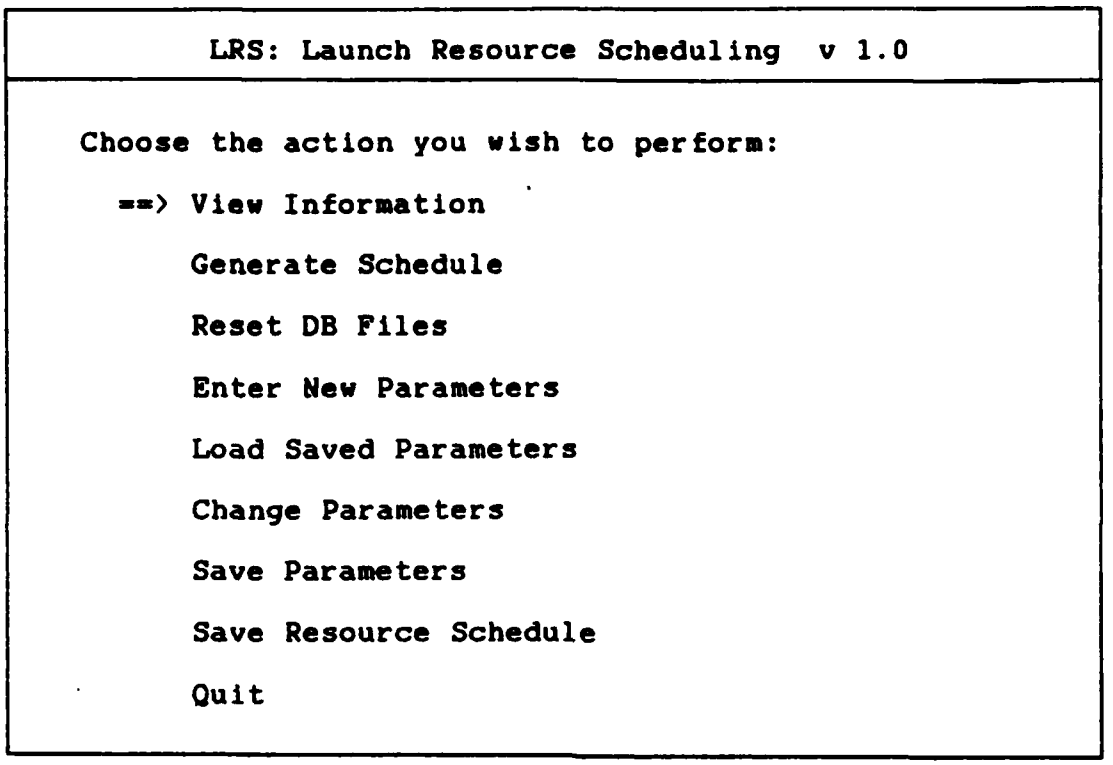


Figure 1. LRS Main Menu Display

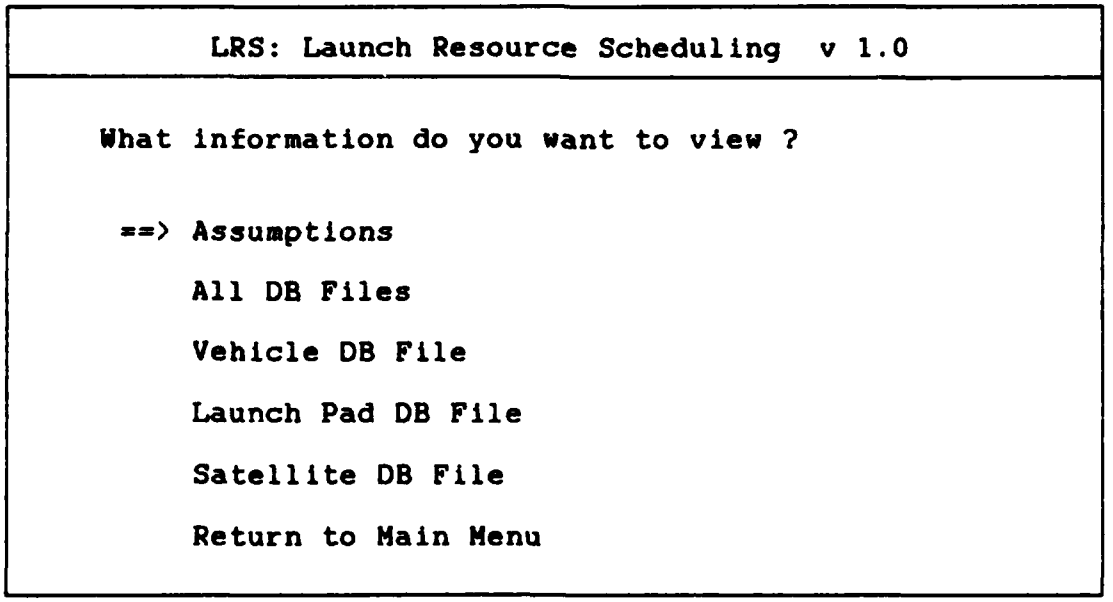


Figure 2. View Information Menu

information is also sent to a disk file which can be saved for future reference or printout. The schedule information is sent to the selected device as the schedule is generated. The matching of requirements to resources continues until all requirements are scheduled or the end schedule day is reached.

It is possible to edit data base files from within an Insight 2+ Knowledge Base program. This feature would be added to the operational version of LRS but is not included in the prototype.

Program Flow

The general program flow is cyclical in nature. The Main Menu is the starting and ending point from which all operations branch. After a choice is made, that operation is executed with return to the Main Menu after completion. The general program flow is shown in Figures 3 to 7.

The Scheduling Process. The scheduling process looks for an available satellite. Once a satellite is found a suitable launch vehicle is sought. After a launch vehicle is selected, a compatible launch pad is located. If all this is accomplished before the not later than (NLT) launch date of the satellite, the satellite is listed as launched. The satellite name and launch day are displayed with the vehicle and pad used to accomplish the launch. The satellite is marked as launched and the vehicle and pad next

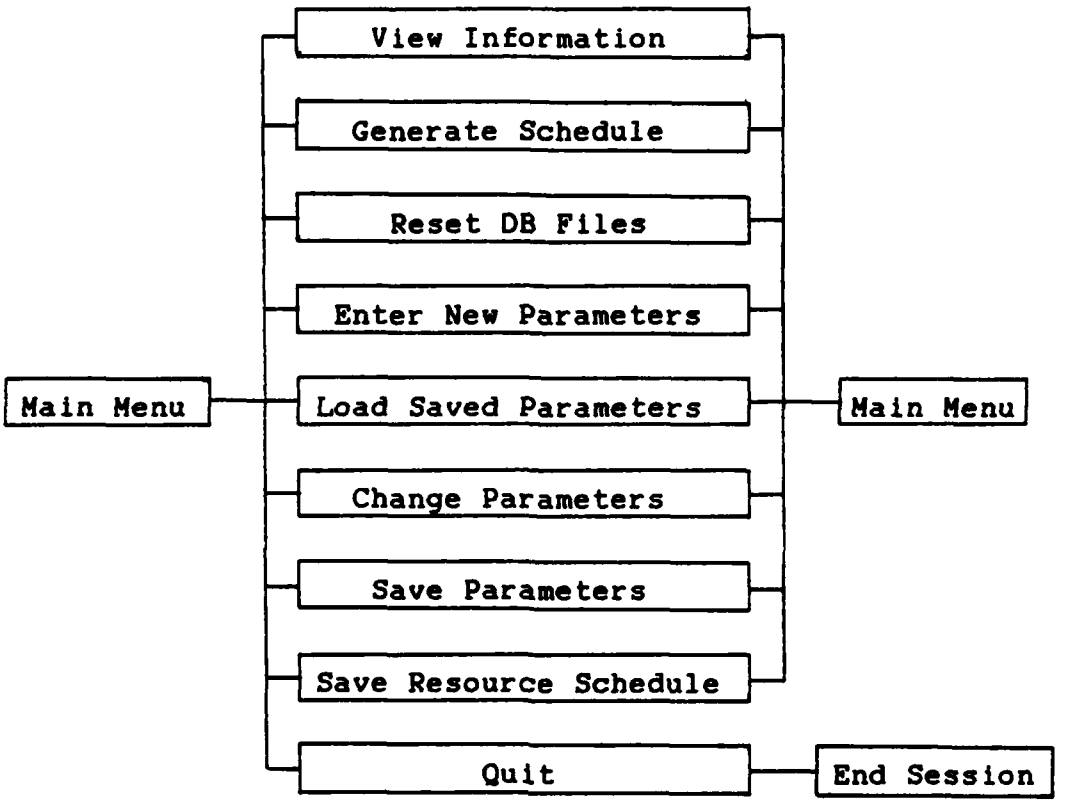


Figure 3. General Program Flow

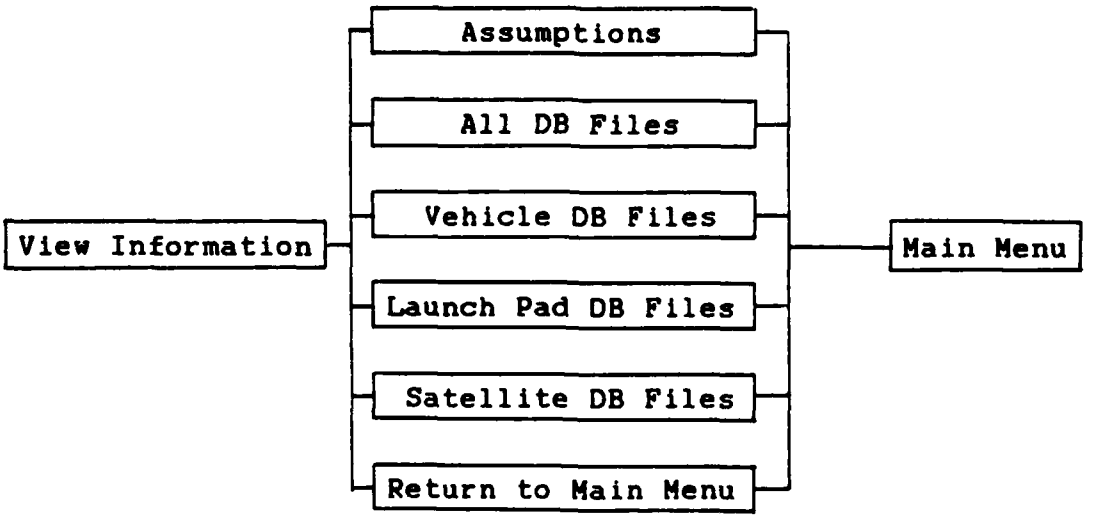


Figure 4. View Information Option Program Flow

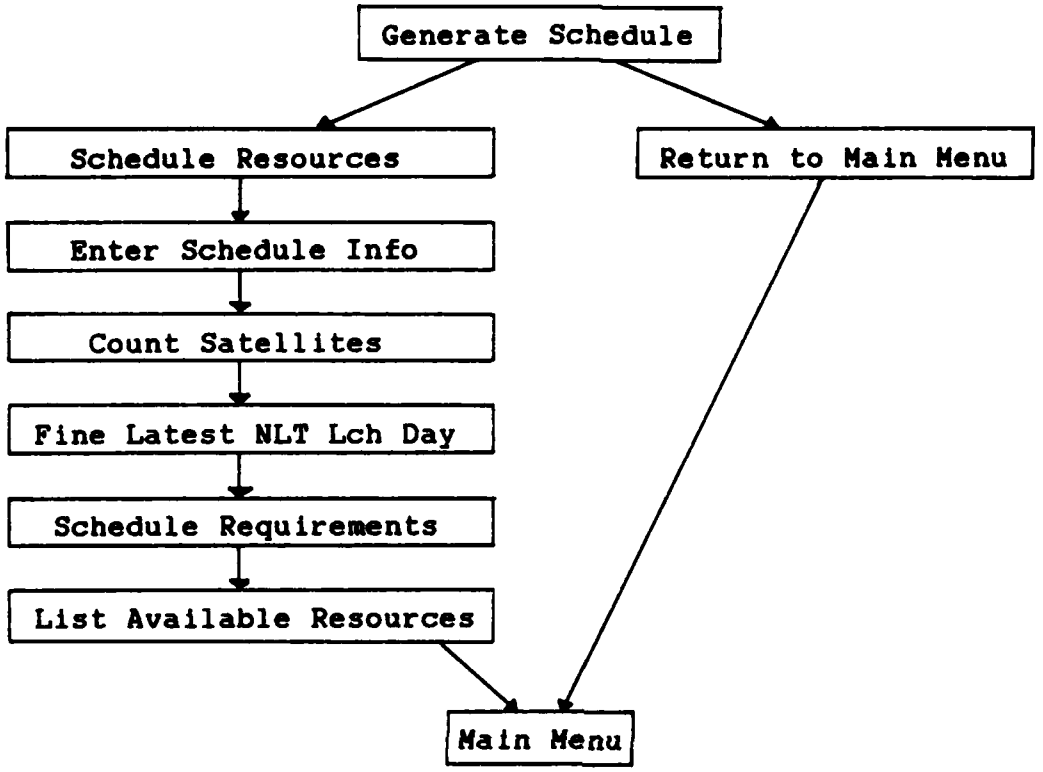


Figure 5. Generate Schedule Option Program Flow

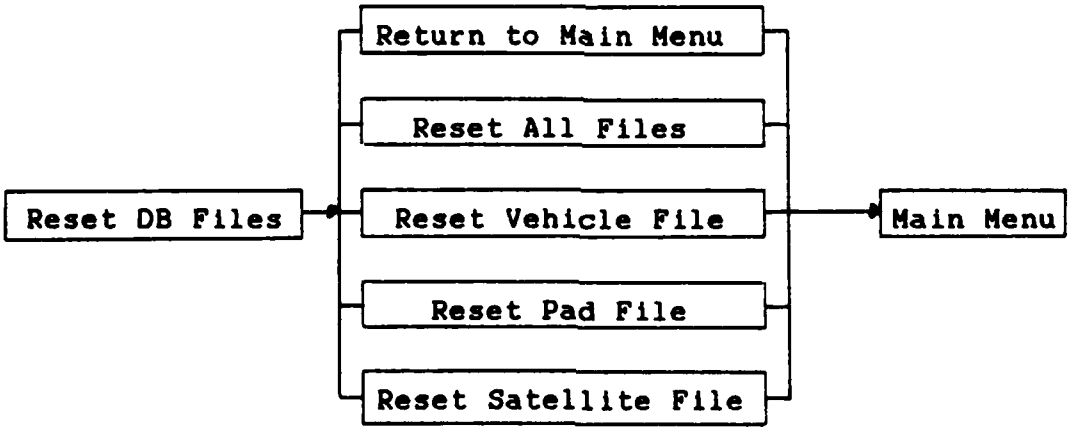


Figure 6. Reset DB Files Option Program Flow

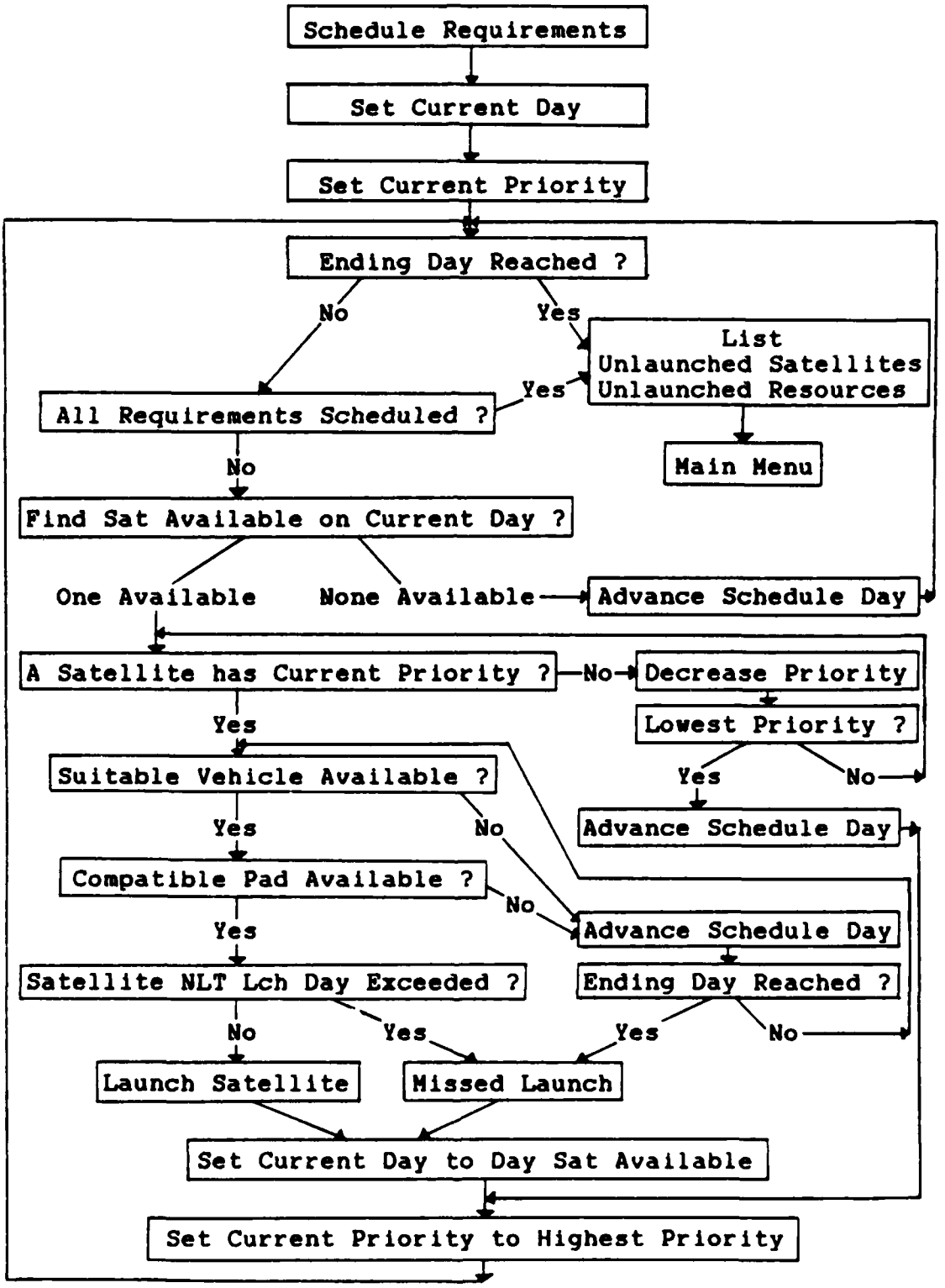


Figure 7. Schedule Requirements Option Program Flow

available dates are computed and updated in the data base.

In cases where the NLT launch day for the satellite is exceeded by the time all launch resources are available, the satellite is marked as missing its launch, an appropriate message is displayed, and another satellite is considered.

The scheduling cycle ends when no more satellites are available for launch, or the ending schedule day is reached. The details of each search process is presented below.

Satellite Selection. The satellite selection process begins on the start schedule day. The first action is to establish the latest NLT launch day for the satellites in the data base. This allows stopping the scheduling process when all available satellites have been considered instead of stepping through until the last schedule day is reached. Next the satellite data base file is searched for a satellite with a desired launch day less than or equal to the start schedule day. When a satellite is found whose NLT launch day is not exceeded its launch priority is checked. If it is not the highest priority launch the search continues for a higher priority satellite. If no higher priority satellite is found the priority is reduced until a satellite with that priority is found.

Vehicle Selection. With a satellite identified for launch, the next step is to locate a suitable launch vehicle. The vehicle data base file is searched for a vehicle which is available on the current day. If none is

available the schedule day is advanced until a vehicle is found or the ending schedule day is reached.

Pad Selection. With a suitable vehicle found the next step is to match the vehicle to a compatible launch pad. The pad data base file is searched for a pad which is available on the current schedule day. The schedule day is advanced until a compatible pad is found or the ending schedule day is reached.

Satellite Launch. Assuming a suitable vehicle and compatible pad have been located, the current schedule day is checked against the NLT launch day for the satellite. If this day has not been exceeded the satellite is listed as launched on the current schedule day in the suitable vehicle from the compatible pad. The satellite is marked as launched and the vehicle and pad next available times are updated. The current schedule day is returned to the day when the satellite was first found to be available and the scheduling process continues. It is necessary to return to the same schedule day the satellite was found to prevent a satellite being skipped due to the schedule day being advanced because a vehicle or pad is not available the same day the satellite is available.

Satellite Missed Launch. If the ending schedule day is reached before all required resources are located, the satellite is marked as missing its launch, and it is listed as missing its launch with the reason why the launch was

missed. The current schedule day is reset to the day the satellite was available and the search for another satellite begins. This does not mean the satellite could never be launched, it only shows there were not sufficient resources to launch this satellite during the specified schedule period given the conditions established for the schedule.

Data Base Files. The data base files are updated as scheduling progresses. This allows a schedule to be continued during a later session. The resource files are updated with the resource next available times as computed during the scheduling process and the satellites launched or missing launch are still marked. If a new schedule is desired, the Reset DB Files option resets all or an individual data base file to the initial state before any schedule matching was attempted.

The data base files may also be updated manually. The data base editor of Insight 2+ may be used to change the information in any record or to add additional records to the files. In this way the user can create any launch resource and launch requirement combination desired. The record format must be maintained the same as the original data base files or a program error will occur. If different information is desired to be stored in the data base files, the file access routines used in LRS must be changed to access the new file structure. The information currently stored in the data base files and which information is used

by the LRS prototype is shown in Tables V and VI. See the LRS User Manual, Appendix D, for more detail on file information.

How Program Features Are Implemented

Menu selections are obtained by creating an object with attributes which match the menu choices. By selecting the proper attribute value for the object, the user makes the desired menu selection. The display screen is generated by Insight 2+ during the attempt to determine the status of each object attribute. The wording used to request user input may be generated automatically by Insight 2+ or specified by the knowledge engineer. In LRS all questions are specified by the knowledge engineer.

The sequencing of knowledge base rules is accomplished using the forward chaining capability of Insight 2+. Each group of rules which needs to be executed at a specific time are given a step number or a status. The step or status is set to the proper value to cause execution of the rules at the proper time. Step numbers are used for major portions of the program such as Main Menu operation and each selection from the Main Menu. Status is used for the rules under each sub menu selection. Rules with no step number or status specified can be attempted at any time. Backward chaining is used to simulate subroutine calls during program execution. Since the Insight 2+ inference engine is primarily a backward chaining engine, placing a condition in

TABLE V

LRS Data Base File Information

Satellite Data Base File:

File Information	Value Stored in File
Satellite Name	Designation of Satellite
Constellation Part Of	Constellation Name
* Weight	Number of Pounds
* Size	Number of Cubic Feet
* Orbit Altitude	Number of Miles
* Launch Vehicle 1	Vehicle Type
* Launch Vehicle 2	Vehicle Type
* Launch Priority	Number 1 to 9. 1 Highest
Launch Restrictions	Comment Line
Classified	T or F
Launch with Classified Payload	T or F
* Desired Launch Day	Schedule Day Number
Day Available for Launch	Schedule Day Number
* Not Later Than Launch Day	Schedule Day Number
Type Orbit Required	Orbit Category
Orbit Inclination	Number of Degrees
* Launched	n, m, x or 1

* indicates currently used in scheduling logic

TABLE VI

LRS Data Base File Information (cont)

Vehicle Data Base File:

File Information	Value Stored in File
Vehicle Name	Designation of Vehicle
* Vehicle Type	Vehicle Category
* Boost Capability	Maximum Payload Weight
* Payload Size	Maximum Payload Cu Ft
* Orbit Altitude Capability	Maximum Orbit Altitude
* Launch Facility	Name of Facility
Maximum Mission Duration	Number of Days
* Vehicle Turn Time	Number of Days After Lch
* When First Available for Launch	Schedule Day Number
* When Next Available for Launch	Schedule Day Number

Launch Pad Data:

Pad Name	Designation of Pad
* Vehicle Type Launched by Pad	Vehicle Type
* Launch Facility	Facility Name
* Pad Turn Time	Days After Launch
* Pad First Available for Launch	Schedule Day Number
* Pad Next Available for Launch	Schedule Day Number

* indicates currently used in Scheduling logic

a rule which is the conclusion of another rule causes the inference engine to seek satisfaction of the rules which can provide the status of the required condition. When a conclusion is reached, program control returns to the rule it was originally seeking to satisfy. By selective use of the FORGET command rules providing the condition conclusions may be only executed once during program operation, or each time a rule is called.

This is demonstrated in the LRS rule shown below.

```
RULE Enter Schedule Info
IF Decision IS Schedule Resources
  AND User Info Entered
  AND Get Printer Status
  AND Schedule Information Received
THEN Schedule Info Entered
  AND Current Schedule Day := Start Schedule Day
  AND Next Available Lch Day := Ending Schedule Day + 1
```

The conditions; User Info Entered, Get Printer Status, and Schedule Information Received are all conclusions of other rules in the knowledge base. When the Enter Schedule Info rule is called the inference engine checks the status of each of these conclusions. If they are unknown, the rules which can provide the status of the desired conclusion are attempted. Once a conclusion is established it is remembered and no rule which provides that conclusion is attempted again until the conclusion is forgotten with the FORGET command. Once a conclusion is forgotten its status becomes unknown, and rules which provide the status of that conclusion will again be considered when the conclusion is required.

System Limitations

The LRS prototype has several limitations which should be mentioned. The limitations are due to development time constraints rather than Insight 2+ tool limits. All the limitations mentioned could be eliminated with additional development time devoted to the LRS system.

The LRS prototype does not schedule more than one satellite per launch vehicle. Some of the required rules to accomplish this were developed but all rules required to implement this were not completed.

The scheduling process must use the information stored in the data base files. There is no provision to allow the user to preplan required launches and have LRS schedule around these known launch times. This may be simulated using the launch priority and careful selection of availability day, but there is no easy way to assign a satellite to a specific vehicle for launch on a specific day.

There is no provision to allow the user to interrupt the scheduling process to specify certain desired matches of launch resources to launch requirements. All resource matching must be done by the LRS scheduler.

The schedule produced by the LRS prototype is not an optimal schedule. It might not even be a good schedule compared to what a human scheduler could produce. However,

it will always be a feasible schedule. All the matching will conform to the launch resource and requirements entered in the data base and with the scheduling rules in the Knowledge Base.

Summary

The Zenith Z-150 and Z-100 microcomputers were selected as the development hardware since they are available for development work and at US Space Command. The Insight 2+ knowledge system tool was chosen for prototype development. It provides a forward chaining capability and has an easily understood program structure making update easier to perform.

LRS program execution is cyclical due to the repeating nature of the launch requirement scheduling process. Both forward and backward chaining is used during LRS execution. The LRS system stores the logic rules within the program code and stores the launch resources and requirements in external data base files.

LRS uses menus and help screens to make the program simpler to learn and operate. All actions are selected from these menus. The schedule may be directed to the screen or a printer during program execution and a disk file of the schedule may be saved for later display or printing. The data base files can be edited using the Insight 2+ data base editor or the dbase II or dbase III programs.

LRS program flow is cyclical in nature. All operations start and end at the Main Menu. The scheduling process consists of finding an available satellite, then a suitable launch vehicle, then a compatible launch pad. The satellite is listed as launched if the NLT launch day is not exceeded by the time all resources are available. If all required resources are not available to launch a satellite by the ending schedule day, the satellite is listed as missing its launch and another satellite considered for launch.

The LRS prototype system has several limitations due to limited prototyping time. It only schedules one satellite per vehicle, it does not allow user scheduling, it does not provide an optimal schedule and not all features are implemented internal to the program. All the LRS prototype limitations could be eliminated if more development time is allowed.

IV. Test and Evaluation

Overview

This chapter will present the test procedure used for LRS. The information used during testing is provided with the test results. A brief explanation of the purpose of each test run is given with possible causes for any test failures.

Test Procedure

LRS matches resources to requirements. During the matching process a schedule is generated. After all resource and requirement matching is completed a list of requirements not met and resources still available is provided for evaluation by the planner. To test LRS, fourteen hypothetical scheduling problems were used. Each test case generates a different requirement on the matching rules within LRS.

The test cases most likely do not cover all possible problems which may be encountered during the process of matching resources to requirements. The tests do provide an overview of the major problems which may be encountered. Each test is designed to check the operation of one or more LRS program functions and to check the operation of one or more of the scheduling rules. The successful completion of all tests will verify the proper operation of all LRS

program functions and all scheduling rules in the knowledge base.

To provide the proper test cases the requirements and resources available needed to be specially generated. Table VII provides the meaning of the symbols used in the test data base information Tables VIII to XVIII. The information presented in these tables was entered into the LRS data base files for the test case indicated.

TABLE VII

Test Information Entered Symbol Explanations

Name	Name used for resource or requirement
Wt	Satellite Weight or Vehicle Boost Capability
Sz	Satellite Size or Vehicle Payload Capacity
Alt	Satellite Orbit Altitude, Vehicle Orbit Capability
Lv1	Satellite Launch Vehicle 1, Vehicle & Pad Type
Lv2	Satellite Launch Vehicle 2
LP	Satellite Launch Priority (1 High, 9 Low)
Fa	Satellite Desired Launch Day Vehicle & Pad First Available Day
Na	Day Next Available for Launch
NLT	Satellite Not Later Than Launch Day
Lch	Launch Status (n not launched, 1 launched)
LF	Launch Facility
TT	Turn Time

The information presented in Table VIII will be explained to demonstrate how the tables are read. The information is for tests A and B. The start schedule day entered was 1 and the end day entered was 8. The top group of data provides the satellite information available to LRS.

The first satellite record contained Sat 1 as the satellite name. The satellite weight was 20,000 pounds with a size of 100 cubic feet. It required an orbit altitude of 300 miles. The satellite could be launched on a shuttle vehicle or a t2 type vehicle. It has a launch priority of 3, 1 is the highest and 9 the lowest. The desired launch day is day 4 with a not later than launch day of 9. The satellite is not launched indicated by the n. An m would indicate the satellite is marked as loaded for launch, an x indicates it missed the launch day, and an l indicates it was launched.

The Vehicle information is presented in the middle group of data. The first record in the vehicle data base file has a vehicle name of Shuttle 1. It has a boost capability of 100,000 pounds and a payload bay size of 300 cubic feet. It can reach an orbit altitude of 350 miles. It is a shuttle type vehicle which is first available on schedule day 1 and has not been used so it is next available also on day 1. It is located at Kennedy and has a turn time after launch of 30 days.

The last group of data is for the launch pads. The

first record in the pad data base contains the name Pad 1. This is a shuttle type vehicle launch pad which is first available on day 1 and next available also on day 1 showing it is not in use or recently used. The pad is located at Kennedy and has a turn time after launch of 15 days.

TABLE VIII

Test A & B Information Entered

Start Day 1 End Day 8

Name	Wt	Sz	Alt	Lv1	Lv2	LP	Fa	Na	NLT	Lch	LF	TT
Sat 1	20	100	300	sht	t2	3	4		9	n		
Sat 2	3	100	250	sht	t3	1	4		10	n		
Sat 3	20	100	600	alv		3	1		5	n		
Sat 4	25	100	500	sht	alv	1	5		7	n		
Sht 1	100	300	350	sht			1	1			Ken	30
Sht 2	100	350	350	sht			4	4			Ken	45
Alv 1	20	100	600	alv			1	1			VBG	99
Alv 2	30	150	600	alv			4	4			VBG	99
Pad 1				sht			1	1			Ken	15
Pad 2				sht			4	4			Ken	15
Pad 3				alv			1	1			VBG	10
Pad 4				alv			4	4			VBG	10

It is possible to generate more than one schedule during each LRS session. A first run describes running a schedule for the first time after loading LRS. Certain actions are not required if the schedule is generated on the second or later run in the session. This includes items as asking the user's name, and the date. This information will stay the same for the entire session and need not be requested before each schedule generation process.

TABLE VIV

Test C & D Information Entered

Start Day 1 End Day 8

Name	Wt	Sz	Alt	Lv1	Lv2	LP	Fa	Na	NLT	Lch	LF	TT
Sat 1	20	100	300	sht	t2	3	4		9	1		
Sat 2	3	100	250	sht	t3	1	4		10	1		
Sat 3	20	100	600	alv		3	1		5	1		
Sat 4	25	100	500	sht	alv	1	5		7	1		
Sht 1	100	300	350	sht			1	1			Ken	30
Sht 2	100	350	350	sht			4	4			Ken	45
Alv 1	20	100	600	alv			1	1			VBG	99
Alv 2	30	150	600	alv			4	4			VBG	99
Pad 1				sht			1	1			Ken	15
Pad 2				sht			4	4			Ken	15
Pad 3				alv			1	1			VBG	10
Pad 4				alv			4	4			VBG	10

TABLE X

Test E Information Entered

Start Day 1 End Day 8

Name	Wt	Sz	Alt	Lv1	Lv2	LP	Fa	Na	NLT	Lch	LF	TT
Sat 1	20	100	300	sht	t2	1	3		9	n		
Sat 2	3	100	250	sht	t3	1	3		10	n		
Sat 3	35	100	600	alv		3	1		5	n		
Sat 4	25	200	500	sht	alv	1	5		7	n		
Sht 1	100	300	350	sht			1	1			Ken	30
Sht 2	100	350	350	sht			3	3			Ken	45
Alv 1	20	100	600	alv			1	1			VBG	99
Alv 2	30	150	600	alv			4	4			VBG	99
Pad 1				sht			1	1			Ken	15
Pad 2				sht			4	4			Ken	15
Pad 3				alv			1	1			VBG	10
Pad 4				alv			4	4			VBG	10

TABLE XI

Test F Information Entered

Start Day 1 End Day 8

Name	Wt	Sz	Alt	Lv1	Lv2	LP	Fa	Na	NLT	Lch	LF	TT
Sat 1	20	100	400	sht	t2	9	3		9	n		
Sat 2	3	100	250	sht	t3	9	3		10	n		
Sat 3	20	100	600	t3		1	1		5	n		
Sat 4	20	100	500	sht	alv	1	5		7	n		
Sht 1	100	300	350	sht			1	1			Ken	30
Sht 2	100	350	350	sht			3	3			Ken	45
Alv 1	20	100	600	alv			1	1			Ken	99
Alv 2	30	150	600	alv			4	4			VBG	99
Pad 1				sht			7	7			Ken	15
Pad 2				sht			4	4			Ken	15
Pad 3				alv			1	1			VBG	10
Pad 4				alv			4	4			VBG	10

TABLE XII

Test G Information Entered

Start Day 1 End Day 8

Name	Wt	Sz	Alt	Lv1	Lv2	LP	Fa	Na	NLT	Lch	LF	TT
Sat 1	20	100	300	sht	t2	1	1		9	n		
Sat 2	3	100	250	sht	t3	1	1		10	n		
Sat 3	20	100	600	alv		1	1		5	n		
Sat 4	20	100	500	sht	alv	1	5		7	n		
Sht 1	100	300	350	sht			1	99			Ken	30
Sht 2	100	350	350	sht			4	99			Ken	45
Alv 1	20	100	600	alv			1	99			VBG	99
Alv 2	30	150	600	alv			4	99			VBG	99
Pad 1				sht			1	99			Ken	15
Pad 2				sht			4	99			Ken	15
Pad 3				alv			1	99			VBG	10
Pad 4				alv			4	99			VBG	10

TABLE XIII

Test H Information Entered

Start Day 1 End Day 8

Name	Wt	Sz	Alt	Lv1	Lv2	LP	Fa	Na	NLT	Lch	LF	TT
Sat 1	20	100	300	sht	t2	1	1		9	n		
Sat 2	3	100	250	sht	t3	1	1		10	n		
Sat 3	20	100	600	alv		1	1		5	n		
Sat 4	20	100	500	sht	alv	1	5		7	n		
Sht 1	100	300	350	sht			1	1			Ken	30
Sht 2	100	350	350	sht			3	3			Ken	45
Alv 1	20	100	600	alv			1	1			VBG	99
Alv 2	30	150	600	alv			4	4			VBG	99
Pad 1				sht			1	99			Ken	15
Pad 2				sht			4	99			Ken	15
Pad 3				alv			1	99			VBG	10
Pad 4				alv			4	99			VBG	10

TABLE XIV

Test I & J Information Entered

Test I Start Day 1 End Day 1

Test J Start Day 1 End Day 5

Name	Wt	Sz	Alt	Lv1	Lv2	LP	Fa	Na	NLT	Lch	LF	TT
Sat 1	20	100	300	sht	t2	1	2		2	n		
Sat 2	3	100	250	sht	t3	1	3		6	n		
Sat 3	20	100	600	alv		1	6		8	n		
Sat 4	20	100	500	sht	alv	1	5		7	n		
Sht 1	100	300	350	sht			6	6			Ken	30
Sht 2	100	350	350	sht			6	6			Ken	45
Alv 1	20	100	600	alv			5	5			VBG	99
Alv 2	30	150	600	alv			6	6			VBG	99
Pad 1				sht			5	5			Ken	15
Pad 2				sht			5	5			Ken	15
Pad 3				alv			7	7			VBG	10
Pad 4				alv			7	7			VBG	10

TABLE XV

Test K Information Entered

Start Day 1 End Day 3

Name	Wt	Sz	Alt	Lv1	Lv2	LP	Fa	Na	NLT	Lch	LF	TT
Sat 1	20	100	300	sht	t2	1	1		5	n		
Sat 2	3	100	250	sht	t3	1	2		5	n		
Sat 3	20	100	600	alv		1	2		5	n		
Sat 4	20	100	500	sht	alv	1	5		7	n		
Sht 1	100	300	350	sht			6	6			Ken	30
Sht 2	100	350	350	sht			6	6			Ken	45
Alv 1	20	100	600	alv			2	2			VBG	99
Alv 2	30	150	600	alv			3	3			VBG	99
Pad 1				sht			5	5			Ken	15
Pad 2				sht			5	5			Ken	15
Pad 3				alv			2	2			VBG	10
Pad 4				alv			7	7			VBG	10

TABLE XVI

Test L Information Entered

Start Day 1 End Day 3

Name	Wt	Sz	Alt	Lv1	Lv2	LP	Fa	Na	NLT	Lch	LF	TT
Sat 1	20	100	300	sht	t2	1	1		5	n		
Sat 2	3	100	250	sht	t3	1	2		5	n		
Sat 3	20	100	600	alv		1	2		5	n		
Sat 4	20	100	500	sht	alv	1	5		7	n		
Sht 1	100	300	350	sht			1	1			Ken	30
Sht 2	100	350	350	sht			6	6			Ken	45
Alv 1	20	100	600	alv			2	2			VBG	99
Alv 2	30	150	600	alv			3	3			VBG	99
Pad 1				sht			5	5			Ken	15
Pad 2				sht			5	5			Ken	15
Pad 3				alv			2	2			VBG	10
Pad 4				alv			7	7			VBG	10

TABLE XVII

Test M Information Entered

Start Day 1 End Day 3

Name	Wt	Sz	Alt	Lv1	Lv2	LP	Fa	Na	NLT	Lch	LF	TT
Sat 1	20	100	300	sht	t2	2	1		5	n		
Sat 2	3	100	250	sht	t3	1	2		5	n		
Sat 3	20	100	600	alv		2	2		5	n		
Sat 4	20	100	500	sht	alv	1	1		7	n		
Sht 1	100	300	350	sht			3	3			Ken	30
Sht 2	100	350	350	sht			3	3			Ken	45
Alv 1	20	100	600	alv			3	3			VBG	99
Alv 2	30	150	600	alv			3	3			VBG	99
Pad 1				sht			1	1			Ken	15
Pad 2				sht			1	1			Ken	15
Pad 3				alv			1	1			VBG	10
Pad 4				alv			1	1			VBG	10

TABLE XVIII

Test N Information Entered

Start Day 1 End Day 3

Name	Wt	Sz	Alt	Lv1	Lv2	LP	Fa	Na	NLT	Lch	LF	TT
Sat 1	20	100	300	sht	t2	2	1		5	n		
Sat 2	3	100	250	sht	t3	1	1		5	n		
Sat 3	20	100	600	alv		2	2		5	n		
Sat 4	20	100	500	sht	alv	1	1		7	n		
Sht 1	100	300	350	sht			3	3			Ken	30
Sht 2	100	350	350	sht			3	3			Ken	45
Alv 1	20	100	600	alv			3	3			VBG	99
Alv 2	30	150	600	alv			3	3			VBG	99
Pad 1				sht			1	1			Ken	15
Pad 2				sht			1	1			Ken	15
Pad 3				alv			1	1			VBG	10
Pad 4				alv			1	1			VBG	10

The test file information was entered into each data base file prior to starting each LRS test session. When ever possible multiple schedule generations were used during each session. As each run executed the Test Result tables (Tables XIX to XXII) were completed.

Each X in the Test Result Tables indicates a successfully completed operation. Once an operation is verified by a test it is not required to be marked again even if it is checked again during another test.

Each LRS function was tested during the test procedure. The LRS functions being tested are evident from the Test Results Tables XIX to XXII and are not given any explanation. The explanations below for each test provide details on what scheduling problem was being tested during each run.

Test A. Test A was the first run in the testing session. All data base files were reset. Satellite 3 tested for each value being equal to the capability of the launch vehicle. It also tested for LRS finding the earliest launch requirement regardless where it is in the data base file. The capability to reduce the launch priority when no higher priority satellite is available was tested by Satellite 3. Satellite 2 tested for each satellite requirement being less the vehicle capabilities, and a return of the priority to 1 on the days following launch of a lower priority satellite. It also tested for the highest

TABLE XIX

Test Results

TEST ITEM	TEST LETTER													
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
First Run	X	X			X	X	X		X		X	X	X	X
INPUT TESTS														
Information														
Name - only first run	X	X			X	X	X		X		X	X	X	X
Date - only first run	X	X			X	X	X		X		X	X	X	X
Comment - every run	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Start Day - every run	X	X	X	X	X	X	X	X	X	X	X	X	X	X
<=0 request new start	X													
no input request new start						X								
End Day - every run	X	X	X	X	X	X	X	X	X	X	X	X	X	X
<start request new end	X													
no input - schedule all		X												
Output Tests														
Screen Displays	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Intro Screen	X	X			X	X	X		X		X	X	X	X
Explanation Screens	X													
Main Menu	X	X	X	X	X	X	X	X	X	X	X	X	X	X
View Information	X	X												
Return to Main Menu		X												
Assumptions	X	X												
On the Screen	X	X												
On the Printer		X												
All DB Files	X													
Vehicle DB File		X												
Launch Pad DB File		X												
Satellite DB File		X												
Last Record Display	X	X												
Generate Schedule	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Explanation	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Menu displayed	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Return to Main Menu	X													
Schedule Resources	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Request Name	X	X			X	X	X		X		X	X	X	X
Request Date	X	X			X	X	X		X		X	X	X	X
Request Comment	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Printout Wanted ? (Y/N)	N	Y	N	Y	N	Y	N	Y	N	Y	N	N	N	N
Print Resource Schedule		X		X		X		X		X				
Printer On Msg		X		X		X		X		X				
Invalid Start	X						X							
Invalid End	X													
Program Action Messages														
Establish NLT Lch Day	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Looking for Satellite	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Looking for Vehicle	X	X			X	X	X	X	X	X	X	X	X	X
Looking for Launch Pad	X	X			X	X		X		X	X	X	X	X

TABLE XX

Test Results (cont)

TEST ITEM	TEST LETTER													
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Sat Launched	X	X			X	X					X	X	X	X
Sat Missed Launch no Veh					X	X	X			X	X			
Sat Missed Launch no Pad								X		X		X		
Schedule Completed	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Look for Unlaunched Sat	X	X	X	X	X	X	X	X	X	X	X	X	X	X
List Unlaunched Sat					X	X	X	X		X	X	X		
Look for next Sat					X		X	X			X	X		
Look for Available Veh	X	X	X	X	X	X	X	X	X	X	X	X	X	X
List Available Vehicles			X	X	X	X		X		X	X	X		
Look for next Veh			X		X			X			X	X		
Look for Available Pads	X	X	X	X	X	X	X	X	X	X	X	X	X	X
List Available Pads			X	X	X	X				X	X	X		
Look for next Pad			X		X						X	X		
Reset DB Files	X	X	X	X	X	X		X				X	X	X
Explanation	X	X	X	X	X	X		X				X	X	X
Menu displayed	X	X	X	X	X	X		X				X	X	X
Return to Main Menu		X												
Reset All DB Files	X	X			X	X						X	X	X
Reset Vehicle DB File			X	X				X						
Resetting Veh File	X	X	X	X	X	X		X				X	X	X
Reset Pad DB File			X	X										
Resetting Pad File	X	X	X	X	X	X						X	X	X
Reset Satellite DB File								X						
Resetting Sat File	X	X			X	X		X				X	X	X
Enter New Parameters Scrn	X													
Load Saved Parameters Scrn	X													
Change Parameters Scrn	X													
Save Parameters Scrn	X													
Save Resource Sched Scrn	X													
Quit				X										
Printer Output														
Name		X		X		X		X		X				
Date		X		X		X		X		X				
Comment		X		X		X		X		X				
Banner		X		X		X		X		X				
Assumptions		X		X		X		X		X				
Schedule Days		X		X		X		X		X				
Sat Launched		X				X								
Sat Missed Launch No Veh						X				X				
Sat Missed Launch No Pad								X						
List Unlaunched Sat						X								
List Available Veh				X		X								
List Available Pad				X		X								

TABLE XXI

Test Results (cont)

TEST ITEM	TEST LETTER													
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
File Output														
Name						X								
Date						X								
Comment						X								
Banner						X								
Assumptions						X								
Schedule Days						X								
Sat Launched						X								
Sat Missed Launch No Veh						X								
Sat Missed Launch No Pad								X						
List Unlaunched Satellites						X								
List Available Vehicles						X								
List Available Launch Pads						X								
Program Operation														
Scheduling														
No Sats Available			X	X					X					
No Vehicles Available							X		X					
No Pads Available								X	X					
Sufficient Resources	X	X												
One Resource/Satellite	X	X											X	X
One/Day	X	X												
Multiple/Day	X	X			X	X	X	X					X	X
Different Priorities	X	X											X	X
All same priority					X	X	X	X						
Resource Excess			X	X										
Vehicles			X	X										
Pads			X	X										
Resource Shortage					X	X	X	X						
Vehicles					X	X	X							
All used							X							
Not satellite compatible					X	X								
Satellite Weight					X									
Satellite Size						X								
Orbit Altitude						X								
Vehicle Type					X									
1st Veh no pad choose 2nd						X								
Pads						X		X						
All used								X						
Not vehicle compatible														
Wrong Facility						X								
Time Shortage							X	X	X					
NLT before resource avail										X				X
End before desired launch									X	X				
All satellites									X					
Some satellites										X				

TABLE XXII

Test Results (cont)

TEST ITEM	TEST LETTER													
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
End before vehicle avail							X			X				
All satellites							X							
Some satellites										X				
1st sat no vehicle											X			
other sat vehicle avail											X			
End before pad available								X		X		X		
All satellites								X						
Some satellites										X				
1st sat no pad												X		
other sat pad available												X		
Time Variations	X	X												
Res avail on desired day	X	X										X		
vehicle	X	X										X		
pad	X	X										X		
Res after des before NLT					X					X			X	
vehicle										X			X	
pad				X	X					X				
Res avail after desire Lch													X	
Higher Priority Sat has:													X	
later launch day													X	
earlier launch day													X	
same launch day														X
Schedule Output	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Satellites	X	X			X	X	X	X	X	X	X	X	X	X
All Satellites Launched	X	X											X	X
Highest Priority First	X	X											X	X
No Satellites Launched			X	X			X	X	X	X				
Some Satellites Launched					X	X					X	X		
Unlaunched Satellites Listed					X	X	X	X		X	X	X		
Vehicles	X	X	X	X	X	X	X	X	X	X	X	X	X	X
All Vehicles Used	X	X											X	X
Some Vehicles Used					X	X					X	X		
No Vehicles Used			X	X			X	X	X	X				
All Vehicles Available			X	X				X						
Available Vehicles Listed			X	X	X	X		X		X	X	X		
Launch Pads	X	X	X	X	X	X	X	X	X	X	X	X	X	X
All Pads Used	X	X											X	X
Some Pads Used					X	X					X	X		
No Pads Used			X	X			X	X	X	X				
All Pads Available			X	X										
Available Pads Listed			X	X	X	X				X	X	X		

priority satellite being launched first when multiple satellites are available on the same day. Satellite 1 tested a second satellite being launched on the same day with a lower priority. Satellite 4 tested using the second vehicle type when the first is not compatible with the satellite requirements.

Test B. All data base files were reset for test B. It was a test of the printer output. The same information was used from test A.

Test C. Test C was a test for no satellites available for launch during the schedule period. Only the Vehicle and Pad data base files were reset to make the resources available after the first session. It also tested the ability of LRS to list the available resources after schedule completion.

Test D. Test D was a printer output test. The same data base information was used as in test C. No new schedule capability tests were performed.

Test E. The data base files were changed for test E. Satellite 3 tested missing a launch due to no vehicle available with sufficient boost capability. Satellite 1 tested for a normal launch capability after missing a launch as well as having two vehicles available on the same day with the same priority. In this case the first vehicle found is launched first. Satellite 2 tested a satellite having to delay launch for an available launch pad and a

second satellite available on the same day with the same priority being launched. Satellite 4 was a test for missing launch due to no vehicle available with sufficient payload capacity.

Test F. Test F required changing the data base files. They were then all reset to eliminate the effects of the last session. Satellite 3 tested for a missed launch because no launch vehicle of the proper type was available. Satellite 1 tested missing launch for no vehicle available capable of the required orbit altitude. Satellite 2 tested launching a satellite with the lowest priority as well as waiting for an available launch pad. Testing for the ability to change vehicles was accomplished by satellite 4. The first vehicle which matched the satellite requirements did not have an available launch pad. A second vehicle and launch pad needed to be matched to launch this satellite. The file saving capability was tested after this session. The sample output is provided as Appendix E.

Test G. The case with no available vehicles during the scheduling period was tested by test G. All available satellites needed to be listed as missing launch at completion of the schedule. No vehicles were to be listed and all pads were available.

Test H. The satellite and vehicle data base files were reset prior to attempting to generate a schedule for test H. Test H tested for no launch pads being available during the

schedule period. The pads were not available due to being previously used and their turn times were not reached before the end of the schedule time. This test required checking every possible satellite/vehicle combination since no launch pads were available. The number of iterations required for this test were the greatest by at least a factor of three, than for any other test. All operations worked properly and all satellites were listed as unlaunched and all vehicles as available at the end of the schedule period. No launch pads were listed as available.

Test I. Testing for a one day schedule with no requirements to be scheduled or resources available during that day was accomplished by test I. The day used for the test was day 1, but any day could have been used with the same effect. At schedule completion no satellites were listed as launched and no resources were listed as available.

Test J. Test J continued the testing for schedule timing difficulties. Satellite 1 tested a satellite with a one day launch window and no compatible vehicle available in time. Satellite 2 tested for a missed launch because the satellite NLT launch day was reached before the vehicle was available for launch. Satellite 4 was to test for a missed launch with no launch pad available before the end schedule day. Satellite 3 tested for the schedule day ending before the satellite was available for launch.

Test K. Test K was to test for a vehicle not being compatible with a launch pad and another vehicle required to obtain a launch. Satellite 3 was launched after having to reject the first vehicle selected due to no available launch pad.

Test L. Test L was to test for a missed launch due to a pad not being available before the end schedule day. The search continued to launch Satellite 3 on the following day demonstrating the ability to launch a satellite on the next day after a satellite missed a launch the previous day.

Test M. Test M was to test for multiple satellites with different launch priorities and different launch days, all waiting for the same resources. The satellites were available on different launch days to assure this did not affect the highest priority satellite being launched first when the resources became available. Satellites were launched according to desired launch day, and by launch priority where multiple satellites had the same desired launch day. It tested a resource being available on the last day of the schedule.

Test N. Test N was the same as test M only with all satellites having the same desired launch day. As in test M the highest priority satellite was launched first. The expected launch results were obtained.

Evaluation

All the input and output routines operate properly in LRS. All resource and requirement matching involving capacities and capabilities operate properly. Scheduling requirements involving waiting on an available satellite or vehicle operate properly.

During the testing phase several new problems with Insight 2+ occurred. These were corrected by changing the rules in the knowledge base and changing the DBPAS access routines. The problems encountered are documented in Appendix D and are said to be corrected in version 1.3 (36).

Summary

The tests applied to LRS in no way cover all the possible resource and requirement matching cases. All the input and output functions of LRS operate properly. The schedule generated meets all the requirements and constraints of the entered data. Unlaunched satellites and available resources are listed for consideration by the planner. All the test cases tried were executed properly by LRS. As with any software testing, this does not prove LRS will always operate properly, only that no faults were detected during this testing process.

V. Conclusions and Recommendations

Overview

This chapter presents a summary of the conclusions reached during this project. A brief restatement of the problem and the solution methodology is presented. The appropriateness of the knowledge based approach to the launch resource scheduling problem is examined. Recommendations for additional information and features to be incorporated in the operational system are provided. A comparison of the three tools used to examine the launch resource problem is made. The chapter ends with a summary of the work completed.

The Problem Addressed

U.S. Space Command's Operations Plans office (J3X), requested a computer program be developed to estimate the launch support required to maintain any number of satellite constellations at a given level of performance. The goal was to provide a prototype tool which a planner can use to determine how well launch resources meet the launch requirements.

The Solution

LRS provides a prototype tool which attempts to match the available launch resources to the given launch requirements. After completion of the matching process a

list of requirements not met and available resources is provided. This allows a planner to examine how resources are used and where a shortage or excess capability exists. This information may be used to make estimates of the type and number of launch resources required to meet the predicted launch need.

Is A Knowledge Based Approach Appropriate ?

Matching launch resources to launch requirements requires establishing a schedule for how the resources will be used. This scheduling process is mainly a matter of satisfying various requirements and constraints and does not involve complex computations. Knowledge based systems are good at matching problems and not as good at computational problems. The launch resource scheduling problem is knowledge intensive therefore a knowledge based approach is applicable. The requirements and constraints may change as the launch needs change, requiring a different scheduling methodology be used. In a knowledge based system it is easier to make changes than in a conventional program since changes may be made without full understanding of the total knowledge base operation. New information may be handled by adding new rules which will properly consider the new information.

AI is being used for scheduling applications due to the large number of requirements and constraints which must be considered to construct a schedule. It is difficult for a

human scheduler to remember all the factors involved and to consider all the interactions which may occur. A computer does not forget to consider any situation and does not forget to check for all interactions. The difficulty is providing the computer with all the information which must be considered and detailing the interactions which must be checked. A knowledge base contains all the known requirements and constraints of the scheduling problem. The scheduling process is then a matter of trying to satisfy these known requirements and constraints with the available resources.

Knowledge based systems have several advantages over conventional programming. The knowledge base is generally easier to update than conventional programs. If an additional requirement or constraint becomes known, a new rule can be added to the knowledge base to consider this constraint. Few other changes are required in the remainder of the knowledge base. There is no need to understand the complete knowledge base operation as is generally required to make changes to a conventional program. A knowledge base is normally easier to understand than a conventional program. This makes updating the program much easier. Easy updating is an important consideration in the launch resource scheduling problem. The method used to match resources to requirements changes with time. A system must be able to be updated to match these changes in scheduling

practice if it is to remain useful.

A disadvantage of a knowledge based system is usually operating speed. Since the program must develop the execution sequence every time the program is run instead of the sequencing being designed into the program, the knowledge base has a slower execution speed than a similar conventional program. In the launch resource scheduling problem this reduced execution speed is not a serious consideration, as long as the execution time does not get too long to be useful.

LRS provides a prototype system for the launch resource scheduling problem using a knowledge based structure. The tests demonstrate this approach provides acceptable results. The rule base which accomplishes the matching is separate from the launch requirements and launch resources. This provides easier changes to match the different considerations made in the scheduling problem. If the method used to accomplish resource allocation is changed the rule base must be modified to match the change. This has no effect on the resources or requirements. If the requirements or resources change the changes are made there with no effect on the matching procedure. The execution speed of the prototype is not too slow to be useful to a planner. It takes minutes but not hours to obtain a schedule.

The LRS prototype demonstrates a knowledge based

approach is appropriate for the launch resource scheduling problem.

Operational System Enhancements and Improvements

There are several areas of the prototype system which require enhancements and improvements to develop an operational system. The major areas requiring additional development work are explained below and summarized in Table XXIII.

Additional Information Required. The prototype does not consider all the information which an operational system should consider. These factors were not implemented in the prototype to reduce the development time. Some of the additional information which should be included in the operational system is listed below.

The avalibility of other resources is a consideration in the actual scheduling problem. These include items such as launch crews, launch support facilities, etc. None of these are considered in the prototype. Information concerning which of these are required for each launch and the availability of each of these other resources should be included in the data base information.

The launch priority of a satellite is fixed for the entire scheduling process in the prototype system. In an operational system the launch priority of a satellite may change as time progresses. A satellite designed to replace

TABLE XXIII

Summary of Operational System Recommendations

Recommendation	Area of Prototype Affected				
	KB Rules		DB Files		
	Sch	Con	Sat	Veh	Pad
Other Resources Required	X	X	X	X	X
Changing Launch Priority	X		X		
Satellite Storage Location		X	X		
Multiple Satellites/Vehicle	X	X			
Maximum Launches/Day	X	X			
Moving Launch Day for Availability	X				
Optimal Schedule	X				
Mission Duration Consideration	X	X			
Satellite Checkout Time		X	X		
Multiple Launch Windows	X	X	X		
Preassigned Launch Resources	X				
Enter New Parameters Function	X		X	X	X
Load Saved Parameters Function	X		X	X	X
Change Parameters Function	X		X	X	X
Save Parameters	X		X	X	X
Partial Schedule Saving	X		X	X	X
Record Pointers/Tracking	X				
Next Available Pointers	X				
Condition Flags	X				
Schedule Date Entry	X				

a failing satellite is assigned a planning launch priority which assures it will be launched before the satellite it is replacing fails. However if during scheduling it is discovered the satellite cannot be launched by the date the satellite it is replacing fails, the priority may become much greater. A means by which LRS can increase the priority of a satellite after a certain day would be desirable to accommodate this type consideration. The launch requirement information should contain the times when the priority should be increased and the rate at which the priority should be increased.

No consideration is given to the storage location of the satellite. The LRS prototype assumes the satellite will be prepositioned to the location where compatible vehicles are located. The launch requirement information should provide the location of the satellite, so a vehicle is chosen which is co-located with the satellite or consideration given to transportation time.

Scheduling Considerations. The LRS prototype system uses simplified scheduling considerations to reduce development time. An operational system needs to consider many more factors in generating the schedule. Some of the factors which might be included are listed below.

A major consideration not provided in the prototype is launching multiple satellites per launch vehicle. This is a common practice for meeting launch requirements with

vehicles like the space shuttle. The operational system must consider launching multiple satellites per launch vehicle when ever possible.

There is no limit in the prototype on how many vehicles can be launched on any one day. In reality there is definitely a limit. This limitation must be considered in an operational system.

Providing a means to move launches to better use the available launch resources would provide a better schedule. Satellites may be launched earlier than the desired day in order to allow for more satellites to be launched. The prototype does not check for this possibility.

The prototype does not provide an optimal schedule. The schedule provided by the operational system should be an optimal schedule or one which is very close to optimal. This allows the planner to make the best use of the available resources. It would also provide greater confidence in the recommendations made based on the system results.

The prototype gives no consideration to mission duration or required checkout time for each satellite. This information should be considered in an operational system. This affects the turn around time of the launch vehicles.

Providing for multiple launch windows for satellites would be a good feature. This allows for satellites which can only be lunched on specific days or they must wait until

the next available launch window. The prototype only allows one launch window per satellite.

The ability to preassign launch resources may be desired in the operational system. This allows matching satellites to dedicated launch vehicles but still allow LRS to match a launch pad and fit the launch into the schedule where appropriate.

Functions to Implement. The operational system should have all functions implemented within LRS. The functions not implemented within the LRS prototype, which should be implemented in the operational system are described below.

The Enter New Parameters Function allows entering any information required by LRS to generate a launch schedule for planning. This should include adding information to any of the data base files. This function should provide for creating all the satellites in a constellation without having to enter each satellite individually. This could be accomplished by entering default values for the constellation, then establishing the launch rate and the total number of satellites required. The program could then enter the required number of satellites into the data base each with the proper launch day to meet the specified launch rate. New vehicles or vehicle replacements could also be generated in a similar manner by specifying the production rate for the vehicles.

The Load Saved Parameters option should allow selecting

the desired data base information to use for each LRS session. Currently LRS must be exited and DOS commands used to rename the desired files so LRS will access them during program execution.

The Change Parameters option should allow changing any data base information from within LRS. This should include changing constellation launch rates and priorities. This would allow changing all the related entries in the data base with a single entry instead of making each change manually.

Save Parameters should allow saving the current data base files under specified names for later use. The prototype LRS system requires the use of DOS to rename the files.

The ability to save a partially completed session is provided by the Insight 2+ program (20). However, this does not save the condition of the data base files, which is critical to proper execution when the session is later restored for continuation. Saving the data base file states should be provided. This should also specify a file name for the partially executed files to prevent them from being used in another session.

Execution Speed. There are several areas in the LRS prototype system where the program execution speed can be greatly improved by some additional development work. The major areas for improvement are explained below.

The prototype system is very dependent on disk access speed. This results from the data base files being accessed every time a match is attempted. Each time a new requirement or resource is required the search process starts at the beginning of the data base file and searches every record until the needed item is found. By implementing a pointer system and a tracking system for which records are no longer applicable, the number of records searched could be greatly reduced and the execution speed improved.

Another improvement which could significantly improve the execution speed is the use of next available pointers. This was used in the prototype to track the next available satellite, vehicle, and pad day. This allows skipping schedule days where no requirements or resources are available and prevents searching the data base files for those days. If this were implemented for satellite launch priorities a similar time savings could be realized when searching for a lower priority satellite.

The use of flags which can indicate the presence or absence of possible solutions would save time. The prototype has flags to indicate a lower priority satellite is available on the current schedule day. This prevents searching for all launch priorities every schedule day. Lower priorities are only searched for if this flag is set. Setting up similar flags for the availability of secondary

vehicles, other launch pads, etc. would save search time if these items are not available and the first resource found does not meet the requirement.

User Convenience. The ability to enter scheduling period as calendar dates instead of schedule days would be an improvement. This could be further extended to allow entry of launch requirements and launch resource availability as calendar dates and providing for the output to be listed as calendar dates. This would make program use easier in some cases and the output more understandable.

Tool Comparisons

A knowledge base tool has several advantages over OPS-5 or the simulation language SLAM (33) for the launch resource scheduling problem. Both OPS-5 and SLAM were used to implement preprototype systems before deciding to use Insight 2+.

OPS-5 is a programming language implemented in Franz Lisp. It has a forward chaining inference engine and uses production rules as the knowledge representation system. OPS-5 uses a last first rule execution priority (7, 5). This promotes a depth first search strategy. In the launch resource scheduling problem this strategy is not beneficial. As mentioned in Chapter 3 the schedule process is cyclic and requires repeated application of a subset of rules until a condition is met. This type operation is not easy to accomplish in OPS-5. Another factor is the code

readability. OPS-5 uses Lisp structure for its rules. Without Lisp experience the rules are difficult to comprehend. Even with Lisp experience determining what each rule accomplishes can be difficult without adequate comments. This makes system updating more difficult than a system with more easily understood code. Insight 2+ uses production rules also, but the code is created with IF THEN ELSE rules which are more easily understood than the Lisp constructs of OPS-5. This makes the system using Insight 2+ easier to update.

Creating displays with OPS-5 requires developing program statements to send the information to the desired device. Although not difficult, it is not as easy as the method used in Insight 2+ which requires simply naming the display and typing it as you wish it to be displayed. The paging function is already designed into Insight 2+, requiring no special code.

OPS-5 has some advantages over Insight 2+. Since it is a language not a tool, it has greater flexibility. Any desired operation can be performed internally to the program with access to all program variable values. Insight 2+ allows calling external programs to accomplish any desired operation, however this is an external operation requiring loading of the program, passing desired parameter values, then execution, so some time is lost during this process. Also, only the values passed are available to the

external program, not all program values. Both have design flexibility, OPS-5 has it internally, Insight 2+ has it externally, the major difference is execution time and accessibility of program variable values.

An OPS-5 program is easier to debug than an Insight 2+ program. OPS-5 allows several options for tracing program execution. It also allows setting flags which cause the program to begin tracing when a certain point is reached. Insight 2+ has no similar trace features. The trace feature makes OPS-5 programs much easier to debug than the Insight 2+ programs.

OPS-5 is not a compiled language. Although a non-compiled language may be a disadvantage in the final product, it is helpful during the development work. A change can be made and the program immediately executed to test the change. Insight 2+ is a compiled language requiring recompiling before execution. The time required to compile the program becomes significant as the program size becomes larger. Also if a syntax error is made, the compile process must be restarted from the beginning, not continued from where the error occurred. In some cases the compiling may be continued past the detected error to check for other errors, but the compiling process must be reaccomplished once the error is corrected.

SLAM and Insight 2+ are designed for totally different purposes, however since they were both used for a similar

problem a comparison as applied to this problem will be made.

SLAM is a simulation language (33). In simulations the primary emphasis is on probabilities of events. SLAM is FORTRAN based, consequently it has many of the drawbacks of conventional programming relating to launch resource scheduling.

Understanding SLAM program code can be difficult. Variable names in SLAM are very limited compared to Insight 2+. This can further complicate understanding the program code.

A very good understanding of the SLAM language is required before changes can be made due to the complex nature of simulation. Insight 2+ requires an understanding of the general principles of the tool but an extensive knowledge is not required to make changes.

Producing custom displays in SLAM requires FORTRAN programming and calls to external FORTRAN routines. This is much more difficult than the method employed in Insight 2+.

Entering the required information for a SLAM program requires a good understanding of exactly what information the program will use and how it will use it. This may be circumvented by using FORTRAN programs to format the data properly and request for the proper data, but all this must be accomplished by the developer. Insight 2+ has a much simpler system by using the internal question system or

using the data base interface.

SLAM is a very complex simulation language capable of many different simulations. Consequently it requires several weeks to obtain a working knowledge of the program. Insight 2+ has a simpler structure which is much easier to learn. A working knowledge can be obtained in a much shorter time.

SLAM and Insight 2+ are not designed to handle the same type problems. This is the main reason why Insight 2+ has several distinct advantages over SLAM in handling the launch resource scheduling problem.

Summary of Work

The goal of this work was to develop a prototype launch resource scheduling tool which could be used by US Space Command planners to evaluate future launch resource requirements. The tool needed to be simple to operate and be as easy as possible to update as policy and needs change. To accomplish this a knowledge based tool, Insight 2+, was used to develop the LRS prototype system.

The LRS prototype provides a planning tool which is menu driven and provides explanation of program features for ease of use. The code is written in IF THEN ELSE production rule format making it as easy as possible to understand and thus easy to update as policy and needs change. LRS uses dbase II format files for launch resource and launch

requirement information. This makes information entry simple and easy to understand. It also allows using data base functions to make required changes to the scheduling information.

LRS matches launch resources to launch requirements. A launch schedule is produced during the matching process which demonstrated how the resources meet the launch requirements. At completion of the matching process LRS provides a list of unsatisfied requirements and available resources which may be used by the planner to determine how well the planned launch resources meet the estimated launch requirements.

The LRS prototype system demonstrates the applicability of the knowledge based approach to the launch resource scheduling problem. The advantages of ease of use and ease of update are demonstrated in the prototype. An operational system would require several enhancements and improvements to the prototype, but should meet the desired need when implemented.

Appendix: A Description of Insight 2+

The following discussion provides a summary of the capabilities of Insight 2+. It is designed to provide enough information to allow a basic understanding of the program code supplied in Appendix B. For a detailed discussion see the Insight 2+ Reference Manual (20).

Inference Engine

Backward chaining inference engine with forward chaining capability.

Rule sequencing is determined by the goal precedence established in the control section of the program. The first goal is tried to be satisfied first. All rules which have an antecedent of that goal are examined to try to satisfy the goal. If it can be satisfied then the subgoals under the first goal are tried in the same manner. If a goal has no rule associated with it which can be examined or satisfied the system generates a question asking if the goal is true or false to attempt to satisfy the goal. Once the first goal and all its subgoals are satisfied the second goal is attempted.

Instructions are provided to restart KB with or without forgetting previously provided or derived facts.

Knowledge Representation

Uses a Production Rule Language (PRL) for all rule and fact representation.

Rule Structure:

```
RULE name of rule
IF the first condition
  AND the second condition
  OR alternate second condition
  AND/OR ...
THEN simple action
  AND any other actions
ELSE any other actions
```

Fact Structure

Simple Facts - True/False

Numeric - numbers, integer or real

relational comparisons allowed <, >, <=, >=, =, <>
Object-

Attribute - single or multiple attributes allowed

object IS/ARE characteristic

String - 80 characters max.

relational comparisons allowed <, >, <=, >=, =, <>
Object-

All facts may have a confidence factor assigned by user. A confidence level threshold may be assigned by KB engineer.

Data Base access supported using dBase II or dBase III format. Has data base file construction and editing program.

Program Control

INIT - initialize a value for a fact at start of program only.

REINIT - initialize a value for a fact when CYCLE called.

MULTI - establish an object as having multiple values.

GOALSELECT - determines if user should be asked to select desired goal to pursue or not.

SUPPRESS - determines which goal summaries to suppress at the end of the session.

Other Features

FORGET - in command section specifies which goals and facts to forget when CYCLE is called. In rule, forgets the specified fact.

CYCLE - causes the inference engine to start at top of rule base again, reinitialize any REINIT values and forget any facts listed under control section FORGET.

CALL - execute the specified external program code.

SEND - send the values stored in the listed facts.

RETURN - receive values from the external program and store in the listed facts.

TITLE function - display any amount of introductory information at start of program. Automatic paging.

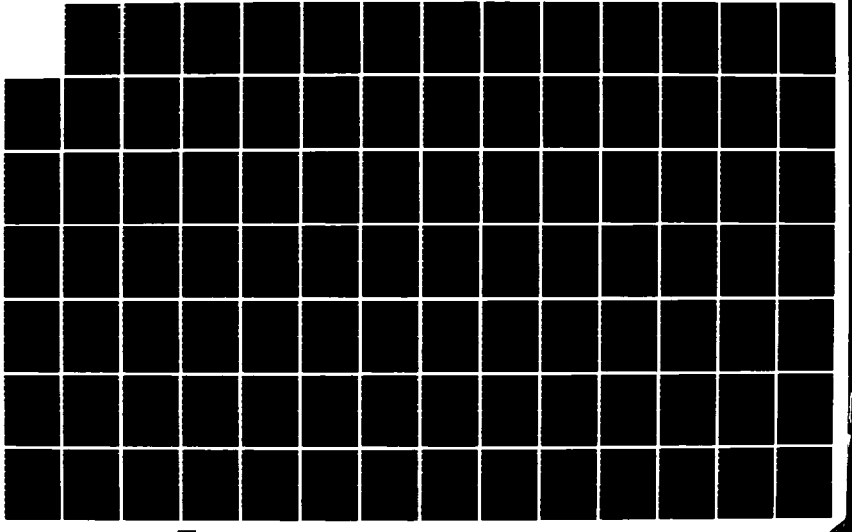
DISPLAY function - display any amount of information when rule calls for it. Automatic paging. May display variable values.

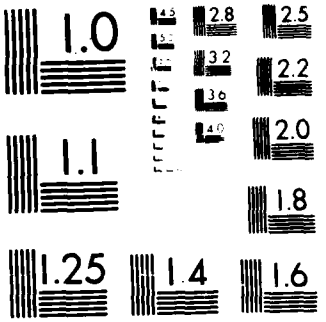
PRINT function - send display information to the printer when rule calls for it. Uses same format as DISPLAY.

FILE function - in the control section of the program, opens the specified file for FILE output. In a rule, sends

AD-A179 426 LRS: A KNOWLEDGE BASED APPROACH TO LAUNCH RESOURCE SCHEDULING(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING F H KOCH DEC 86 2/3

UNCLASSIFIED AFIT/GSO/ENS/86D-4 F/G 22/4 NL





display information to the file opened in the control section of the program.

EXPAND function - display any amount of information when requested by user. Automatic paging.

TEXT function - allows specifying wording of question to be asked user.

Appendix B: Program Code

The following LRS program code is broken into two major parts. The first part is the LRS Production Rule Language (PRL) code. This code contains all the scheduling and program operation rules. The second part is the DBPAS code. This code provides access to the data base files, and performs the file update functions.

LRS PRL Code

```
! Insight 2+ LRS: A Knowledge Based Approach to Launch Resource Scheduling
! Version 1.0   1 Dec 86  by  Major Fred H. Koch
! Copyright 1986 by Fred H. Koch
!
!=============================================================================
!  Display Title of the program and basic program discription
!=============================================================================
!
TITLE LRS: Launch Resource Scheduling   v 1.0 DISPLAY
*******************************************************************************
|                                                                             |
| *************************************************************************** |
| |                                                                         | |
| |                                                                         | |
| |                              L R S                                      | |
| |                                                                         | |
| |                    Launch   Resource  Scheduling                        | |
| |                                                                         | |
| |                         A  Planning  Tool                               | |
| |                                                                         | |
| |                              by                                         | |
| |                                                                         | |
| |                         Major Fred H. Koch                              | |
| |                                                                         | |
| |                                                                         | |
| |           Press Function Key 1 PAGE for program discription             | |
| |                                                                         | |
| *************************************************************************** |
|              Press  Function Key 3 STRT to start the program                |
*******************************************************************************

*******************************************************************************
|                     LRS: Launch Resource Scheduling          Page 1 of 7    |
| *************************************************************************** |
```

LRS is a planning tool. It is designed to allow a planner to estimate the effect various launch resource configurations will have on meeting the planned launch requirements.

Before entering the program the planner enters the planned launch resources and the planned launch requirements. Running the program will attempt to match the requirements to available resources. The program will end when all requirements are met, the schedule time is out or resources are exhausted.

The results can be printed or saved to a disk file for future reference.

Press Function Key 1 PAGE to view additional information

Press Function Key 3 STRT to start the program

LRS: Launch Resource Scheduling Page 2 of 7

The required resource and requirements information is stored in several data base files. This information includes launch vehicle data, launch pad data, and satellite data. This information is used by the program along with the resource utilization rules to match resources to requirements. The information is updated as the schedule is created, but may be reset prior to the next run.

The program can be exited at any time by pressing Function Key 10 EXIT. This will leave the schedule in a partially completed state. If the program is exited in this manner, the data base files should be reset prior to attempting to schedule the resources again.

Press Function Key 1 PAGE to view additional information

Press Function Key 3 STRT to start the program

LRS: Launch Resource Scheduling Page 3 of 7

When the matching process is completed, a list of any requirements not met and available resources is provided. This list along with the schedule generated during matching can be used to evaluate how well the resources meet the requirements.

```
|                                                                              |
|                                                                              |
|                                                                              |
|        Press Function Key 1 PAGE to view additional information              |
|                                                                              |
********************************************************************************
|             Press  Function Key 3 STRT to start the program                  |
********************************************************************************
```

```
********************************************************************************
|                    LRS: Launch Resource Scheduling          Page 4 of 7      |
********************************************************************************
|                                                                              |
|   The schedule information is saved to a disk file during execution.         |
|   The file must be copied before the program is re-run to prevent it         |
|   from being over written during the next matching process.                  |
|                                                                              |
|   Use the DOS function from the Insight 2+ Main Menu or the DOS prompt       |
|   after exiting the program to copy the file SCHEDULE.DAT.                   |
|                                                                              |
|   The disk file can be displayed to the screen or printed at a later         |
|   time using the DOS command TYPE.  The file can also be loaded into         |
|   a Word Processor for addition to any other text or reformatting as         |
|   desired.                                                                   |
|                                                                              |
|        Press Function Key 1 PAGE to view additional information              |
|                                                                              |
********************************************************************************
|             Press  Function Key 3 STRT to start the program                  |
********************************************************************************
```

```
********************************************************************************
|                    LRS: Launch Resource Scheduling          Page 5 of 7      |
********************************************************************************
|                                                                              |
|   Further explanation is provided for selected items during the              |
|   running of the program.  The items which have explanations available       |
|   are indicated by the EXPL Function Key being highlighted at the            |
|   bottom of the screen.  Pressing this function key will present the         |
|   additional explanation text.                                               |
|                                                                              |
|   More detailed explanation of the program operation is available in         |
|   user's manual.                                                             |
|                                                                              |
|                                                                              |
|                                                                              |
|         Press Function Key 1 PAGE to view additional information             |
|                                                                              |
********************************************************************************
|             Press  Function Key 3 STRT to start the program                  |
```

```
                    LRS: Launch Resource Scheduling          Page 6 of 7

   The following files are required for LRS Program Operation:

   LRS      .KNB            DBPAS    .COM            SCRNMSG .PCO
   PAD      .DBF            SATELITE.DBF             VEHICLE .DBF
   RESETPAD.PCO             RESETSAT.PCO             RESETVEH.PCO
   VIEWPAD .PCO             VIEWSAT .PCO             VIEWVEH .PCO
   GETPAD  .PCO             GETSAT  .PCO             GETVEH  .PCO
   MARKPAD .PCO             MARKSAT .PCO             MARKVEH .PCO

   The Insight 2+ program must be configured as specified in the
   I2+ manual for either floppy disk or hard disk operation.  The
   above files are in addition to the normal I2+ required files.

         Press Function Key 1 PAGE to view additional information

             Press  Function Key 3 STRT to start the program
```

```
                    LRS: Launch Resource Scheduling          Page 7 of 7

   Hard disk operation is recommended for LRS operations.  The
   program is very disk intensive during operation which makes its
   execution speed very dependent on the disk access speed.  Using
   a hard disk improves the execution speed significantly.

   The files listed above must all be located in the defaunt direc-
   tory on the default drive.  See the Insight 2+ operator manual
   for details on setting the defaunt disk parameters.

   An listing of what information is contained in each of the above
   files may be found in the LRS Operators Manual.

         Press Function Key 1 PAGE to start with Page 1 again

             Press  Function Key 3 STRT to start the program
```

```
!
!=============================================================================
!  Declaration of facts shared by chained knowledge bases
!=============================================================================
!
```

```
!    No shared facts in this program
!
!=============================================================================
!  Data type declarations
!=============================================================================
!
SIMPLEFACT Sat Classified
  AND Sat Lch With Classified
  AND Sat Avail With Lower Priority

OBJECT Find Information
  AND Info Output Status
  AND Option
  AND Print Status
  AND Program Option
  AND Reset File
  AND Status
  AND View Choice

NUMERIC Boost Capability
  AND Current Priority
  AND Current Schedule Day
  AND Day Vehicle Found
  AND Ending Schedule Day
  AND Highest Priority
  AND Latest NLT Lch Day
  AND Loaded Sat 1 Record Number
  AND Loaded Sat 1 NLT Lch Day
  AND Lowest Priority
  AND Mission Duration
  AND Next Avail Lch Day
  AND Number Of Sat To Lch
  AND Number Of Sat
  AND Orbit Altitude Cap
  AND Pad Record Number
  AND Pad Turn Time
  AND Pad First Available
  AND Pad Next Available
  AND Payload Size
  AND Record Number
  AND Sat Available
  AND Sat Desire Lch
  AND Sat Inclinatn
  AND Sat Lch Priority
  AND Sat NLT Lch
  AND Sat Orbit Alt
  AND Sat Record Number
  AND Sat Size
  AND Sat Weight
  AND Schedule Day Sat Available
  AND Start Schedule Day
  AND Step
```

```
    AND Veh First Available
    AND Veh Last Tried
    AND Veh Next Available
    AND Veh Record Number
    AND Veh Turn Time

  STRING Comment
    AND Constel
    AND Constellation Data
    AND Date
    AND L_char
    AND Veh Launch Facility
    AND Lch Veh 1
    AND Lch Veh 2
    AND M_char
    AND Message
    AND Pad Data
    AND Pad Last Record
    AND Pad Launch Facility
    AND Pad Name
    AND Pad Type
    AND Sat Last Record
    AND Sat Launched
    AND Sat Name
    AND Sat Orbit Type
    AND Sat Restrictn
    AND Satellite Data
    AND Site Data
    AND User Name
    AND Veh Last Record
    AND Vehicle Data
    AND Vehicle Name
    AND Vehicle Type
    AND X_char

  !=============================================================================
  !  Parameter initialization statements
  !=============================================================================

  INIT Current Priority := 1
    AND Highest Priority := 1
    AND L_char := l
    AND Latest MLT Lch Day := 1
    AND Lowest Priority := 9
    AND M_char := m
    AND Pad Last Record := n
    AND Pad Record Number := 0
    AND Sat Record Number := 0
    AND Step := 1
    AND Veh Record Number := 0
    AND X_char := x
```

```
!REINIT

FORGET Option

!=============================================================================
!  Control element selectors
!=============================================================================

FILE SCHEDULE.DAT

!THRESHOLD
!CONFIDENCE

GOALSELECT OFF

!  Must make Option a multi-attribute object to keep KB operating
!    once a single operation is performed.  This along with the CYCLE
!    command is what makes the KB forward chain while seeking the Option
!    Goal.

MULTI Option

!  Suppress all the conclusions which would normally be displayed at the
!    completion of the KB operation.

SUPPRESS ALL

!EXHAUSTIVE

!=============================================================================
!  The Goals of the knowledge base
!=============================================================================
!
!  Goal Outline
!    Determine what action the user wants to make.

1. Option IS WHAT

!=============================================================================
!  The rules for the knowledge base
!=============================================================================
!
!     RULE ... IF ... THEN ... ELSE
!
!-----------------------------------------------------------------------------
!  Present Main Menu of program options
!-----------------------------------------------------------------------------
!
!  Use steps to limit which rules can be attempted at any time.
!
!       Step 1  - Main Menu rules
!       Step 2  - Generate Schedule rules
```

```
!
!-------------------------------------------------------------------------
!   Main Menu Options
!
!     Present the options the user can choose and get the desired option

!     View Information
!       This uses the rules resulting in Present Information being true to
!         allow choice of viewing program assumptions, or the db info on
!         vehicles, pads, or satellites.

RULE Main Menu: View Information
IF Step = 1
  AND Program Option IS View Information
  AND Present Information
THEN Option IS View Information
  AND FORGET Program Option
  AND FORGET Present Information
  AND CYCLE

!     Generate Schedule
!       Present the instruction display and then set the the step to
!       move to the schedule generation rules

RULE Main Menu: Generate Schedule
IF Step = 1
  AND Program Option IS Generate Schedule
  AND Generate Schedule Screen Displayed
THEN Option IS Generate Schedule
  AND FORGET Program Option
  AND Step := 2
  AND CYCLE

!    Generate Schedule screen displayed only once each session

   RULE Display Generate Schedule Screen
   IF DISPLAY Generate Schedule Display
   THEN Generate Schedule Screen Displayed

!    Rule to choose to reset db files
!      Present Reset DB instructions then move to sub-menu to allow
!      choice of resetting all DB files or any one file.

RULE Main Menu: Reset DB files
IF Step = 1
  AND Program Option IS Reset DB Files
  AND Reset DB Screen Displayed
  AND Choose Which To Reset
THEN Option IS Reset Files
  AND FORGET Program Option
  AND FORGET Choose Which To Reset
  AND CYCLE
```

```
!    Display Reset DB screen only once each session

RULE Display Reset DB screen
IF DISPLAY Reset DB Display
THEN Reset DB Screen Displayed

!     Enter New Parameters
!       This rule presents a display stating the DB editor must be
!       used to enter new information.  In later versions this rule
!       would allow entry into the sub-menu to enter new parameters
!       from within the KB program.

RULE Main Menu: Enter New Parameters
IF Step = 1
  AND Program Option IS Enter New Parameters
  AND DISPLAY Enter New Parameters Display
THEN Option IS Enter New Parameters
  AND FORGET Program Option
  AND CYCLE

!     Load Saved Parameters
!       This currently displays instructions stating the DB files
!       are used to store information.  In later versions this would
!       allow entry into a sub-menu from which the desired DB files to
!       be used could be chosen.

RULE Main Menu: Load Saved Parameters
IF Step = 1
  AND Program Option IS Load Saved Parameters
THEN Option IS Load Saved Parameters
  AND DISPLAY Load Saved Parameters Display
  AND FORGET Program Option
  AND CYCLE

!     Change Parameters
!       Currently presents instructions to use DB editor to change
!       the information in the DB files used by the KB program.  In
!       later versions this would enter a sub-menu to allow changing
!       the DB values from within the KB program.

RULE Main Menu: Change Parameters
IF Step = 1
  AND Program Option IS Change Parameters
THEN Option IS Change Parameters
  AND DISPLAY Change Parameters Display
  AND FORGET Program Option
  AND CYCLE

!     Save Parameters
!       Currently displays a message which explains the DB files are
!       automatically saved during program execution, and how to save
```

```
!       the information using DOS commands.  In later versions this would
!       allow choice to save certain DB files for later use from within
!       the KB program.

RULE Main Menu: Save Parameters
IF Step = 1
  AND Program Option IS Save Parameters
THEN Option IS Save Parameters
  AND DISPLAY Save Parameters Display
  AND FORGET Program Option
  AND CYCLE

!     Save Resource Schedule
!       Displays a message explaining how to rename the file generated
!       during the schedule generation process for later use.  In later
!       versions this would allow saving the SCHEDULE.DAT file under the
!       file name entered by the user.

RULE Main Menu: Save Resource Schedule
IF Step = 1
  AND Program Option IS Save Resource Schedule
THEN Option IS Save Resource Schedule
  AND DISPLAY Save Resource Schedule Display
  AND FORGET Program Option
  AND CYCLE

!    Rule to end program
!      Terminates the KB program and returns to the Insight 2+ displaying
!       End of Session.

RULE Main Menu: Quit program rule
IF Step = 1
  AND Program Option IS Quit
THEN Option IS Quit
  AND STOP

!-------------------------------------------------------------------------
!  Main Menu Supporting Rules
!-------------------------------------------------------------------------
!
!-------------------------------------------------------------------------
!  Present Information Rules
!    These rules are used to present the information stored in the KB or
!    accessed by the KB from the data base files.  The information may be
!    sent to the screen or the printer.

!    View Assumptions choice
!      Display text information listing program assumptions whicn may not
!      be changed by the user during program execution.

RULE PI: View Assumptions
IF View Choice IS Assumptions
```

```
THEN Present Information
  AND DISPLAY Program Assumptions Display
  AND FORGET View Choice

!    View All Data base information
!      Call the three DBPAS programs which will display all the
!      information in the data base files.

!    View all information on the screen

RULE PI: View All DB Files on Screen
IF View Choice IS All DB Files
  AND Info Output Status IS on the screen
THEN Present Information
  AND CALL Viewveh
  AND CALL Viewpad
  AND CALL Viewsat
  AND DISPLAY Last Record Display
  AND FORGET View choice
  AND FORGET Info Output Status

!    Send all information to the printer

RULE PI: Send All DB Files to the Printer
IF View Choice IS All DB Files
  AND Info Output Status IS on the printer
  AND Check Printer On
  AND Message := Printing All Data Base Files
  AND CALL Scrnmsg
  SEND Message
  AND Get Vehicle Data
  AND Get Pad Data
  AND Get Satelite Data
THEN Present Information
  AND FORGET View Choice
  AND FORGET Info Output Status
  AND FORGET Get Vehicle Data
  AND FORGET Get Pad Data
  AND FORGET Get Satelite Data

!      Check Printer On

     RULE PI: Check Printer On
     IF DISPLAY Printer On Msg
     THEN Check Printer On

!      Get Vehicle Data

     RULE PI: Get Vehicle Data
     IF Veh Record Number := Veh Record Number + 1
       AND CALL Getveh
       SEND Veh Record Number
```

```
      RETURN Veh Last Record
      RETURN Vehicle Name
      RETURN Vehicle Type
      RETURN Boost Capability
      RETURN Payload Size
      RETURN Orbit Altitude Cap
      RETURN Veh Launch Facility
      RETURN Mission Duration
      RETURN Veh Turn Time
      RETURN Veh First Available
      RETURN Veh Next Available
      AND Veh Last Record <> e
    THEN Get Vehicle Data
      AND FORGET Get Vehicle Data
      AND PRINT Vehicle Record Format
      AND CYCLE
    ELSE Get Vehicle Data
      AND Veh Record Number := 0

!           Get Pad Data

    RULE PI: Get Pad Data
    IF Pad Record Number := Pad Record Number + 1
      AND CALL Getpad
      SEND Pad Record Number
      RETURN Pad Last Record
      RETURN Pad Name
      RETURN Pad Type
      RETURN Pad Launch Facility
      RETURN Pad Turn Time
      RETURN Pad First Available
      RETURN Pad Next Available
      AND Pad Last Record <> e
    THEN Get Pad Data
      AND FORGET Get Pad Data
      AND PRINT Pad Record Format
      AND CYCLE
    ELSE Get Pad Data
      AND Pad Record Number := 0

!         Get Satelite Data

    RULE PI: Get Satelite Data
    IF Sat Record Number := Sat Record Number + 1
      AND CALL Getsat
      SEND Sat Record Number
      RETURN Sat Last Record
      RETURN Sat Name
      RETURN Constel
      RETURN Sat Weight
      RETURN Sat Size
      RETURN Sat Orbit Alt
```

```
        RETURN Lch Veh 1
        RETURN Lch Veh 2
        RETURN Sat Lch Priority
        RETURN Sat Restrictn
        RETURN Sat Classified
        RETURN Sat Lch With Classified
        RETURN Sat Desire Lch
        RETURN Sat Available
        RETURN Sat NLT Lch
        RETURN Sat Orbit Type
        RETURN Sat Inclinatn
        RETURN Sat Launched
        RETURN Number Of Sat To Lch
        AND Sat Last Record <> e
      THEN Get Satelite Data
        AND FORGET Get Satelite Data
        AND PRINT Satelite Record Format
        AND CYCLE
      ELSE Get Satelite Data
        AND Sat Record Number := 0

!     View Launch Vehicle Data Choice
!       Call DBPAS programs which display the vehicle data from the
!       data base file VEHCILE.DBF to the screen or retrieve it for
!       sending to the printer.

!     View Vehicle on the Screen

RULE PI: View Vehicle DB Files on Screen
IF View Choice IS Vehicle DB Files
  AND Info Output Status IS on the screen
THEN Present Information
  AND CALL Viewveh
  AND DISPLAY Last Record Display
  AND FORGET View Choice
  AND FORGET Info Output Status

!     Send Vehicle DB File information to the Printer

RULE PI: Send Vehicle DB Files to Printer
IF View Choice IS Vehicle DB Files
  AND Info Output Status IS on the printer
  AND Check Printer On
  AND Message := Printing Vehicle DB Files
  AND CALL Scrnmsg
  SEND Message
  AND Get Vehicle Data
THEN Present Information
  AND FORGET View Choice
  AND FORGET Info Output Status
  AND FORGET Get Vehicle Data
```

```
!    View Pad Data Choice
!      Call DBPAS Programs which display the Launch Pad data from the
!      data base file PAD.DBF to the screen or retrieve it for
!      sending to the printer.

!    View Pad DB Files on the Screen

RULE PI: View Launch Pad Data on the Screen
IF View Choice IS Launch Pad DB Files
  AND Info Output Status IS on the screen
THEN Present Information
  AND CALL Viewpad
  AND DISPLAY Last Record Display
  AND FORGET View Choice
  AND FORGET Info Output Status

!    Send Pad DB Files to the Printer

RULE PI: Send Pad DB Files to the Printer
IF View Choice IS Launch Pad DB Files
  AND Info Output Status IS on the printer
  AND Check Printer On
  AND Message := Printing Launch Pad DB Files
  AND CALL Scrnmsg
  SEND Message
  AND Get Pad Data
THEN Present Information
  AND FORGET View Choice
  AND FORGET Info Output Status
  AND FORGET Get Pad Data

!    View Satellite Data Choice
!      Call DBPAS programs which display the satellite data from the
!      data base file Satellit.DBF to the screen or retrieve it for
!      sending to the printer.

!    View Satelite DB Files on the screen

RULE PI: View Satellite Data
IF View Choice IS Satellite DB Files
  AND Info Output Status IS on the screen
THEN Present Information
  AND CALL Viewsat
  AND DISPLAY Last Record Display
  AND FORGET View Choice
  AND FORGET Info Output Status

!    Send Satelite DB Files to the printer

RULE PI: Send Satelite DB Files to the Printer
IF View Choice IS Satellite DB Files
  AND Info Output Status IS on the printer
```

```
  AND Check Printer On
  AND Message := Printing Satellite DB Files
  AND CALL Scrnmsg
  SEND Message
  AND Get Satelite Data
THEN Present Information
  AND FORGET View Choice
  AND FORGET Info Output Status
  AND FORGET Get Satelite Data

!    Return to Main Menu Choice
!      Forget the choice and return to the Main Menu

RULE PI: Return to Main Menu
IF View Choice IS Return to Main Menu
THEN Present Information
  AND FORGET View Choice

!-------------------------------------------------------------------------
!  Generate Schedule Rules

!    Status is used to mark which rules to use at which time
!         Find Latest NLT Lch Day used to set value of Latest NLT Lch Day
!         Find Satellite is used to find an available satellite
!         Find Vehicle is used to find an available vehicle
!         Find Pad is used to find an available launch pad
!         Advance Schedule Day used to execute Advance Schedue Day rules
!         Load Satellite is used to load a satellite into vehicle
!         Mark Loaded is used to assign the sat to veh and veh to pad
!         Mark Missed Launch is used to list the satellite as missing launch

!-------------------------------------------------------------------------
!  Schedule Resources Menu (SRM)

!   If a schedule is wanted
!     and the schedule parameters are initialized
!   then schedule resources until:
!     Out Of Schedule Time or
!     All Requirements Scheduled

!  Should be able to combine Out of schedule time rule with all requirements
!  scheduled rule using an or condition, but the or condition did not always
!  work during testing.

RULE SRM: Schedule Resources out of schedule time
IF Step = 2
  AND Decision IS Schedule Resources
  AND Schedule Info Entered
  AND Set Run Step
  AND Satellites Counted
  AND Latest NLT Lch Day Established
  AND Out Of Schedule Time
```

```
  AND Schedule Completed Message Displayed
  AND List Available Resources
THEN Option IS Schedule Resources
  AND FORGET Decision
  AND FORGET Schedule Info Entered
  AND FORGET Set Run Step
  AND FORGET Satellites Counted
  AND FORGET Latest NLT Lch Day Established
  AND Latest NLT Lch Day := 0
  AND FORGET Out Of Schedule Time
  AND FORGET All Requirements Scheduled
  AND FORGET Schedule Completed Message Displayed
  AND FORGET List Available Resources
  AND FORGET Comment
  AND FORGET Start Schedule Day
  AND FORGET Ending Schedule Day
  AND FORGET Print Status
  AND Step := 1
  AND CYCLE

RULE SRM: Schedule Resources all requirements scheduled
IF Step = 2
  AND Decision IS Schedule Resources
  AND Schedule Info Entered
  AND Set Run Step
  AND Satellites Counted
  AND Latest NLT Lch Day Established
  AND All Requirements Scheduled
  AND Schedule Completed Message Displayed
  AND List Available Resources
THEN Option IS Schedule Resources
  AND FORGET Decision
  AND FORGET Schedule Info Entered
  AND FORGET Set Run Step
  AND FORGET Satellites Counted
  AND FORGET Latest NLT Lch Day Established
  AND Latest NLT Lch Day := 0
  AND FORGET Out Of Schedule Time
  AND FORGET All Requirements Scheduled
  AND FORGET Schedule Completed Message Displayed
  AND FORGET List Available Resources
  AND FORGET Comment
  AND FORGET Start Schedule Day
  AND FORGET Ending Schedule Day
  AND FORGET Print Status
  AND Step := 1
  AND CYCLE
!    No Schedule Wanted
!      Return to the Main Menu

RULE SRM: No Schedule Wanted
IF Step = 2
```

```
  AND Decision IS Return to Main Menu
THEN Option IS Schedule Resources
  AND FORGET Decision
  AND Step := 1
  AND CYCLE

!  Rules to enter schedule information

RULE Enter Schedule Info
IF Decision IS Schedule Resources
  AND User Info Entered
  AND Get Printer Status
  AND Schedule Information Received
THEN Schedule Info Entered
  AND Current Schedule Day := Start Schedule Day
  AND Next Avail Lch Day := Ending Schedule Day + 1
  AND FORGET User Info Entered
  AND FORGET Get Printer Status
  AND FORGET Schedule Information Received

!-------------------------------------------------------------------------
!     Set-up procedure
!
!       Get the user's name and date of session
!         Use <> comparison instead of ASK so the information will not be
!         requested again during execution of the CYCLE command.
!       Determine if a printout is desired
!       Ask for schedule information

RULE Set up questions
IF User Name <> 24642
  AND Date <> dummy
  AND Comment <> dummy cmt
THEN User Info Entered

!     If printout is desired then do this step

RULE Choose printout
IF Print Status IS Yes
  AND Print Resource Schedule Screen Displayed
  AND Check Printer On
  AND Message := Printing Heading Information
  AND CALL Scrnmsg
  SEND Message
THEN Get Printer Status
  AND PRINT User ID
  AND PRINT Banner
  AND PRINT Printer Program Assumptions
  AND FILE User ID
  AND FILE Banner
  AND FILE Printer Program Assumptions
```

```
!    Assure Print Resource Schedule Screen displayed only once each session

RULE Print resource schedule screen
IF DISPLAY Print Resource Schedule Display
THEN Print Resource Schedule Screen Displayed

!     If no printout is desired then do this step

RULE Choose no printout
IF Print Status IS No
THEN Get Printer Status
  AND FILE User ID
  AND FILE Banner
  AND FILE Printer Program Assumptions

!-------------------------------------------------------------------------
!  Get Schedule Information
!
!    Ask for start and end dates for schedule

RULE Get Schedule Information
IF Start Schedule Day <> -.0987
  AND Valid Start Schedule Day
  AND Ending Schedule Day <> -.0987
  AND Valid Ending Schedule Day
  AND Print Schedule Information
THEN Schedule Information Received
  AND FORGET Valid Start Schedule Day
  AND FORGET Valid Ending Schedule Day
  AND FORGET Print Schedule Information

!  If the start day is greater than 0 and less than the ending day
!   set Current Schedule Day to Start Schedule Day.
!  If the start day is less than 1, print an error message, forget the
!   Start Schedule Day, and return to get a valid day.

RULE Start Day greater than 0
IF Start Schedule Day > 0
THEN Valid Start Schedule Day
  AND Current Schedule Day := Start Schedule Day
ELSE Valid Start Schedule Day
  AND DISPLAY Invalid Start Day
  AND FORGET Start Schedule Day
  AND FORGET Valid Start Schedule Day
  AND CYCLE


!  If the ending day is 0 set end day to very large number to assure
!  satellites run out before the number of days runs out to schedule them.
!
!  If the ending day is greater than the or equal to the start day run
!  schedule until that day is reached.
```

```
!
!  Else print an error message, forget the entered value and return to
!  get the correct value.

RULE Ending Schedule Day equals 0
IF Ending Schedule Day = 0
THEN Valid Ending Schedule Day
  AND Ending Schedule Day := 100000000000

RULE Ending Schedule Day not equal to 0
IF Ending Schedule Day >= Start Schedule Day
THEN Valid Ending Schedule Day
ELSE Valid Ending Schedule Day
  AND DISPLAY Invalid Ending Day
  AND FORGET Ending Schedule Day
  AND FORGET Valid Ending Schedule Day
  AND CYCLE


!  If Schedule is to be printed send information to printer and to file.
!  If not to be printed send information only to file.

RULE Print Schedule Information Print Status Yes
IF Print Status IS Yes
  AND Valid Start Schedule Day
  AND Valid Ending Schedule Day
THEN Print Schedule Information
  AND PRINT Schedule Information
  AND FILE Schedule Information

RULE Print Schedule Information Print Status No
IF Print Status IS No
  AND Valid Start Schedule Day
  AND Valid Ending Schedule Day
THEN Print Schedule Information
  AND FILE Schedule Information

!-------------------------------------------------------------------------
!  Set Run Step

RULE Set Run Step
IF Decision IS Schedule Resources
THEN Set Run Step
  AND Status IS Find Latest NLT Lch Day

!    Satellites Counted Rule
!      Determine the number of satellites possible to launch by calling a
!      DBPAS program to find the total number of records in the SATELITE.DBF
!      file.

RULE Satellites Counted
IF CALL Getsat
```

```
    SEND Sat Record Number
    RETURN Sat Last Record
    RETURN Sat Name
    RETURN Constel
    RETURN Sat Weight
    RETURN Sat Size
    RETURN Sat Orbit Alt
    RETURN Lch Veh 1
    RETURN Lch Veh 2
    RETURN Sat Lch Priority
    RETURN Sat Restrictn
    RETURN Sat Classified
    RETURN Sat Lch With Classified
    RETURN Sat Desire Lch
    RETURN Sat Available
    RETURN Sat NLT Lch
    RETURN Sat Orbit Type
    RETURN Sat Inclinatn
    RETURN Sat Launched
    RETURN Number Of Sat
  THEN Satellites Counted
    AND Number Of Sat To Lch := Number Of Sat

  !    If Status is Find Latest NLT Lch Day and the end of the satellite
  !      records is found, then the lastest NLT Lch Day has been set.

  RULE Find Latest NLT Lch Day of satellites EOF
  IF Status IS Find Latest NLT Lch Day
    AND Latest NLT Lch Day Msg Displayed
    AND Sat Record Number := Sat Record Number + 1
    AND CALL Getsat
    SEND Sat Record Number
    RETURN Sat Last Record
    RETURN Sat Name
    RETURN Constel
    RETURN Sat Weight
    RETURN Sat Size
    RETURN Sat Orbit Alt
    RETURN Lch Veh 1
    RETURN Lch Veh 2
    RETURN Sat Lch Priority
    RETURN Sat Restrictn
    RETURN Sat Classified
    RETURN Sat Lch With Classified
    RETURN Sat Desire Lch
    RETURN Sat Available
    RETURN Sat NLT Lch
    RETURN Sat Orbit Type
    RETURN Sat Inclinatn
    RETURN Sat Launched
    RETURN Number Of Sat
    AND Sat Last Record = e
```

```
THEN Latest NLT Lch Day Established
  AND Sat Record Number := 0
  AND FORGET Latest NLT Lch Day Msg Displayed
  AND Status IS Find Satellite
  AND Message := Looking for an available satellite
  AND CALL Scrnmsg
  SEND Message
  AND CYCLE

!    If the status is Find Latest NLT Lch Day then look through satellite
!      records setting Latest NLT Lch Day to the highest day number found
!      for satellites launchable before the ending schedule day.

RULE Find Latest Lch Day of Satellites later day
IF Status IS Find Latest NLT Lch Day
  AND Sat Last Record = n
  AND Sat Launched = n
  AND Sat NLT Lch > Latest NLT Lch Day
THEN Latest NLT Lch Day Established
  AND Latest NLT Lch Day := Sat NLT Lch
  AND FORGET Latest NLT Lch Day Established
  AND CYCLE
ELSE Latest NLT Lch Day Established
  AND FORGET Latest NLT Lch Day Established
  AND CYCLE

!    Latest NLT Lch Day Msg Displayed rule
!     Required to assure message displayed only once each time executed

RULE Latest NLT Lch Day Msg Displayed
IF Message := Establishing Latest NLT Launch Day for Satellites
  AND CALL Scrnmsg
  SEND Message
THEN Latest NLT Lch Day Msg Displayed

!-------------------------------------------------------------------------
!  Out of schedule time considerations
!
!    You are Out Of Schedule Time
!      if the Current Schedule Day > Ending Schedule Day

!    Out of Schedule Time when looking for a satellite
!      If the ending day is exceeded the end of the schedule has been reached

RULE Satellite Out Of Schedule Time
IF Status IS Find Satellite
  AND Current Schedule Day > Ending Schedule Day
THEN Out Of Schedule Time
  AND CYCLE

!    Out of Schedule Time when looking for a vehicle
!      If the ending day is exceeded then the satellite cannot be launched
```

```
!      look for another satellite

RULE Vehicle not available display on screen
IF Status IS Find Vehicle
  AND Current Schedule Day > Ending Schedule Day
  AND Pad Last Record <> e
  AND Print Status IS No
THEN Out Of Schedule Time
  AND FILE Sat Missed Lch No Vehicle Printer
  AND DISPLAY Sat Missed Lch No Vehicle Display
  AND Status IS Mark Missed Launch
  AND FORGET Out Of Schedule Time
  AND CYCLE

RULE Vehicle not available display on printer
IF Status IS Find Vehicle
  AND Current Schedule Day > Ending Schedule Day
  AND Pad Last Record <> e
  AND Print Status IS Yes
THEN Out Of Schedule Time
  AND FILE Sat Missed Lch No Vehicle Printer
  AND PRINT Sat Missed Lch No Vehicle Printer
  AND Status IS Mark Missed Launch
  AND FORGET Out Of Schedule Time
  AND CYCLE

RULE Vehicle available but no pad for vehicle screen
IF Status IS Find Vehicle
  AND Current Schedule Day > Ending Schedule Day
  AND Pad Last Record = e
  AND Print Status IS No
THEN Out Of Schedule Time
  AND FILE Sat Missed Lch No Pad Printer
  AND DISPLAY Sat Missed Lch No Pad Display
  AND Status IS Mark Missed Launch
  AND FORGET Out Of Schedule Time
  AND CYCLE

RULE Vehicle available but no pad for vehicle printer
IF Status IS Find Vehicle
  AND Current Schedule Day > Ending Schedule Day
  AND Pad Last Record = e
  AND Print Status IS Yes
THEN Out Of Schedule Time
  AND FILE Sat Missed Lch No Pad Printer
  AND PRINT Sat Missed Lch No Pad Printer
  AND Status IS Mark Missed Launch
  AND FORGET Out Of Schedule Time
  AND CYCLE

!    Out of Schedule Time when looking for a Launch Pad
!      If the ending day is exceeded
```

```
!       and all vehicles have not been tried
!     then return to the day vehicle was found
!       and look for another vehicle

RULE No Pad for Vehicle try another vehicle
IF Status IS Find Pad
  AND Current Schedule Day > Ending Schedule Day
  AND Veh Last Record = n
  AND Message := Looking for a different vehicle
  AND CALL Scrnmsg
  SEND Message
THEN Out Of Schedule Time
  AND Current Schedule Day := Day Vehicle Found
  AND Status IS Find Vehicle
  AND FORGET Out Of Schedule Time
  AND CYCLE

!    Out of Schedule Time when looking for a Launch Pad
!      If the ending day is exceeded
!        and all vehicles have been tried
!      then the satellite cannot be launched look for another satellite

RULE Pad not available display on screen
IF Status IS Find Pad
  AND Current Schedule Day > Ending Schedule Day
  AND Veh Last Record = e
  AND Print Status IS No
THEN Out Of Schedule Time
  AND FILE Sat Missed Lch No Pad Printer
  AND DISPLAY Sat Missed Lch No Pad Display
  AND Status IS Mark Missed Launch
  AND FORGET Out Of Schedule Time
  AND CYCLE

RULE Pad not available display on printer
IF Status IS Find Pad
  AND Current Schedule Day > Ending Schedule Day
  AND Veh Last Record = e
  AND Print Status IS Yes
THEN Out Of Schedule Time
  AND FILE Sat Missed Lch No Pad Printer
  AND PRINT Sat Missed Lch No Pad Printer
  AND Status IS Mark Missed Launch
  AND FORGET Out Of Schedule Time
  AND CYCLE

!-------------------------------------------------------------------------
!  No more satellites to launch
!
!    All Requirements Scheduled Rule (ARS)
!    If All Satellites are launched
!      or the NLT launch date for all remaining satellites is < current day
```

```
!    Then All Requirements are Scheduled

RULE ARS: All Possible Satellites Launched
IF Status IS Find Satellite
  AND Number Of Sat To Lch = 0
  OR Latest NLT Lch Day < Current Schedule Day
THEN All Requirements Scheduled

!-------------------------------------------------------------------------
!  Find available satellite
!
!    A Satellite is Available
!      if it is not launched
!      and it has the same or higher priority as the Current Priority
!      and its desired lch day is less than or equal to the current day
!      and its NLT day is greater than or equal to the current day

RULE Find Satellite Available
IF Status IS Find Satellite
  AND Sat Record Number := Sat Record Number + 1
  AND CALL Getsat
  SEND Sat Record Number
  RETURN Sat Last Record
  RETURN Sat Name
  RETURN Constel
  RETURN Sat Weight
  RETURN Sat Size
  RETURN Sat Orbit Alt
  RETURN Lch Veh 1
  RETURN Lch Veh 2
  RETURN Sat Lch Priority
  RETURN Sat Restrictn
  RETURN Sat Classified
  RETURN Sat Lch With Classified
  RETURN Sat Desire Lch
  RETURN Sat Available
  RETURN Sat NLT Lch
  RETURN Sat Orbit Type
  RETURN Sat Inclinatn
  RETURN Sat Launched
  RETURN Number Of Sat
  AND Sat Last Record = n
  AND Sat Launched = n
  AND Sat Lch Priority <= Current Priority ! lower number is higher priority
  AND Sat Desire Lch <= Current Schedule Day
  AND Sat NLT Lch >= Current Schedule Day
THEN All Requirements Scheduled
  AND FORGET All Requirements Scheduled
  AND Status IS Find Vehicle
  AND Schedule Day Sat Available := Current Schedule Day
  AND Message := Looking for an available vehicle
  AND CALL Scrnmsg
```

```
  SEND Message
  AND Next Avail Lch Day := Ending Schedule Day + 1
  AND CYCLE

!    A satellite is available at a lower priority
!    If it is available for launch
!      and it has a priority less than the current priority

RULE Sat Avail With Lower Priority
IF Sat Last Record = n
  AND Sat Launched = n
  AND Sat Lch Priority > Current Priority
  AND Sat Desire Lch <= Current Schedule Day
  AND Sat NLT Lch >= Current Schedule Day
THEN All Requirements Scheduled
  AND Sat Avail With Lower Priority
  AND FORGET All Requirements Scheduled
  AND CYCLE

!    If All satellites have been checked and none are available at the
!      Current Priority then decrease the priority level and check again.

RULE Satellite Available Decrease Priority
IF Status IS Find Satellite
  AND Sat Last Record = e
  AND Sat Avail With Lower Priority
  AND Current Priority < Lowest Priority
THEN All Requirements Scheduled
  AND Sat Record Number := 0
  AND Current Priority := Current Priority + 1  ! higher is lower priority
  AND FORGET Sat Avail With Lower Priority
  AND FORGET All Requirements Scheduled
  AND CYCLE

!    If All satellites have been checked
!      and none are available at the lowest priority
!    then advance Current Schedule Day
!      and start with highest priority
!      and check all satellites again

RULE Satellite Available Advance Current Schedule Day
IF Status IS Find Satellite
  AND Sat Last Record = e
  AND Next Avail Lch Day >= Current Schedule Day
  OR Current Priority = Lowest Priority
  AND Advance Schedule Day
THEN All Requirements Scheduled
  AND Sat Record Number := 0
  AND Current Priority := 1
  AND FORGET Advance Schedule Day
  AND FORGET All Requirements Scheduled
  AND CYCLE
```

```
!    No Satellite match and not at end of records

RULE No Match Check Next Record
IF Status IS Find Satellite
  AND Sat Last Record = n
  AND Check Next Available Sat Day
THEN All Requirements Scheduled
  AND FORGET Check Next Available Sat Day
  AND FORGET All Requirements Scheduled
  AND CYCLE

!    Find next available satellite launch day after current schedule day

RULE Next Available Sat Day
IF Sat Last Record = n
  AND Sat Launched = n
  AND Sat Desire Lch > Current Schedule Day
  AND Sat Desire Lch <= Ending Schedule Day
  AND Sat Desire Lch < Next Avail Lch Day
THEN Check Next Available Sat Day
  AND Next Avail Lch Day := Sat Desire Lch

!    Sat not to be considered for next available launch day

RULE Next Available Sat Day Skip This Sat
IF Sat Last Record = n
  AND Sat Launched <> n
  OR Sat Desire Lch <= Current Schedule Day
  OR Sat Desire Lch > Ending Schedule Day
  OR Sat Desire Lch >= Next Avail Lch Day
THEN Check Next Available Sat Day

!-------------------------------------------------------------------------
!    Advance Schedule Day Rule
!    If the current schedule day is <= the ending schedule day
!    then set the current schedule day to next available lch day

RULE Advance Schedule Day
IF Current Schedule Day <= Ending Schedule Day
  AND Next Avail Lch Day > Current Schedule Day
THEN Advance Schedule Day
  AND Current Schedule Day := Next Avail Lch Day

!    Advance day when next avail = current day

RULE Advance Schedule Day By One Day
IF Current Schedule Day <= Ending Schedule Day
  AND Next Avail Lch Day <= Current Schedule Day
THEN Advance Schedule Day
  AND Current Schedule Day := Current Schedule Day + 1
```

```
!    Advance day when next available = ending schedule day

RULE Advance Schedule Day to ending schedule day
IF Current Schedule Day < Ending Schedule Day
  AND Next Avail Lch Day = Ending Schedule Day
THEN Advance Schedule Day
  AND Current Schedule Day := Next Avail Lch Day

!    Advance Schedule Day error rule
!     If this rule is executed the out of schedule time rule has failed

RULE Advance Schedule Day Error
IF Current Schedule Day > Ending Schedule Day
THEN Advance Schedule Day
  AND DISPLAY The out of schedule time rule has failed.

!-------------------------------------------------------------------------
!    Find available vehicle Rule
!    A vehicle is available
!      If the vehicle available day is <= current schedule day
!      and the vehicle is capable of the required orbit
!      and the vehicle has sufficient boost capacity
!      and the vehicle has sufficient size capacity
!      and the vehicle is compatable with the satellite

RULE Find Available Vehicle
IF Status IS Find Vehicle
  AND Veh Record Number := Veh Record Number + 1
  AND CALL Getveh
  SEND  Veh Record Number
  RETURN Veh Last Record
  RETURN Vehicle Name
  RETURN Vehicle Type
  RETURN Boost Capability
  RETURN Payload Size
  RETURN Orbit Altitude Cap
  RETURN Veh Launch Facility
  RETURN Mission Duration
  RETURN Veh Turn Time
  RETURN Veh First Available
  RETURN Veh Next Available
  AND Veh Last Record = n
  AND Veh Next Available <= Current Schedule Day
  AND Orbit Altitude Cap >= Sat Orbit Alt
  AND Boost Capability >= Sat Weight
  AND Payload Size >= Sat Size
  AND Vehicle Type = Lch Veh 1
  OR Vehicle Type = Lch Veh 2
THEN All Requirements Scheduled
  AND FORGET All Requirements Scheduled
  AND Veh Last Tried := Veh Record Number
  AND Status IS Find Pad
```

```
  AND Message := Looking for an available Launch Pad
  AND CALL Scrnmsg
  SEND Message
  AND Day Vehicle Found := Current Schedule Day
  AND Next Avail Lch Day := Ending Schedule Day + 1
  AND CYCLE

!    If All Vehicles have been checked and none are available then
!      advance the current day and check again.

RULE Vehicle Available Advance Schedule Day
IF Status IS Find Vehicle
  AND Veh Last Record = e
  AND Next Avail Lch Day >= Current Schedule Day
  AND Pad Last Record <> e
  AND Advance Schedule Day
THEN All Requirements Scheduled
  AND FORGET Advance Schedule Day
  AND FORGET All Requirements Scheduled
  AND Veh Record Number := 0
  AND Next Avail Lch Day := Ending Schedule Day + 1
  AND CYCLE

RULE Veh Available Advance Schedule Day Alt Veh Search
IF Status IS Find Vehicle
  AND Veh Last Record = e
  AND Next Avail Lch Day >= Current Schedule Day
  AND Pad Last Record = e
  AND Advance Schedule Day
THEN All Requirements Scheduled
  AND FORGET Advance Schedule Day
  AND FORGET All Requirements Scheduled
  AND Veh Record Number := Veh Last Tried
  AND Next Avail Lch Day := Ending Schedule Day + 1
  AND CYCLE

!    No matching vehicle found check next record

RULE No Match Check Next Veh Record
IF Status IS Find Vehicle
  AND Veh Last Record = n
  AND Check Next Available Veh Day
THEN All Requirements Scheduled
  AND FORGET Check Next Available Veh Day
  AND FORGET All Requirements Scheduled
  AND CYCLE

!    Next Available Vehicle Day
!      If the vehicle is available after the current day
!        and before the end of the schedule
!        and before the current next available day
!        and it can meet the required orbit
```

```
!       and it can launch the satellite
!     then set a new next available day
!     else check the next vehicle

RULE Next Available Vehicle Day
IF Veh Last Record = n
  AND Veh Next Available > Current Schedule Day
  AND Veh Next Available <= Ending Schedule Day
  AND Veh Next Available < Next Avail Lch Day
  AND Orbit Altitude Cap >= Sat Orbit Alt
  AND Boost Capability >= Sat Weight
  AND Payload Size >= Sat Size
  AND Vehicle Type = Lch Veh 1
  OR Vehicle Type = Lch Veh 2
THEN Check Next Available Veh Day
  AND Next Avail Lch Day := Veh Next Available
ELSE Check Next Available Veh Day

!-------------------------------------------------------------------------
!    Find an available Launch Pad Rule
!    A Pad is available
!      If the pad next available day is <= current schedule day
!      and the pad is compatable with the vehicle
!      and the pad and vehicle are at the same location

RULE Find Available Pad
IF Status IS Find Pad
  AND Pad Record Number := Pad Record Number + 1
  AND CALL Getpad
  SEND Pad Record Number
  RETURN Pad Last Record
  RETURN Pad Name
  RETURN Pad Type
  RETURN Pad Launch Facility
  RETURN Pad Turn Time
  RETURN Pad First Available
  RETURN Pad Next Available
  AND Pad Last Record = n
  AND Pad Type = Vehicle Type
  AND Pad Launch Facility = Veh Launch Facility
  AND Pad Next Available <= Current Schedule Day
THEN All Requirements Scheduled
  AND Status IS Load Satellite
  AND FORGET All Requirements Scheduled
  AND Next Avail Lch Day := Ending Schedule Day + 1
  AND CYCLE

!    If all pads have been checked and none are available then advance
!      the current day and check again.

RULE Pad Available Advance Schedule Day
IF Status IS Find Pad
```

```
  AND Pad Last Record = e
  AND Next Avail Lch Day >= Current Schedule Day
  AND Advance Schedule Day
THEN All Requirements Scheduled
  AND Pad Record Number := 0
  AND FORGET Advance Schedule Day
  AND FORGET All Requirements Scheduled
  AND CYCLE

!    No Match check next available pad record

RULE No Match Check Next Pad Record
IF Status IS Find Pad
  AND Pad Last Record = n
  AND Check Next Available Pad Day
THEN All Requirements Scheduled
  AND FORGET Check Next Available Pad Day
  AND FORGET All Requirements Scheduled
  AND CYCLE

!    Next Available Pad Day

RULE Next Available Pad Day
IF Pad Last Record = n
  AND Pad Next Available > Current Schedule Day
  AND Pad Next Available <= Ending Schedule Day
  AND Pad Next Available < Next Avail Lch Day
  AND Pad Type = Vehicle Type
  AND Pad Launch Facility = Veh Launch Facility
THEN Check Next Available Pad Day
  AND Next Avail Lch Day := Pad Next Available
ELSE Check Next Available Pad Day

!-------------------------------------------------------------------------
!    Load Satellite Rule
!    A compatable satellite can be loaded
!      If the satellite NLT launch day is >= current schedule day
!       Lines of code commented out are for use with multiple satellites
!       per vehicle which is not currently implemented

RULE Load Satellite Display on screen
IF Status IS Load Satellite
  AND Sat NLT Lch >= Current Schedule Day
  AND Print Status IS No
THEN All Requirements Scheduled
!  AND FILE Sat Loaded Display
!  AND DISPLAY Sat Loaded Display
  AND Status IS Launch Satellite
    ! For multiple sat per vehicle use Status IS Mark Loaded
  AND FORGET All Requirements Scheduled
  AND CYCLE
```

```
RULE Load Satellite Display on printer
IF Status IS Load Satellite
  AND Sat NLT Lch >= Current Schedule Day
  AND Print Status IS Yes
THEN All Requirements Scheduled
!  AND FILE Sat Loaded Display
!  AND PRINT Sat Loaded Display
  AND Status IS Launch Satellite
    ! For multiple sat per vehicle use Status IS Mark Loaded
  AND FORGET All Requirements Scheduled
  AND CYCLE

!    Load Satellite Rule Launch Too Late
!    The satellite cannot be loaded
!      If the satellite NLT launch day is < current schedule day

RULE Load Satellite Too Late display on the screen
IF Status IS Load Satellite
  AND Sat NLT Lch < Current Schedule Day
  AND Print Status IS No
THEN All Requirements Scheduled
  AND FILE Sat Missed Lch Printer Display
  AND DISPLAY Sat Missed Lch Display
  AND Status IS Mark Missed Launch
  AND FORGET All Requirements Scheduled
  AND CYCLE

RULE Load Satellite Too Late display on the printer
IF Status IS Load Satellite
  AND Sat NLT Lch < Current Schedule Day
  AND Print Status IS Yes
THEN All Requirements Scheduled
  AND FILE Sat Missed Lch Printer Display
  AND PRINT Sat Missed Lch Printer Display
  AND Status IS Mark Missed Launch
  AND FORGET All Requirements Scheduled
  AND CYCLE

!-------------------------------------------------------------------------
!  Mark Missed Launch Rule
!    Mark satellite missing launch and reset variables to starting values to
!    search for the next avaiable satellite.

RULE Mark Missed Launch
IF Status IS Mark Missed Launch
THEN All Requirements Scheduled
  AND CALL Marksat
  SEND X_char
  SEND Sat_Record Number
  AND Current Schedule Day := Schedule Day Sat Available
  AND FORGET All Requirements Scheduled
  AND FORGET Set Run Step
```

```
  AND FORGET Latest NLT Lch Day Established
  AND Latest NLT Lch Day := 1
  AND FORGET Out Of Schedule Time
  AND Sat Record Number := 0
  AND Veh Record Number := 0
  AND Pad Record Number := 0
  AND Veh Last Tried := 0
  AND Veh Last Record := n
  AND Pad Last Record := n
  AND CYCLE

!-------------------------------------------------------------------------
!    Mark Loaded Rule
!      Mark the satellite as assigned to the vehicle and the vehicle as
!      assigned to the pad.  Reset pointers for next satellite.
!       This rule to be used for multiple satellites per vehicle not currently
!       implemented

!RULE Mark Loaded
!IF Status IS Mark Loaded
!  AND Mark Sat Record Loaded
!THEN All Requirements Scheduled
!  AND Loaded Sat 1 Record Number := Sat Record Number
!  AND Loaded Sat 1 NLT Lch Day := Sat NLT Lch
!  AND FORGET All Requirements Scheduled
!  AND FORGET Mark Sat Record Loaded
! AND Status IS Find Second Sat
!  AND Status IS Launch Satellite
!  AND CYCLE

!    Mark Sat Record Loaded Rule
!      This rule to be used for multiple satellites per vehicle
!      not currently implemented

!RULE Mark Sat Record Loaded
!IF Status IS Mark Loaded
!THEN Mark Sat Record Loaded
!  AND CALL Marksat
!  SEND M_char
!  SEND Sat Record Number

!-------------------------------------------------------------------------
!    Launch Satellite Rule
!    If status is launch sat
!    Then
!      Mark Sat Launched
!      Set Vehicle next available day to launch + turn time
!      Set Pad next available day to launch + turn time
!      Reset variables to starting values to search for next available sat

RULE Launch Satellite display on the screen
IF Status IS Launch Satellite
```

```
  AND Print Status IS No
THEN All Requirements Scheduled
  AND CALL Marksat
  SEND L_char
  SEND Sat Record Number
  AND CALL Markveh
  SEND Current Schedule Day
  SEND Veh Record Number
  AND CALL Markpad
  SEND Current Schedule Day
  SEND Pad Record Number
  AND FILE Sat Launched Printer Display
  AND DISPLAY Sat Launched Display
  AND FORGET All Requirements Scheduled
  AND FORGET Set Run Step
  AND FORGET Latest NLT Lch Day Established
  AND Latest NLT Lch Day := 0
  AND FORGET Out Of Schedule Time
  AND Current Schedule Day := Schedule Day Sat Available
  AND Sat Record Number := 0
  AND Veh Record Number := 0
  AND Pad Record Number := 0
  AND Veh Last Tried := 0
  AND Veh Last Record := n
  AND Pad Last Record := n
  AND Number Of Sat To Lch := Number Of Sat To Lch - 1
  AND CYCLE


RULE Launch Satellite display on the printer
IF Status IS Launch Satellite
  AND Print Status IS Yes
THEN All Requirements Scheduled
  AND CALL Marksat
  SEND L_char
  SEND Sat Record Number
  AND CALL Markveh
  SEND Current Schedule Day
  SEND Veh Record Number
  AND CALL Markpad
  SEND Current Schedule Day
  SEND Pad Record Number
  AND FILE Sat Launched Printer Display
  AND PRINT Sat Launched Printer Display
  AND FORGET All Requirements Scheduled
  AND FORGET Set Run Step
  AND FORGET Latest NLT Lch Day Established
  AND Latest NLT Lch Day := 0
  AND FORGET Out Of Schedule Time
  AND Current Schedule Day := Schedule Day Sat Available
  AND Sat Record Number := 0
  AND Veh Record Number := 0
```

```
  AND Pad Record Number := 0
  AND Veh Last Tried := 0
  AND Veh Last Record := n
  AND Pad Last Record := n
  AND Number Of Sat To Lch := Number Of Sat To Lch - 1
  AND CYCLE

!-------------------------------------------------------------------------
!     Schedule Completed Message Displayed Rule
!       Required to be sure the message is only displayed once each scheduling
!       cycle.

RULE Schedule Completed Message Displayed
IF DISPLAY Schedule Completed Display
THEN Schedule Completed Message Displayed
  AND Status IS List Unlaunched Sat
  AND Sat Record Number := 0
  AND Message := Looking for unlaunched Satellites after schedule completion
  AND CALL Scrnmsg
  SEND Message

!=========================================================================
!  The following rules list the unlaunched satellites and the remaining
!  resources after the schedule matching process is completed

!-------------------------------------------------------------------------
!     Display unlaunched Satellites Rule
!      If the satellite is not marked as launched
!      Then list it as not launched

RULE List Unlaunched Satellites
IF Status IS List Unlaunched Sat
  AND Sat Record Number := Sat Record Number + 1
  AND CALL Getsat
  SEND Sat Record Number
  RETURN Sat Last Record
  RETURN Sat Name
  RETURN Constel
  RETURN Sat Weight
  RETURN Sat Size
  RETURN Sat Orbit Alt
  RETURN Lch Veh 1
  RETURN Lch Veh 2
  RETURN Sat Lch Priority
  RETURN Sat Restrictn
  RETURN Sat Classified
  RETURN Sat Lch With Classified
  RETURN Sat Desire Lch
  RETURN Sat Available
  RETURN Sat NLT Lch
  RETURN Sat Orbit Type
  RETURN Sat Inclinatn
```

```
  RETURN Sat Launched
  RETURN Number Of Sat
  AND Print Status IS Yes
  AND Sat Desire Lch >= Start Schedule Day
  AND Sat Desire Lch <= Ending Schedule Day
  AND Sat Last Record = n
  AND Sat Launched <> 1
  AND Unlaunched Sat Heading Printed
  AND Unlaunched Sat File Heading
THEN List Available Resources
  AND PRINT Unlaunched Sat Display
  AND FILE Unlaunched Sat Display
  AND FORGET List Available Resources
  AND CYCLE

RULE List Unlaunched Sat on the screen
IF Status IS List Unlaunched Sat
  AND Print Status IS No
  AND Sat Desire Lch >= Start Schedule Day
  AND Sat Desire Lch <= Ending Schedule Day
  AND Sat Last Record = n
  AND Sat Launched <> 1
  AND Unlaunched Sat File Heading
THEN List Available Resources
  AND DISPLAY Unlaunched Sat Screen Display
  AND FILE Unlaunched Sat Display
  AND Message := Looking for next unlaunched Satellite
  AND CALL Scrnmsg
  SEND Message
  AND FORGET List Available Resources
  AND CYCLE

RULE List Unlaunched Satellites skip launched sat
IF Status IS List Unlaunched Sat
  AND Sat Last Record = n
  AND Sat Launched = 1
  OR Sat Desire Lch < Start Schedule Day
  OR Sat Desire Lch > Ending Schedule Day
THEN List Available Resources
  AND FORGET List Available Resources
  AND CYCLE

RULE List Unlaunched Satellites EOF
IF Status IS List Unlaunched Sat
  AND Sat Last Record = e
THEN List Available Resources
  AND Status IS List Available Vehicles
  AND Veh Record Number := 0
  AND Message := Looking for available Vehicles after schedule completion
  AND CALL Scrnmsg
  SEND Message
  AND Sat Record Number := 0
```

```
  AND FORGET List Available Resources
  AND CYCLE

! Unlaunched Satellite Heading Printed Rule

RULE Unlaunched Satellite Heading Printed
IF PRINT Unlaunched Sat Heading Display
THEN Unlaunched Sat Heading Printed

! Unlaunched Satellite File Heading Printed Rule

RULE Unlaunched Satellite File Heading Printed
IF FILE Unlaunched Sat Heading Display
THEN Unlaunched Sat File Heading

!-------------------------------------------------------------------------
!    List Resources Available at end of schedule
!     If the resource next available date is <= ending schedule day
!     Then list resource as available

RULE List Available Vehicles
IF Status IS List Available Vehicles
  AND Veh Record Number := Veh Record Number + 1
  AND CALL Getveh
  SEND  Veh Record Number
  RETURN Veh Last Record
  RETURN Vehicle Name
  RETURN Vehicle Type
  RETURN Boost Capability
  RETURN Payload Size
  RETURN Orbit Altitude Cap
  RETURN Veh Launch Facility
  RETURN Mission Duration
  RETURN Veh Turn Time
  RETURN Veh First Available
  RETURN Veh Next Available
  AND Print Status IS Yes
  AND Veh Last Record = n
  AND Veh Next Available <= Ending Schedule Day
  AND Available Vehicle Heading Printed
  AND Available Vehicle File Heading
THEN List Available Resources
  AND PRINT Available Vehicle Display
  AND FILE Available Vehicle Display
  AND FORGET List Available Resources
  AND CYCLE

RULE List Available Vehicle on screen
IF Status IS List Available Vehicles
  AND Print Status IS No
  AND Veh Last Record = n
  AND Veh Next Available <= Ending Schedule Day
```

```
  AND Available Vehicle File Heading
THEN List Available Resources
  AND DISPLAY Available Vehicle Screen Display
  AND FILE Available Vehicle Display
  AND Message := Looking for next available Vehicle
  AND CALL Scrnmsg
  SEND Message
  AND FORGET List Available Resources
  AND CYCLE

RULE List Available Vehicles skip ones not available
IF Status IS List Available Vehicles
  AND Veh Last Record = n
  AND Veh Next Available > Ending Schedule Day
THEN List Available Resources
  AND FORGET List Available Resources
  AND CYCLE

RULE List Available Vehicles EOF
IF Status IS List Available Vehicles
  AND Veh Last Record = e
THEN List Available Resources
  AND Status IS List Available Pads
  AND Pad Record Number := 0
  AND Message := Looking for available Launch Pads after schedule completion
  AND CALL Scrnmsg
  SEND Message
  AND Veh Record Number := 0
  AND FORGET List Available Resources
  AND CYCLE

!    Available Vehicle Heading Printed

RULE Available Vehicle Heading Printed
IF PRINT Available Vehicle Heading Display
THEN Available Vehicle Heading Printed

!    Available Vehicle File Heading

RULE Available Vehicle File Heading Printed
IF FILE Available Vehicle Heading Display
THEN Available Vehicle File Heading

!-------------------------------------------------------------------------
!    List Available Pads
!     If the pad next available day is <= ending schedule day
!     Then list pad as available

RULE List Available Pads on the printer
IF Status IS List Available Pads
  AND Pad Record Number := Pad Record Number + 1
  AND CALL Getpad
```

```
  SEND Pad Record Number
  RETURN Pad Last Record
  RETURN Pad Name
  RETURN Pad Type
  RETURN Pad Launch Facility
  RETURN Pad Turn Time
  RETURN Pad First Available
  RETURN Pad Next Available
  AND Print Status IS Yes
  AND Pad Last Record = n
  AND Pad Next Available <= Ending Schedule Day
  AND Available Pads Heading Printed
  AND Available Pads File Heading
THEN List Available Resources
  AND PRINT Available Pad Display
  AND FILE Available Pad Display
  AND FORGET List Available Resources
  AND CYCLE

RULE List Available Pads on the screen
IF Status IS List Available Pads
  AND Print Status IS No
  AND Pad Last Record = n
  AND Pad Next Available <= Ending Schedule Day
  AND Available Pads File Heading
THEN List Available Resources
  AND DISPLAY Available Pad Screen Display
  AND FILE Available Pad Display
  AND Message := Looking for next available Pad
  AND CALL Scrnmsg
  SEND Message
  AND FORGET List Available Resources
  AND CYCLE

RULE List Available Pads skip ones not available
IF Status IS List Available Pads
  AND Pad Last Record = n
  AND Pad Next Available > Ending Schedule Day
THEN List Available Resources
  AND FORGET List Available Resources
  AND CYCLE

RULE List Available Pads EOF
IF Status IS List Available Pads
  AND Pad Last Record = e
THEN List Available Resources
  AND Status IS Quit List
  AND Pad Record Number := 0

!    Available Pads Heading Printed

RULE Available Pads Heading Printed
```

```
IF PRINT Available Pads Heading Display
THEN Available Pads Heading Printed

RULE Available Pads File Heading
IF FILE Available Pads Heading Display
THEN Available Pads File Heading

!-------------------------------------------------------------------------
!  Choose Which To Reset (CWR) Rules

!  Present option to reset all db files, or any particular file.

!  Don't want to reset any files return to main menu

RULE CWR: Return to Main Menu
IF Reset File IS Return to Main Menu
THEN Choose Which To Reset
  AND FORGET Reset File

!  Reset all files choice

RULE CWR: Reset All Files
IF Reset File IS All Data Base Files
THEN Choose Which To Reset
  AND CALL Resetveh
  AND CALL Resetpad
  AND CALL Resetsat
  AND FORGET Reset File

!  Reset Vehicle file

RULE CWR: Reset Vehicle File
IF Reset File IS Vehicle File
THEN Choose Which To Reset
  AND CALL Resetveh
  AND FORGET Reset File

!  Reset Pad file

RULE CWR: Reset Pad File
IF Reset File IS Pad File
THEN Choose Which To Reset
  AND CALL Resetpad
  AND FORGET Reset File

!  Reset Satelite File

RULE CWR: Reset Satelite File
IF Reset File IS Satelite File
THEN Choose Which To Reset
  AND CALL Resetsat
  AND FORGET Reset File
```

```
!=============================================================================
!  Information display text
!=============================================================================

!DISPLAY text

!-----------------------------------------------------------------------------
!  Display information for Main Menu
!-----------------------------------------------------------------------------

DISPLAY Program Assumptions Display
*****************************************************************************
                          Program Assumptions

    This program uses all deterministic computations.  This means if a
    vehicle launches it always successfully places its payload into orbit.
    If a satellite takes two weeks checkout time it will always take exactly
    two weeks to checkout.  This reasoning was used to simplify the program
    operation.

    All specific parameters for vehicles, launch pads, and satellites are
    entered by the user.  This information is used by the program to provide
    a possible matching of launch resources to launch requirements.

    The schedule provided is not necessarily the optimum shedule.  It is a
    feasible schedule.  One which meets all the constraints established by
    the user entered information and the scheduling rules of the program.



            Press Function Key 2 CONT to return to the Main Menu

DISPLAY Load Saved Parameters Display
*****************************************************************************
                        Load Saved Parameters

    The Program uses the files VEHICLE.DBF, PAD.DBF, and SATELITE.DBF to
    store all the required resource and requirement information.  You may
    create as many data base files as you desire using the Edit Data Base
    option from the Insight 2+ Main Menu.  Be sure to rename each created
    file to some name other than the three used by the LRS program.

    Choose the file you wish to use before running the LRS program and give
    it the proper name using the DOS ren command.  The LRS program will then
    use the desired file during the program execution.

    See the LRS User Manual for details on creating and changing db files.
```

Press Function Key 2 CONT to return to the Main Menu

DISPLAY Enter New Parameters Display

****

Enter New Parameters

New Parameters are entered by adding information to the data base files which store the information used by the program scheduler.

There are three files which store the required information. The VEHCILE db file contains all the information about the vehicle resources available to be used to meet the launch requirements. The PAD db file contains all the information about launch pads to be used. The SATELITE db file contains all the satellite information.

This information may be updated using the Edit Data Base choice from the Main Menu of the Insight 2+ program. A new data base may also be created using this program option. See the LRS User Manual for details.

The program always uses the files named VEHICLE.DBF, PAD.DBF, and SATELITE.DBF during program execution. See the LRS User Manual for details on using multiple data files.

Press Function Key 2 CONT to continue the program

DISPLAY Change Parameters Display

****

Change Parameters

Parameters are changed by editing the data base files used during program execution. The VEHICLE file is used to store the vehicle resource information. The PAD file is used to store the launch pad resource information. The SATELITE file is used to store the launch requirements information. These files may be changed using the Edit Data Base File option from the Insight 2+ Main Menu.

Information can be easily added to the files or changed using the ADD and EDIT functions of the Edit DB file program. Complete new files may be created using the Edit DB file program for quick information changes. See the LRS User Manual for details on creating new files and changing which files are used by the LRS program.

Press Function Key 2 CONT to choose which items to change

DISPLAY Save Parameters Display

****

Save Parameters

The LRS program automatically updates the data base files during program execution. This allows another session to be run assuming the previous schedules created are completed. This is useful to continue a schedule past the original end schedule date.

If a new schedule is desired use the Reset Data Base option to restore all the parameters changed during execution to their original values.

Press Function Key 2 CONT to return to the Main Menu

DISPLAY Generate Schedule Display

****

Generate Schedule

Generate Schedule will attempt to match the entered resources with the entered launch requirements. The program will continue the matching process until all requirements are met, or a required resource is no longer available.

A schedule showing how the resources and requirements were matched can be displayed on the screen or sent to the printer. The schedule is also stored in the file SCHEDULE.DAT which may be printed using the DOS TYPE command after exiting the program.

After all scheduling is completed. All satellites available for launch but not launched will be listed. All vehicles and launch pads available for launch at the end of the schedule will be listed.

IMPORTANT The program only stores the last schedule created. The SCHEDULE.DAT file must be renamed from DOS to keep it for future use.

Press Function Key 2 CONT to begin resource and requirement matching

DISPLAY Print Resource Schedule Display

****

Print Resource Schedule

Print Resurce Schedule will send a copy of the current resource and requirement matching schedule to the printer. A copy of this schedule is also saved under the file name SCHEDULE.DAT during program execution

and may be printed using the DOS TYPE command.

Only the last schedule generated is saved by the program and may be printed.

Press Function Key 2 CONT for the Main Menu

DISPLAY Save Resource Schedule Display

****

Save Resource Schedule

The last resource and requirement matching schedule is saved under the file name SCHEDULE.DAT.

To save this file for future use;
EXIT the program using Function Key 10 EXIT.
Use the DOS command REN SCHEDULE.DAT filename.ext
where filename.ext is the file name and extension you desire

The program automatically erases any previous data in the SCHEDULE.DAT file when the program is started.

Press Function Key 10 EXIT to exit to DOS to save the file

Press Function Key 2 CONT to return to the Main Menu

DISPLAY Reset DB Display

****

Reset Data Base Files

The Reset Data Base Files option allows returning the db files to the condition of no resources used and no satellites launched. Use this option when a new schedule is desired.

This option marks all resources as available on their first available date listed in the data base, and marks all satellites as not launched.

If Reset Data Base Files is not used prior to starting the scheduling process, the next available dates for resources will be the dates as

updated during the last schedule run. This is useful to continue a schedule past the ending date used for the previous run. All satellites launched during the last session will still be considered launched by the LRS scheduler.

Press Function Key 2 CONT to continue the program

DISPLAY Printer On Msg

Be sure the Printer is turned ON.

Press Function Key 2 CONT to begin printing

```
!-------------------------------------------------------------------------
!  Main Menu Sub Rule Displays
!-------------------------------------------------------------------------
!
```

DISPLAY Invalid Start Day

The Starting Day must be greater than 0 and less than the Ending Day.

Entered Starting Day was [Start Schedule Day (4,0)]

Press Function Key 2 CONT to reenter the Start Day

DISPLAY Invalid Ending Day

The Ending Day must be greater than or equal to the start day, or

0 if a schedule is desired to launch all possible satellites.

Entered Starting Day was [Start Schedule Day (4,0)]

Entered Ending Day was [Ending Schedule Day (4,0)]

Press Function Key 2 CONT to reenter the Ending Day

DISPLAY Schedule Information

The Starting Day of the Schedule is [Start Schedule Day (6,0)].

The Ending Day of the Schedule is [Ending Schedule Day (10,0)].

DISPLAY Last Record Display

The last record has been shown.

Press Function Key 2 CONT to continue the program

DISPLAY Schedule Completed Display

The Resource and Requirement matching process has been completed.

Press Function Key 2 CONT to list available resources

DISPLAY Unlaunched Sat Display

[Sat Name] with desired launch day of [Sat Desire Lch (6,0)] and NLT launch day of [Sat NLT Lch (6,0)] and priority of [Sat Lch Priority].

DISPLAY Unlaunched Sat Screen Display

[Sat Name] with desired launch day of [Sat Desire Lch (6,0)] and NLT launch day of [Sat NLT Lch (6,0)] and priority of [Sat Lch Priority] was not launched.

Press Function Key 2 CONT for next listing

DISPLAY Available Vehicle Display

[Vehicle Name] available on day [Veh Next Available (6,0)] was available at the end of the schedule.

DISPLAY Available Vehicle Screen Display

[Vehicle Name] available on day [Veh Next Available (6,0)] was available at the end of the schedule.

Press Function Key 2 CONT for next listing

DISPLAY Available Pad Display

[Pad Name] able to launch [Pad Type] available on day [Pad Next Available(6,0)].

DISPLAY Available Pad Screen Display

[Pad Name] able to launch [Pad Type] available on day [Pad Next Available(6,0)] was available at the end of the schedule.

Press Function Key 2 CONT for next listing

DISPLAY Unlaunched Sat Heading Display

The following Satellites were not launched during the schedule period.

DISPLAY Available Vehicle Heading Display

The following Vehicles were available at the end of the schedule.

DISPLAY Available Pads Heading Display

The following Pads were available at the end of the schedule.

DISPLAY Sat Missed Lch Display

Satellite [Sat Name] missed launch due to no resource available by LNT Launch day [Sat NLT Lch (6,0)].

Press Function Key 2 CONT to resume the scheduling process

DISPLAY Sat Missed Lch Printer Display

Satellite [Sat Name] missed launch due to no resource available by LNT Launch day [Sat NLT Lch (6,0)].

DISPLAY Sat Missed Lch No Vehicle Display

Satellite [Sat Name] missed launch due to no vehicle available by the ending schedule day. The satellite was available on day [Schedule Day Sat Available (6,0)].

Press Function Key 2 CONT to continue scheduling process

DISPLAY Sat Missed Lch No Vehicle Printer

Satellite [Sat Name] missed launch due to no vehicle available by the ending schedule day. The satellite was available on day [Schedule Day Sat Available (6,0)].

DISPLAY Sat Missed Lch No Pad Display

Satellite [Sat Name] missed launch due to no launch pad available by the ending schedule day. The satellite was available on day [Schedule Day Sat Available (6,0)].

Press Function Key 2 CONT to continue scheduling process

DISPLAY Sat Missed Lch No Pad Printer

```
     Satellite [Sat Name] missed launch due to no launch pad available
       by the ending schedule day.  The satellite was available on
       day [Schedule Day Sat Available (6,0)].

DISPLAY Sat Launched Display

     Satellite [Sat Name] launched on [Vehicle Name] from [Pad Name]
       on Schedule Day [Current Schedule Day (6,0)].


       Press Function Key 2 CONT to resume the scheduling process

DISPLAY Sat Launched Printer Display

     Satellite [Sat Name] launched on [Vehicle Name] from [Pad Name]
       on Schedule Day [Current Schedule Day (6,0)].

!-------------------------------------------------------------------------------
!  Printer Unique Text
!-------------------------------------------------------------------------------
!
!  Define format for printer user id banner
!
DISPLAY User ID
*********************************************************************************

Product created by [User Name] on [Date]

[Comment]

DISPLAY Banner
*********************************************************************************
|                                                                               |
|  *************************************************************************    |
|  |                                                                       |    |
|  |                                                                       |    |
|  |                                L R S                                  |    |
|  |                                                                       |    |
|  |                    Launch   Resource  Scheduling                      |    |
|  |                                                                       |    |
|  |                         A  Planning  Tool                             |    |
|  |                                                                       |    |
|  |                                by                                     |    |
|  |                                                                       |    |
|  |                           Major Fred H. Koch                          |    |
|  |                                                                       |    |
|  |                                                                       |    |
|  *************************************************************************    |
|                                                                               |
*********************************************************************************
```

```
!
!  Define printer program assumptions format
!
DISPLAY Printer Program Assumptions

*******************************************************************************
                          Program Assumptions

    This program uses all deterministic computations.  This means if a
    vehicle launches it always successfully places its payload into orbit.
    If a satellite takes two weeks checkout time it will always take exactly
    two weeks to checkout.  This reasoning was used to simplify the program
    operation.

    All specific parameters for vehicles, launch pads, and satellites are
    entered by the user.  This information is used by the program to provide
    a possible matching of launch resources to launch requirements.

    The schedule provided is not necessarily the optimum schedule.  It is a
    feasible schedule.  One which meets all the constraints established by
    the user entered information and the scheduling rules of the program.


!  Vehicle Record Format for Printer
!
DISPLAY Vehicle Record Format

     Information for Vehicle Record Number [Veh Record Number (3,0)]

          Vehicle Name ..............  [Vehicle Name]
          Vehicle Type ..............  [Vehicle Type]
          Boost Capability ..........  [Boost Capability (10,0)]
          Payload Size ..............  [Payload Size (5,0)]
          Orbit Altitude Capability .  [Orbit Altitude Cap (7,0)]
          Launch Facility ...........  [Veh Launch Facility]
          Max Mission Duration ......  [Mission Duration (3,0)]
          Vehicle Turn Time .........  [Veh Turn Time (3,0)]
          When First Available ......  [Veh First Available (6,0)]
          When Next Available .......  [Veh Next Available (6,0)]

DISPLAY Pad Record Format

     Information for Pad Record Number [Pad Record Number (3,0)]

          Pad Name ..............  [Pad Name]
          Pad Type ..............  [Pad Type]
          Launch Facility .....  [Pad Launch Facility]
          Pad Turn Time .......  [Pad Turn Time (3,0)]
          Pad First Available .  [Pad First Available (6,0)]
          Pad Next Available ..  [Pad Next Available (6,0)]

DISPLAY Satelite Record Format
```

```
     Information for Satellite Record Number [Sat Record Number (3,0)]

          Satellite Name ................. [Sat Name]
          Constellation Part of ......... [Constel]
          Weight ........................ [Sat Weight (10,0)]
          Size .......................... [Sat Size (5,0)]
          Orbit Altitude ................ [Sat Orbit Alt (7,0)]
          Launch Vehicle 1 .............. [Lch Veh 1]
          Launch Vehicle 2 .............. [Lch Veh 2]
          Launch Priority ............... [Sat Lch Priority (2,0)]
          Launch Restrictions ........... [Sat Restrictn]
          Classified .................... [Sat Classified]
          Launch with Classified Payload . [Sat Lch With Classified]
          Desired Launch Day ............ [Sat Desire Lch (6,0)]
          Day Available for Launch ...... [Sat Available (6,0)]
          Not Later Than Launch Day ..... [Sat NLT Lch (6,0)]
          Type Orbit Required ........... [Sat Orbit Type]
          Orbit Inclination ............. [Sat Inclinatn (3,0)]
          Launched ...................... [Sat Launched]

!TEXT text
!
!-------------------------------------------------------------------------
!  Text for Main Menu
!-------------------------------------------------------------------------
!
!  Define the question to be used for getting value of Program Option
!
TEXT Program Option

Choose the action you wish to perform:

!-------------------------------------------------------------------------
!
!  Define the question to be used for getting the value of Decision
!
TEXT Decision

     Schedule Resources Menu

Which Action do you want to perform ?

!-------------------------------------------------------------------------
!  Text for Main Menu Sub Rules
!-------------------------------------------------------------------------
!
!  Define the question to be used for getting value of User Name
!
TEXT User Name

Enter your name:
```

```
! Define the question to be used for getting value of Date
!
TEXT Date

Enter Today's Date

! Define the question to be used for getting value of Comment
!
TEXT Comment

Enter any line of comment desired:

! Define the question to be used for getting value of Print Status
!
TEXT Print Status

Do you want the schedule printed on the printer ?

!    Text for Schedule Information Questions
!
TEXT Start Schedule Day

Enter the day number the schedule should start at.

      This is usually 1 but may be any number less than the end number.

TEXT Ending Schedule Day

Enter the day number the schedule should end with.

      This can be any number greater than the start day number, or 0 if a
      schedule to launch all satellites is desired, regardless how long it
      takes.

! Text for View Choice question
!
TEXT View Choice

What information do you want to view ?


! Text for Choose Output for view information
!
TEXT Info Output Status

Where do you want the information to be displayed ?

TEXT Reset File

Which files do you want to be Reset ?
```

```
!
!  Define the question to be used for getting the value of Decision
!
TEXT Decision

     Schedule Resources Menu

Which Action do you want to perform ?

!EXPAND text
!
!-------------------------------------------------------------------------
!  Expand text for Main Menu
!-------------------------------------------------------------------------
!
!  Define the text to be presented when EXPL is pressed for the Program
!    Option Menu
!
EXPAND Program Option
*****************************************************************************
                          Menu Explanation

    View Information allows viewing of program assumptions and Vehicle, Pad,
        or Satellite db file data.
    Generate Schedule will request the schedule information and start the
        resource/requirement matching process.
    Reset DB Files will return the db files to their original condition
        before a schedule run was accomplished.
    Enter New Parameters will provide a brief explanation of how to enter
        new parameters to be used by the program.
    Load Saved Parameters explains which data files are used by the LRS
        program and how to choose different db files.
    Change Parameters explains how to change the information used by the
        LRS program.
    Save Parameters explains how the LRS program updates the resource and
        requirement files during program execution.
    Save Resource Schedule explains how to save the schedule to a disk
        file for later use.

                Press Any Key to return to the Main Menu

!=========================================================================
!  The ending statement
!=========================================================================
!
END
```

The following DBPAS code retrieves data base record information and returns it to the knowledge base program.

Get Satellite Data Code

```
!  Get Data Base File Information in Satellite file
!  Version 0.06  2 Dec 86   by Fred H. Koch
!
!
!  This program is designed to retreive the information in the
!  data base file SATELITE.DBF used by LRS.


program getsat (receive Rec_num : integer;
                return last_record          : char;
                       name                 : string (30);
                       constellation        : string (30);
                       weight               : real;
                       sat_size             : real;
                       orbit_altitude       : real;
                       launch_vehicle_1     : string (30);
                       launch_vehicle_2     : string (30);
                       launch_priority      : real;
                       launch_restrict      : string (50);
                       classified           : boolean;
                       lch_with_class       : boolean;
                       desired_lch_day      : real;
                       avail_lch_day        : real;
                       nlt_launch_day       : real;
                       orbit_type           : string (10);
                       orbit_inclinatn      : real;
                       launched             : string(1);
                       Last_rec_num         : integer);

var
     dummy : real;

!  Define the record format to match the data base file structure.
!
!  Note:
!    For dbase fields of type "C" use DBPAS string of same length
!    For dbase fields of type "N" use DBPAS real (no length)
!    For dbase fields of type "L" use DBPAS boolean
!    For dbase fields of type "D" use DBPAS string of length 8
!    For dbase fields of type "M" use DBPAS string of length 10
```

```
sat : record
     name                    : string (30);
     constellation           : string (30);
     weight                  : real;
     sat_size                : real;
     orbit_altitude          : real;
     launch_vehicle_1        : string (30);
     launch_vehicle_2        : string (30);
     launch_priority         : real;
     launch_restrict         : string (50);
     classified              : boolean;
     lch_with_class          : boolean;
     desired_lch_day         : real;
     avail_lch_day           : real;
     nlt_launch_day          : real;
     orbit_type              : string (10);
     orbit_inclinatn         : real;
     launched                : string(1);
end;

!  Establish steps to get records from data base

begin

!  Open the file "SATELITE.DBF" and call it sat in the program.

   open (sat, 'SATELITE');
   Last_rec_num := SIZE(sat);

   if Rec_num <= Last_rec_num
     then begin
        goto (Rec_num, sat);
        last_record             := 'n';
        name                    := sat.name;
        constellation           := sat.constellation;
        weight                  := sat.weight;
        sat_size                := sat.sat_size;
        orbit_altitude          := sat.orbit_altitude;
        launch_vehicle_1        := sat.launch_vehicle_1;
        launch_vehicle_2        := sat.launch_vehicle_2;
        launch_priority         := sat.launch_priority;
        launch_restrict         := sat.launch_restrict;
        classified              := sat.classified;
        lch_with_class          := sat.lch_with_class;
        desired_lch_day         := sat.desired_lch_day;
        avail_lch_day           := sat.avail_lch_day;
        nlt_launch_day          := sat.nlt_launch_day;
        orbit_type              := sat.orbit_type;
        orbit_inclinatn         := sat.orbit_inclinatn;
        launched                := sat.launched;
      end
     else begin
```

```
            last_record             := 'e';
            name                    := 'no name';
            constellation           := 'no constellation';
            weight                  := 0.0;
            sat_size                := 0.0;
            orbit_altitude          := 0.0;
            launch_vehicle_1        := 'no vehicle';
            launch_vehicle_2        := 'no vehicle';
            launch_priority         := 0.0;
            launch_restrict         := 'no restriction';
            classified              := false;
            lch_with_class          := false;
            desired_lch_day         := 0.0;
            avail_lch_day           := 0.0;
            nlt_launch_day          := 0.0;
            orbit_type              := 'no orbit';
            orbit_inclinatn         := 0.0;
            launched                := 'n ';
          end
        end;

        close (sat);

    end;
```

Get Vehicle Data Code

```
    ! Get Data Base File Information in Vehicle file
    ! Version 0.05 2 Dec 86   by Fred H. Koch
    !
    !
    ! This program is designed to retreive the information in the
    ! data base file VEHICLE.DBF used by LRS.


    program getveh (receive Rec_num         : integer;
                    return last_record      : char;
                           name             : string (30);
                           vehicle_type     : string (30);
                           boost_cap        : real;
                           payload_size     : real;
                           orbit_cap        : real;
                           launch_facility  : string(30);
                           mission_dur      : real;
                           turn_time        : real;
                           first_avail      : real;
                           next_avail       : real);

    var
        Last_rec_num : integer;
```

```
!  Define the record format to match the data base file structure.
!
!  Note:
!    For dbase fields of type "C" use DBPAS string of same length
!    For dbase fields of type "N" use DBPAS real (no length)
!    For dbase fields of type "L" use DBPAS boolean
!    For dbase fields of type "D" use DBPAS string of length 8
!    For dbase fields of type "M" use DBPAS string of length 10

vehicles : record
     name           : string (30);
     vehicle_type   : string (30);
     boost_cap      : real;
     payload_size   : real;
     orbit_cap      : real;
     launch_facility: string (30);
     mission_dur    : real;
     turn_time      : real;
     first_avail    : real;
     next_avail     : real;
end;

!  Establish steps to get records from data base

begin

!  Open the file "VEHICLE.DBF" and call it vehicles in the program.

   open (vehicles, 'VEHICLE');
   Last_rec_num := SIZE(vehicles);

   if Rec_num <= Last_rec_num
     then begin
        goto (Rec_num, vehicles);
        last_record     := 'n';
        name            := vehicles.name;
        vehicle_type    := vehicles.vehicle_type;
        boost_cap       := vehicles.boost_cap;
        payload_size    := vehicles.payload_size;
        orbit_cap       := vehicles.orbit_cap;
        launch_facility := vehicles.launch_facility;
        mission_dur     := vehicles.mission_dur;
        turn_time       := vehicles.turn_time;
        first_avail     := vehicles.first_avail;
        next_avail      := vehicles.next_avail;
      end
     else begin
        last_record     := 'e';
        name            := 'no vehicle';
        vehicle_type    := 'no type';
        boost_cap       := 0.0;
        payload_size    := 0.0;
```

```
        orbit_cap       := 0.0;
        launch_facility := 'no facility';
        mission_dur     := 0.0;
        turn_time       := 0.0;
        first_avail     := 0.0;
        next_avail      := 0.0;
      end;
      end;


    close (vehicles);

end;
```

Get Pad Data Code

```
! Get Data Base File Information in Pad file
! Version 0.05  2 Dec 86   by Fred H. Koch
!
!
! This program is designed to retreive the information in the
! data base file PAD.DBF used by LRS.


program getpad (receive Rec_num : integer;
                return last_record    : char;
                       name           : string (30);
                       vehicle_type   : string (30);
                       launch_facility: string (30);
                       turn_time      : real;
                       first_avail    : real;
                       next_avail     : real);


var
    Last_rec_num : integer;

! Define the record format to match the data base file structure.
!
! Note:
!   For dbase fields of type "C" use DBPAS string of same length
!   For dbase fields of type "N" use DBPAS real (no length)
!   For dbase fields of type "L" use DBPAS boolean
!   For dbase fields of type "D" use DBPAS string of length 8
!   For dbase fields of type "M" use DBPAS string of length 10

pad : record
    name           : string (30);
    vehicle_type   : string (30);
    launch_facility: string (30);
```

```
      turn_time      : real;
      first_avail    : real;
      next_avail     : real;
end;

!  Establish steps to get records

begin

!  Open the file "PAD.DBF" and call it pad in the program.

   open (pad, 'PAD');
   Last_rec_num := SIZE(pad);

   if Rec_num <= Last_rec_num
     then begin
        goto (Rec_num, pad);
        last_record     := 'n';
        name            := pad.name;
        vehicle_type    := pad.vehicle_type;
        launch_facility := pad.launch_facility;
        turn_time       := pad.turn_time;
        first_avail     := pad.first_avail;
        next_avail      := pad.next_avail;
       end
      else begin
        last_record     := 'e';
        name            := 'no pad';
        vehicle_type    := 'no type';
        launch_facility := 'no facility';
        turn_time       := 0.0;
        first_avail     := 0.0;
        next_avail      := 0.0;
       end;
    end;

   close (pad);

end;
```

The following DBPAS code presents the data base information on the computer screen one record at a time for the view data base file information options of View Information.

View Satellite Data Code

```
!  View Data Base File Information in Satellite file
!  Version 0.04  17 Nov 86   by Fred H. Koch
!
!
!  This program is designed to retreive the information in the
!  data base file SATELITE.DBF used by LRS.


program viewsat;

var
     record_number : integer;
        Dummy_char : char;

!  Define the record format to match the data base file structure.
!
!  Note:
!    For dbase fields of type "C" use DBPAS string of same length
!    For dbase fields of type "N" use DBPAS real (no length)
!    For dbase fields of type "L" use DBPAS boolean
!    For dbase fields of type "D" use DBPAS string of length 8
!    For dbase fields of type "M" use DBPAS string of length 10

sat : record
     name                    : string (30);
     constellation           : string (30);
     weight                  : real;
     Sat_size                : real;
     orbit_altitude          : real;
     launch_vehicle_1        : string (30);
     launch_vehicle_2        : string (30);
     launch_priority         : real;
     launch_restrict         : string (50);
     classified              : boolean;
     lch_with_class          : boolean;
     desired_lch_day         : real;
     avail_lch_day           : real;
     nlt_launch_day          : real;
     orbit_type              : string (10);
     orbit_inclinatn         : real;
     launched                : string (1);
end;

!  Establish steps to display records to screen

begin

!  Open the file "SATELITE.DBF" and call it sat in the program.

   open (sat, 'SATELITE');
```

```
        while not eof (sat) do
         begin
           clear;
           writeln;
           writeln ('         Satellite Data Base File');
           writeln;
           writeln ('            Record Number: ', RECNO (sat));
           writeln ('           Satellite Name: ', sat.name);
           writeln ('            Constellation: ', sat.constellation);
           writeln ('                   Weight: ', sat.weight);
           writeln ('                     Size: ', sat.sat_size);
           writeln ('           Orbit Altitude: ', sat.orbit_altitude);
           writeln ('         Launch Vehicle 1: ', sat.launch_vehicle_1);
           writeln ('         Launch Vehicle 2: ', sat.launch_vehicle_2);
           writeln ('          Launch Priority: ', sat.launch_priority);
           writeln ('      Launch Restrictions: ', sat.launch_restrict);
           writeln ('               Classified: ', sat.classified);
           writeln ('   Launch with Classified: ', sat.lch_with_class);
           writeln ('       Desired Launch Day: ', sat.desired_lch_day);
           writeln ('     Available for Launch: ', sat.avail_lch_day);
           writeln ('Not Later Than Launch Day: ', sat.nlt_launch_day);
           writeln ('               Orbit Type: ', sat.orbit_type);
           writeln ('        Orbit Inclination: ', sat.orbit_inclinatn);
           writeln ('                 Launched: ', sat.launched);
           writeln;
           writeln ('Press any key for next satellite');
           Dummy_char := getchar;
           next (sat);
         end;

     close (sat);

end;
```

View Vehicle Data Code

```
! View Data Base File Information in Vehicle file
! Version 0.04  17 Nov 86   by Fred H. Koch
!
!
! This program is designed to retreive and display the information in the
! data base files used by LRS.


program viewfile;

var
     record_number : integer;
        Dummy_char : char;
```

```
!  Define the record format to match the data base file structure.
!
!  Note:
!    For dbase fields of type "C" use DBPAS string of same length
!    For dbase fields of type "N" use DBPAS real (no length)
!    For dbase fields of type "L" use DBPAS boolean
!    For dbase fields of type "D" use DBPAS string of length 8
!    For dbase fields of type "M" use DBPAS string of length 10

vehicles : record
     name           : string (30);
     vehicle_type   : string (30);
     boost_cap      : real;
     payload_size   : real;
     orbit_cap      : real;
     launch_facility: string (30);
     mission_dur    : real;
     turn_time      : real;
     first_avail    : real;
     next_avail     : real;
end;

!  Establish steps to display records to screen

begin

!  Open the file "VEHICLE.DBF" and call it vehicles in the program.

   open (vehicles, 'VEHICLE');


     while not eof (vehicles) do
      begin
       clear;
       writeln;
       writeln ('    Vehicle Data Base File');
       writeln;
       writeln ('   Record Number: ',RECNO(vehicles));
       writeln ('    Vehicle Name: ', vehicles.name);
       writeln ('            Type: ', vehicles.vehicle_type);
       writeln ('Boost Capability: ', vehicles.boost_cap);
       writeln ('    Payload Size: ', vehicles.payload_size);
       writeln ('Orbit Capability: ', vehicles.orbit_cap);
       writeln (' Launch facility: ', vehicles.launch_facility);
       writeln ('Mission Duration: ', vehicles.mission_dur);
       writeln ('       Turn Time: ', vehicles.turn_time);
       writeln (' First Available: ', vehicles.first_avail);
       writeln ('  Next Available: ', vehicles.next_avail);
       writeln;
       writeln ('Press any key for next vehicle');
       Dummy_char := getchar;
       next (vehicles);
```

```
      end;

     close (vehicles);

end;
```

View Pad Data Code

```
!  View Data Base File Information in Pad file
!  Version 0.04  17 Nov 86   by Fred H. Koch
!
!
!  This program is designed to retreive and display the information in the
!  data base file PAD.DBF used by LRS.


program viewpad;

var
     record_number : integer;
        Dummy_char : char;

!  Define the record format to match the data base file structure.
!
!  Note:
!    For dbase fields of type "C" use DBPAS string of same length
!    For dbase fields of type "N" use DBPAS real (no length)
!    For dbase fields of type "L" use DBPAS boolean
!    For dbase fields of type "D" use DBPAS string of length 8
!    For dbase fields of type "M" use DBPAS string of length 10

pad : record
     name           : string (30);
     vehicle_type   : string (30);
     launch_facility: string (30);
     turn_time      : real;
     first_avail    : real;
     next_avail     : real;
end;

!  Establish steps to display records to screen

begin

!  Open the file "PAD.DBF" and call it pad in the program.

   open (pad, 'PAD');

        while not eof (pad) do
             begin
               clear;
```

```
                writeln;
                writeln (' Launch Pad Data Base File');
                writeln;
                writeln ('   Record Number: ', RECNO (pad));
                writeln ('        Pad Name: ', pad.name);
                writeln ('    Vehicle Type: ', pad.vehicle_type);
                writeln (' Launch facility: ', pad.launch_facility);
                writeln ('       Turn Time: ', pad.turn_time);
                writeln (' First Available: ',pad.first_avail);
                writeln ('  Next Available: ', pad.next_avail);
                writeln;
                writeln ('Press any key for next pad');
                Dummy_char := getchar;
                next (pad);
              end;

    close (pad);

end;
```

The following code is used to eliminate the effect of past LRS scheduling session on the data base files. This code is called during the Reset DB Files option.

Reset Satellite File Code

```
!  Reset Satellite Data Base file
!  Version 0.02  5 Nov 86  Major Fred H. Koch
!
!  This program is designed to reset the parameters in the db which
!   are changed during LRS execution.
!
!  launched is the only parameter currently changed.  This is reset
!   to n during the reset process indicating not launched.

program resetsat;

!  Define the record format to match the data base file structure.
!
!  Note:
!    For dbase fields of type "C" use DBPAS string of same length
!    For dbase fields of type "N" use DBPAS real (no length)
!    For dbase fields of type "L" use DBPAS boolean
!    For dbase fields of type "D" use DBPAS string of length 8
!    For dbase fields of type "M" use DBPAS string of length 10

var
   dummy : real;
```

```
sat : record
     name                    : string (30);
     constellation           : string (30);
     weight                  : real;
     sat_size                : real;
     orbit_altitude          : real;
     launch_vehicle_1        : string (30);
     launch_vehicle_2        : string (30);
     launch_priority         : real;
     launch_restrict         : string (50);
     classified              : boolean;
     launch_with_classified: boolean;
     desired_launch_day      : real;
     available_launch_day    : real;
     nlt_launch_day          : real;
     orbit_type              : string (10);
     orbit_inclination       : real;
     launched                : string (1);
end;

!  Send message to the screen that the reset process is working.

begin
   clear;
   writeln ('  Resetting the SATELITE data base file');

!  Establish steps to get records from data base

!  Open the file "SATELITE.DBF" and call it sat in the program.

   open (sat, 'SATELITE');

   while not eof (sat) do
     begin
        writeln ('  Resetting record number ', RECNO(sat));
        sat.launched := 'n ';
        REPLACE(sat);
        next (sat);
     end;

   close (sat);

end;
```

Reset Vehicle File Code

```
!  Reset Vehicles Data Base file
!  Version 0.02  5 Nov 86  Major Fred H. Koch
!
!  This program is designed to reset the parameters in the db which
```

```
!   are changed during LRS execution.
!
!  next_available is the only parameter currently changed.  This is
!   reset to the first_available date during the reset process.

program resetveh;

var
   dummy : real;

!  Define the record format to match the data base file structure.
!
!  Note:
!    For dbase fields of type "C" use DBPAS string of same length
!    For dbase fields of type "N" use DBPAS real (no length)
!    For dbase fields of type "L" use DBPAS boolean
!    For dbase fields of type "D" use DBPAS string of length 8
!    For dbase fields of type "M" use DBPAS string of length 10

vehicles : record
     name           : string (30);
     vehicle_type   : string (30);
     boost_cap      : real;
     payload_size   : real;
     orbit_cap      : real;
     launch_facility: string (30);
     mission_dur    : real;
     turn_time      : real;
     first_avail    : real;
     next_avail     : real;
end;

!  Send message to the screen that the reset process is working.

begin
   clear;
   writeln ('  Resetting the VEHICLE data base file');

!  Establish steps to get records from data base

!  Open the file "VEHICLE.DBF" and call it vehicles in the program.

   open (vehicles, 'VEHICLE');

   while not eof (vehicles) do
     begin
        writeln ('  Resetting record number ', RECNO(vehicles));
        vehicles.next_avail := vehicles.first_avail;
        REPLACE(vehicles);
        next (vehicles);
     end;
```

```
   close (vehicles);

end;
```

Reset Pad File Code

```
!  Reset Pad Data Base file
!  Version 0.02  5 Nov 86  Major Fred H. Koch
!
!  This program is designed to reset the parameters in the db which
!   are changed during LRS execution.
!
!  next_available is the only parameter currently changed.  This is
!   reset to the first_available date during the reset process.

program resetpad ;

var
  dummy : real;

!  Define the record format to match the data base file structure.
!
!  Note:
!    For dbase fields of type "C" use DBPAS string of same length
!    For dbase fields of type "N" use DBPAS real (no length)
!    For dbase fields of type "L" use DBPAS boolean
!    For dbase fields of type "D" use DBPAS string of length 8
!    For dbase fields of type "M" use DBPAS string of length 10

pad : record
     name           : string (30);
     vehicle_type   : string (30);
     launch_facility: string (30);
     turn_time      : real;
     first_avail    : real;
     next_avail     : real;
end;

!  Send message to the screen that the reset process is working.

begin
   clear;
   writeln ('  Resetting the PAD data base file');

!  Establish steps to get records from data base

!  Open the file "PAD.DBF" and call it pad in the program.

   open (pad, 'PAD');

   while not eof (pad) do
```

```
      begin
         writeln ('  Resetting record number ', RECNO (pad));
         pad.next_avail := pad.first_avail;
         REPLACE(pad);
         next (pad);
      end;

   close (pad);

end;
```

The following code is used to mark the data base files indicating a requirement no longer exists or that a resource has been used.

Mark Satellite File Code

```
! Mark Satellite Data Base file
!  Version 0.01  12 Nov 86  Major Fred H. Koch
!
!  This program is designed to mark the parameters in the db which
!   are changed during LRS execution.
!
!  launched is the only parameter currently changed.  This is marked
!   to 1 for launch and m for missed launch during the scheduling process.

program marksat (receive Mark_char : string (1);
                         Rec_num : integer);

!  Define the record format to match the data base file structure.
!
!  Note:
!    For dbase fields of type "C" use DBPAS string of same length
!    For dbase fields of type "N" use DBPAS real (no length)
!    For dbase fields of type "L" use DBPAS boolean
!    For dbase fields of type "D" use DBPAS string of length 8
!    For dbase fields of type "M" use DBPAS string of length 10

var
   dummy : real;

sat : record
      name                 : string (30);
      constellation        : string (30);
      weight               : real;
      sat_size             : real;
      orbit_altitude       : real;
      launch_vehicle_1     : string (30);
      launch_vehicle_2     : string (30);
      launch_priority      : real;
```

```
      launch_restrict       : string (50);
      classified            : boolean;
      launch_with_classified: boolean;
      desired_launch_day    : real;
      available_launch_day  : real;
      nlt_launch_day        : real;
      orbit_type            : string (10);
      orbit_inclination     : real;
      launched              : string (1);
end;

begin

! Establish steps to get records from data base

! Open the file "SATELITE.DBF" and call it sat in the program.

   open (sat, 'SATELITE');


   begin
     goto (Rec_num, sat);
     sat.launched := Mark_char;
     REPLACE(sat);
   end;

  close (sat);

end;
```

Mark Vehicle File Code

```
! Mark Vehicles Data Base file
! Version 0.01  12 Nov 86  Major Fred H. Koch
!
! This program is designed to mark the parameters in the db which
!  are changed during LRS execution.
!
! next_available is the only parameter currently changed.  This is
!  set to the launch day + turn time.

program markveh (receive Lch_day : real;
                         Rec_num : integer);

var
   dummy : real;

! Define the record format to match the data base file structure.
!
! Note:
!   For dbase fields of type "C" use DBPAS string of same length
```

```
!    For dbase fields of type "N" use DBPAS real (no length)
!    For dbase fields of type "L" use DBPAS boolean
!    For dbase fields of type "D" use DBPAS string of length 8
!    For dbase fields of type "M" use DBPAS string of length 10

vehicles : record
     name           : string (30);
     vehicle_type   : string (30);
     boost_cap      : real;
     payload_size   : real;
     orbit_cap      : real;
     launch_facility: string (30);
     mission_dur    : real;
     turn_time      : real;
     first_avail    : real;
     next_avail     : real;
end;

begin

!  Establish steps to get records from data base

!  Open the file "VEHICLE.DBF" and call it vehicles in the program.

   open (vehicles, 'VEHICLE');

     begin
        goto (Rec_num, vehicles);
        vehicles.next_avail := Lch_day + vehicles.turn_time;
        REPLACE(vehicles);
     end;

   close (vehicles);

end;
```

Mark Pad File Code

```
!  Mark Pad Data Base file
!  Version 0.01  12 Nov 86  Major Fred H. Koch
!
!  This program is designed to mark the parameters in the db which
!   are changed during LRS execution.
!
!  next_available is the only parameter currently changed.  This is
!   set to the launch date plus turn time.

program markpad (receive Lch_day : real;
                         Rec_num : integer);

var
```

```
  dummy : real;

!  Define the record format to match the data base file structure.
!
!  Note:
!    For dbase fields of type "C" use DBPAS string of same length
!    For dbase fields of type "N" use DBPAS real (no length)
!    For dbase fields of type "L" use DBPAS boolean
!    For dbase fields of type "D" use DBPAS string of length 8
!    For dbase fields of type "M" use DBPAS string of length 10

pad : record
     name           : string (30);
     vehicle_type   : string (30);
     launch_facility: string (30);
     turn_time      : real;
     first_avail    : real;
     next_avail     : real;
end;

begin

!  Establish steps to get records from data base

!  Open the file "PAD.DBF" and call it pad in the program.

   open (pad, 'PAD');

     begin
        goto (Rec_num, pad);
        pad.next_avail := Lch_day + pad.turn_time;
        REPLACE(pad);
     end;

   close (pad);

end;
```

The following code is used to display program execution information to the computer screen to advise the user what action the program is taking.

Screen Message Code

```
!  Print Message to the Screen Program
!  Version 0.02  18 Nov 86  by Fred H. Koch
!
!
!  This program is designed to print a message to the screen during I2+
!   program execution without stopping the program execution as a Display
```

```
!   command does.

program scrnmsg (receive Message : string (80));

var
  dummy : real;

begin
   clear;
   writeln;
   writeln;
   writeln;
   writeln (Message);
end;
end;
```

Appendix C: LRS User Manual

LRS version 1.0 is described in the following manual. LRS was developed using Insight 2+ version 1.2. If a later version of Insight 2+ is to be used, refer to the current Insight 2+ Reference Manual (20) for any compatibility problems which may occur.

The information presented below will provide the basic information required to use LRS. The Insight 2+ Reference Manual is required to set-up the computer for either floppy or hard disk operation. Once the computer is configured to run Insight 2+, the following information may be used to run LRS.

Required Equipment

LRS was developed on a Gemini enhanced Zenith Z-100 microcomputer with 640K RAM, one 720K floppy disk drive, and a hard disk drive. The program was also operated on a Zenith Z-150 with 640K RAM and two 720K floppy disk drives.

A hard disk drive is recommended due to the large program size and frequent disk access required during program operation. If a floppy disk is used there is not enough disk space to place the Insight 2+ help file on the disk with the LRS programs. LRS will operate properly using the floppy disk drive configuration, however the Insight 2+ help function will not be available and will cause the program to stop if the HELP function key is pressed.

Required Files

In addition to the required Insight 2+ files, the following files must be available in the default directory, of the default disk drive, for LRS to operate properly. See the Insight 2+ Reference Manual for instructions on making default disk drive selections.

LRS	.KNB	LRS compiled program code
VEHICLE	.DBF	Vehicle resource data base file
PAD	.DBF	Launch Pad resource data base file
SATELITE	.DBF	Satellite requirements data base file
RESETPAD	.PCO	Reset launch pad DBPAS compiled code
RESETVEH	.PCO	Reset vehicle DBPAS compiled code
RESETSAT	.PCO	Reset satellite DBPAS compiled code
VIEWPAD	.PCO	View launch pad DBPAS compiled code
VIEWVEH	.PCO	View vehicle DBPAS compiled code
VIEWSAT	.PCO	View satellite DBPAS compiled code
GETPAD	.PCO	Get launch pad data DBPAS compiled code
GETVEH	.PCO	Get vehicle data DBPAS compiled code
GETSAT	.PCO	Get satellite data DBPAS compiled code
MARKPAD	.PCO	Mark launch pad data DBPAS compiled code
MARKVEH	.PCO	Mark vehicle data DBPAS compiled code
MARKSAT	.PCO	Mark satellite data DBPAS compiled code
SCRNMSG	.PCO	Screen msg display DBPAS compiled code
SCHEDULE	.DAT	Schedule information data file
DBPAS	.COM	Insight 2+ DBPAS execution file

The source code files printed in Appendix B are available on the LRS program disk. These files are not required to run LRS but are provided should program changes be desired.

LRS	.PRL	LRS knowledge base program code
RESETPAD	.PAS	Reset launch pad DBPAS source code
RESETVEH	.PAS	Reset vehicle DBPAS source code
RESETSAT	.PAS	Reset satellite DBPAS source code
VIEWPAD	.PAS	View launch pad DBPAS source code
VIEWVEH	.PAS	View vehicle DBPAS source code
VIEWSAT	.PAS	View satellite DBPAS source code
GETPAD	.PAS	Get launch pad data DBPAS source code
GETVEH	.PAS	Get vehicle data DBPAS source code
GETSAT	.PAS	Get satellite data DBPAS source code
MARKPAD	.PAS	Mark launch pad data DBPAS source code
MARKVEH	.PAS	Mark vehicle data DBPAS source code
MARKSAT	.PAS	Mark satellite data DBPAS source code
SCRNMSG	.PAS	Screen msg display DBPAS source code

Program Operation

See the Insight 2+ Reference Manual for instructions on loading and running Insight 2+. From the Insight 2+ main menu select Run Knowledge Base. Select LRS using the space bar or arrow keys to position the pointer in front of LRS and press RETURN/ENTER.

The LRS introduction screen will be displayed as soon as LRS has completed loading. The top line on the

introduction screen will display the LRS program name and the current version number. You may view a brief description of LRS by pressing function key 1 PAGE. Pressing function key 3 STRT will start LRS program execution.

The Main Menu. The Main Menu is displayed after pressing the STRT function key. From this menu all LRS functions may be selected. Each function is described below.

View Information. The view information selection presents a sub-menu of the information available. Assumptions provides the program assumptions. This information is also printed on all scheduling output generated by LRS.

Choosing All Data Base Files will display the data base information currently being used by LRS. This includes the information in the VEHICLE.DBF file, PAD.DBF file, and SATELITE.DBF file. This information may be viewed on the computer screen, or sent to the printer.

The information in an individual data base file may be viewed on the computer screen or sent to the printer by selecting to view the information in that file.

If no information is desired to be viewed, choosing Return to Main Menu will return to the LRS Main Menu without viewing any of the available information.

Generate Schedule. Generate Schedule allows starting the requirement/resource matching process of LRS. The first time Generate Schedule is selected a brief explanation of the matching process of LRS is displayed. Pressing function key 2 CONT will continue program execution, displaying a sub-menu. From the sub-menu choosing Schedule Resources will begin the matching process. Return to Main Menu will return to the LRS Main Menu without starting the matching process.

The first action after selecting Schedule Resources is to enter your name, the date, and a line of comment. This information is printed on the schedule printout and sent to the SCHEDULE.DAT file to identify the schedule output.

Selecting where the schedule output is to be sent is the next action. The choices are the computer screen or the printer. Using the screen requires pressing function key 2 CONT after each piece of schedule information is presented. Choosing printer output will send the information to the printer requiring no keyboard input during the matching process. In either case the schedule information is also sent to the SCHEDULE.DAT file. This file may be viewed, printed, or saved for later use, after exiting LRS. IMPORTANT !! The information stored in SCHEDULE.DAT is over written at the beginning of each LRS session. See Save Resource Schedule for instructions on how to save this information.

The schedule period information is entered next. This is the start and ending schedule day to be used. The start day may be any number between 1 and 999999. The ending day must be greater than or equal to the start day, or 0 to schedule all resources regardless how long it may take.

Once the schedule information is entered LRS will attempt to match launch resources to all launch requirements within the schedule period entered. As the matching progresses, the computer screen will display messages stating what action is currently being performed. As requirements are met or determined to be unable to be met, the information will be sent to the device designated when entering schedule information.

The matching process will end when all launch requirements within the scheduling period have been attempted to be matched to a launch resource.

Reset DB Files. The Reset DB Files option resets the data base file information to the initial values. The choices are to reset all data base files, reset any particular file, or return to the main menu.

Resetting the satellite file will set the launch status of all satellites as available for launch. Resetting the vehicle data base file will mark all vehicles as next available on their first available day. The launch pad file will have all pads marked as next available on their first available day when reset.

The reset data base option should be used before choosing generate schedule whenever a new schedule is desired. This will eliminate the changes from a previous schedule generation process. If the files are not reset the changes from the last schedule generated will remain. This allows continuing a schedule past the previous ending day, during the next schedule generation process.

Enter New Parameters. The Enter New Parameters choice is not currently implemented in LRS. Adding new information to the data base files must be accomplished using the Insight 2+ data base editor or the dBase II or dBase III programs.

To use the data base editor, exit LRS to return to the Insight 2+ main menu. Choose the Edit a Data Base selection. A list of the data base files available for editing is presented. Use the space bar or cursor arrow keys to place the pointer in front of the desired data base file name, press RETURN/ENTER.

The file format screen will be presented showing the fields in the file and the type and size of each field. Function key 4 ADD is used to add new information to the file. After function key 4 is pressed, the first field in the new record will be presented for value entry. Each field will be presented in turn until all fields have been presented. After the last field is entered the field format screen will again be shown. Repeat the process for each new

record which must be added.

An alternative method is to use the dBase II or dBase III programs to add information to the files. Since the files are in dBase II format, the additional features of the dBase programs may be used to add information to the LRS files.

Load Saved Parameters. The Load Saved Parameters option is not implemented in LRS. This option presents an explanation screen stating LRS uses the same files each time it is executed. The data base files are modified as the scheduling progresses. In this way any requirement or resource changes made are carried to the next session with no user action required. The Reset DB Files option is used to restore the files to their condition before a schedule was attempted.

It is possible to use several different requirement and resource files for planning comparisons. This requires renaming the LRS data files before LRS is executed. The data files may have any name desired for storage purposes, however only the files named SATELITE.DBF, VEHICLE.DBF, and PAD.DBF will be used during LRS execution. By creating several different requirement and resource files, and renaming the desired files before LRS execution, the proper files will be used for the scheduling session. See How to Create Data Base Files for an explanation of how to create new data base files.

Change Parameters. The Change Parameters option is not implemented in LRS. The data base editor in Insight 2+ or the dBase II or dBase III programs must be used to change the data base files.

To use the Insight 2+ data base editor, exit LRS to return to the Insight 2+ main menu. From the main menu choose Edit a Data Base File. After the data base editor loads, a list of the data base files available for editing is presented. Use the space bar or arrow keys to position the pointer in front of the desired file name, press RETURN/ENTER.

The file format screen will be displayed showing the field names with the type and length of each field. Press function key 3 EDIT to edit a data base record. Enter the number of the record to be edited. Each field will be displayed with the current value. If no change is desired press RETURN/ENTER. To make a change enter the new value. When all fields have been displayed the field format screen will again be displayed. Repeat the above process for all records desired to edit or press function key 8 MENU, to return to the Insight 2+ main menu.

Save Parameters. The Save Parameters option is not implemented in LRS. An explanation screen is displayed stating the data base files are updated during the LRS requirement/resource matching process.

To save the status of a data base file for future use, use the DOS copy command to copy the file information in SATELITE.DBF, VEHICLE.DBF, or PAD.DBF to another file with a different name. To use the file in a future session the file will need to be renamed as described above in Load Saved Parameters.

Save Resource Schedule. The Save Resource Schedule option is not implemented in LRS. The schedule information is automatically saved in the file SCHEDULE.DAT during the scheduling process. This file must be copied to another file under a different name to save the information for future use. Use the DOS copy command to copy the SCHEDULE.DAT file.

To save a partially executed schedule, the Insight 2+ Save Session feature must be used. See the Insight 2+ Reference Manual for instructions on using this feature of Insight 2+. In addition to using the Insight 2+ Save Session feature, the data base files must also be saved to allow LRS to continue the scheduling process at a later time. To save these files use the DOS copy command and copy the SATELITE.DBF, VEHICLE.DBF, and PAD.DBF files using different names. The files must be renamed to the original file names used by LRS prior to restoring the saved session

Quit. The Quit option ends the LRS session An END OF SESSION message screen is displayed by Insight 2+ Press function key 8 MENU to return to the Insight 2+ main menu

Function key 10 EXIT is the Insight 2+ exit key. Pressing this key will exit Insight 2+.

LRS Output

LRS can send information to the screen or to the printer. Data base file information is sent only to the device specified when the view information choice is made. Scheduling information is sent to the designated device and to the file named SCHEDULE.DAT.

The SCHEDULE.DAT file information is stored in ASCII format. This allows using the DOS type command to view the file after completion of LRS, or to send the file to the printer. The file may also be entered into many word processors for editing or formatting before printing.

The SCHEDULE.DAT file is over written at the beginning of each LRS session. This erases all information from the previous session. This information may be saved as explained above in Save Resource Schedule.

During each LRS session all schedule information for each run accomplished during that session is saved in the SCHEDULE.DAT file. The information from each run is appended to the end of the information from the previous run

Using Multiple Data Base Files

LRS accesses the same data base files for each session. To use different data files for a session the files to be used must be renamed to the file names used by LRS. The

satellite data file is named SATELITE.DBF. The launch vehicle resource file is named VEHICLE.DBF, and the launch pad resource file is named PAD.DBF.

The easiest way to create a new data base file is to copy an existing file and edit the records to make the new file. In this way there is no danger of accidentally using the wrong file format. The file format is critical to proper LRS operation.

A new file may be created using the procedures explained below in How to Create Data Base Files. This procedure is not recommended since there is no error checking performed by Insight 2+ or LRS when the data base files are accessed. Having the wrong format or wrong field information will cause a program error. The type error depends on the change made in the data base file.

Entering Data Base File Information

The information used in each data base file is critical to the proper operation of LRS. This information is the basis for all requirement and resource matching. What information should be entered in each field is explained below.

Satellite Data File. The satellite data file field names are provided in Table V with a brief description of the meaning of each field. A detailed explanation is provided below.

Name Name is the name of the satellite This may be any letter number combination including spaces This information is used only as an identifying name by LRS

Constel Constel is the name of the constellation the satellite is a member of This information is not currently used in LRS execution, but is provided for reference purposes

Weight Weight is the satellite launch weight The units used are not important as long as the same units are used for all weights entered in all data base files This information is used to see if the satellite is too heavy to be launched by a particular vehicle

Size Size is the satellite volume size The units used are not important as long as the same units are used for all weights and boost capability in the data base files This information is used by LRS to assure the vehicle is not too large to be launched by a particular vehicle No consideration is given to any specific measurement, only overall size

Orbit Orbit Alt is the orbit altitude required by the satellite The units used are not important as long as the same units are used for all altitude information is all the data base files This information is used to see if the launch vehicle is capable of the required launch orbit.

Lch Veh. Lch Veh 1 and Lch Veh 2 are the types of launch vehicles which are compatible with the satellite

requirements This is an alphanumeric name specified by the scheduler The name used is not important except that it be exactly the same name used for the type vehicle which the satellite should be launched on LRS makes a comparison of the launch vehicles listed for the satellite and the type launch vehicle available to match the two. The names must be exactly equal including uppercase characters and spaces for a match to be made

Priority Priority is the launch priority of the satellite 1 is the highest priority and 9 is the lowest priority Any integer between these numbers may be used to assign the satellite launch priority. LRS matches satellite requirements each day by the priority of each satellite.

Restrictn. Restriction is a comment line for listing any specific satellite restrictions. Any alphanumeric characters may be used. This information is not used by LRS, and is provided for scheduler information only.

Classified. Classified is a True/False value used to mark the satellite as being a classified launch or not. This information is not used by LRS.

Lch w/clas. Lch w/clas stands for launch with classified launches. This is a True/False value to mark a satellite as able to be launched with other classified launches or not. This information is not currently used in

LRS but planned for future use in using multiple launches per vehicle.

Desire Lch. Desire Lch is the desired launch day for the satellite. This is the schedule day the satellite is desired to be launched. This number is used by LRS to determine when to consider the satellite for launch. This is the first day the satellite will be attempted to be matched with an available launch vehicle. Desired launch may be any integer from 1 to 999999.

Available. Available is the first schedule day the satellite is available for launch. This is not currently used by LRS. It is designed to allow launching a satellite before the desired day to make more efficient use of launch resources during schedule optimization.

NLT Lch. NLT Lch is the not later than launch for the satellite. If the satellite is launched with all required launch resources by this day it is considered not launchable and marked as missing its launch. This may be any integer from 1 to 999999. The number of days between the desired launch and the NLT launch determines the launch window for the satellite.

Orbit Type. Orbit Type is the type orbit required by the satellite. This is an alphanumeric value which is not currently used by LRS. It is provided for scheduler information and use in later versions where certain orbits

are only capable on certain vehicle types or from certain launch sites.

Inclination. Inclination is the orbit inclination required for the satellite orbit. LRS version 1.0 does not consider inclination limitations. This would be considered in a later version where a launch vehicle may not be able to achieve an inclination from a certain launch site.

Launched. Launched is the launch status of the satellite. This is a character value used as a flag for LRS. If the value is "n" the satellite is not launched and available for consideration. Any value other than n causes LRS to eliminate that satellite from further consideration during the scheduling process. If the value is "x" the satellite has missed its launch. A value of "1" indicates the satellite is launched.

Vehicle Data File. The vehicle data file field names are provided in Table VI with a brief description of the meaning of each field. A detailed explanation is provided below.

Name. Name is the alphanumeric name for the vehicle. This information is used for schedule identification purposes only.

Type. Type is the vehicle type. It is an alphanumeric value which may include spaces. The value entered is not critical as long as it exactly matches the value entered for one of the satellite launch vehicle types.

LRS uses an exact match including case and spaces to determine if the vehicle is of the proper type to launch a satellite. The value entered here must exactly match the value in the satellite Lch Veh 1 or Lch Veh 2 field for the launch vehicle to be matched with the satellite.

Boost Cap. Boost Cap is the boost capability of the launch vehicle. This is the weight the launch vehicle can place into orbit. The units used is not important as long as they are the same units used for the satellite weight. The satellite weight must be less than this value for the two to be matched.

Payload Sz. Payload Sz is the physical size of the launch vehicle payload bay. The units are not important as long as they are the same units used for the satellite size. Any specific dimensions are not considered by LRS, only the overall size. The value of a satellite's size must be less than the value entered here for the vehicle for the vehicle to be used by the satellite.

Orbit Cap. Orbit Cap is the orbit the vehicle is capable of achieving. The units must be the same as the units used for the satellite orbit altitude. The launch vehicle orbit capability must be greater than or equal to the satellite required orbit altitude.

Launch Fac. Launch Fac is the launch facility where the launch vehicle is located. This is an alphanumeric value used to compare with the location of the

launch pad. It these two values are not exactly the same the vehicle cannot be matched to the launch pad. This includes spaces and letter case.

Msn Dur. Msn Dur stands for mission duration. This is the number of days the vehicle can support a satellite launch. This number is not currently used in LRS resource matching.

Turn Time. Turn Time is the number of days from launch until the launch vehicle can be ready for another launch. This number is added to the launch day to determine when the launch vehicle will be next available for launch. The turn time for an expendable vehicle is the production time of the vehicle.

First Aval. First Aval is the schedule day the launch vehicle is first available for launch. This may be any integer from 1 to 999999. This value is used to set the next available day when the data file is reset using the Reset DB File option of LRS.

Next Avail. Next Avail is the schedule day the satellite is next available for launch. This value is set by LRS during the reset data base file process and during the scheduling process. The vehicle can be considered available for launch if the current schedule is equal to or greater than the next available launch day. This may be any integer from 1 to 999999.

Pad Data File. The launch pad data file field names are provided in Table VI with a brief description of the meaning of each field. A detailed explanation is provided below.

Name. Name is the alphanumeric character name for the launch pad. This is identifying name used during the scheduling process.

Type. Type is the launch vehicle type the launch pad is capable of launching. This must match exactly with the vehicle type entered for the launch vehicle. This includes letter case and spaces.

Launch Fac. Launch Fac is the launch facility where the launch pad is located. This is an alphanumeric value which must match exactly the launch facility entered for the launch vehicle. The letter case and spacing is critical.

Turn Time. Turn Time is the number of days required to get the launch pad ready for another launch after a launch has occurred from the pad. This number is added to the launch day to determine when the launch pad will be next available for launch consideration. This may be any number for 1 to 999.

First Aval. First Aval is the schedule day the launch pad is first available for launch. This number is used by LRS to set the next available launch day during

resetting the data base file. It may have a value from 1 to 999999.

Next Avail. Next Avail is the schedule day the launch pad will be next available for launch. This value is set to the value of first available launch day when resetting the data base file. LRS computes the value to store here during execution by adding the turn time to the launch day. The launch pad is not considered for launching a vehicle until the current schedule day is equal to or greater than the value stored here.

How to Create Data Base Files

To create a data base file to be used with LRS use the following procedure:

1. Load the Insight 2+ program.
2. Choose Edit Data Base from the Main Menu.
3. Press Function Key 2 NEW to start a new data base.
   If a file already exists with the name you want to use for the new file, you will need to delete the file before you can create a new one with that name. To delete the file use the Delete function of the Data Base Editor or use the DOS del command from a DOS prompt.
4. Enter file name:
   use VEHICLE for vehicle information
   use SATELITE for satellite information

use PAD for launch pad information

Any name desired may be used when creating a file but only the files with the above names will be used by LRS. A DBF extension is added to all files created with the Insight 2+ data base editor.

5. Choose DB II format using the space bar. The bright letters on the dark background bar is the item currently selected. After choosing the format to use, the top of the screen will show which data base format is being created.

6. Create the fields to be used in the data base.

This is done by entering the proper information to define each field as prompted by the data base creator program. Use the information presented below in tables I thru III to define each field to be used in the data base. When all the fields have been defined Press Function Key 4 DONE to end the creation process and return to the data base editor.

It is very important to enter the fields exactly as described below in Tables XXIV to XXVI. This includes the order of the file information as well as the format definitions. The LRS program depends on the data base files having this exact format to operate properly. If a field is entered

in error the creation process must be started over for that file.

The files may be created or edited with dBase II or dBase III but the field definitions and order must be kept the same. If dBase III is used to create the files be sure to use the dBase II format option or the files will not be compatible with the LRS program.

7. Add the new information to the data base.

Once the data base files have been defined new information may be added, changed, or listed using the data base editor of Insight 2+ or the dBase II or dBase III programs.

New information is added to the data base by pressing Function Key 4 ADD. This presents each field heading allowing entry of the value for that field. You may List the data base file information by pressing Function Key 2 LIST, or Edit an existing record by pressing Function Key 3 EDIT.

8. End the Edit session.

The Edit session can be ended by returning to the Main Menu using Function Key 8 MENU. During the LIST process Function Key 8 is labeled BACK. Pressing this key will return you to the Edit file

display screen from which the Edit session may be ended or another choice made.

TABLE XXIV

Data Base creation information for the SATELITE data base

Field	Name	Type	Width	Decimal
1	Name	C	30	
2	Constel	C	30	
3	Weight	N	10	0
4	Size	N	5	0
5	Orbit Alt	N	7	0
6	Lch Veh 1	C	30	
7	Lch Veh 2	C	30	
8	Priority	N	2	0
9	Restrictn	C	50	
10	Classified	L		
11	Lch w/clas	L		
12	Desire Lch	N	6	0
13	Available	N	6	0
14	NLT Lch	N	6	0
15	Orbit Type	C	10	
16	Inclinatn	N	3	0
17	Launched	C	1	0

TABLE XXV

Data Base creation information for the VEHICLE data base

Field	Name	Type	Width	Decimal
1	Name	C	30	
2	Type	C	30	
3	Boost Cap	N	10	0
4	Payload Sz	N	5	0
5	Orbit Cap	N	7	0
6	Launch Fac	C	30	
7	Msn Dur	N	3	0
8	Turn Time	N	3	0
9	First Aval	N	6	0
10	Next Avail	N	6	0

TABLE XXVI

Data Base creation information for the PAD data base

Field	Name	Type	Width	Decimal
1	Name	C	30	
2	Type	C	30	
3	Launch Fac	C	30	
4	Turn Time	N	3	0
5	First Aval	N	6	0
6	Next Avail	N	6	0

Appendix D: Insight 2+ Lessons Learned

The following information is provided as a guide to follow on work with Insight 2+. Some of the information is covered in the Insight 2+ User Manual, most was discovered by trial an error during the Prototype programming effort. Some of the information was obtained from conversations with Insight 2+ technical personnel.

All information is based on Insight 2+ version 1.2 which was used to create LRS. A copy of the lessons learned was sent to Karl Seiler at Insight 2+ for comment. Mr. Seiler stated many of the errors are corrected in the current version 1.3 (36). The changes made in version 1.3 are indicated following the lesson learned from version 1.2.

Goals

A Goal with no associated rule will cause a question to be asked to see if the goal is true or false.

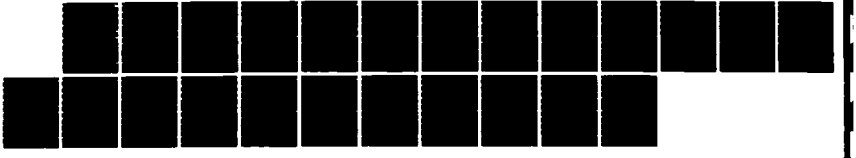
All goals must be numbered sequentially starting with 1. The inference engine will seek to satisfy goals in the order of the goal outline. There must be a rule with a THEN conclusion for each goal, or the program will stop with a further conclusions possible message.

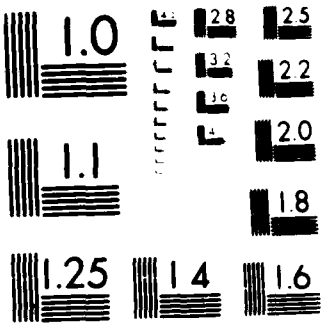
Rules

Rules are attempted in the order found in the rule base. The search for an applicable rule

AD-A179 426 LRS: A KNOWLEDGE BASED APPROACH TO LAUNCH RESOURCE SCHEDULING(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING F H KOCH DEC 86 3/3

UNCLASSIFIED AFIT/GSO/ENS/86D-4 F/G 22/4 NL





THEN conclusion of each rule. See the I2+ manual for more details.

When a rule asks if the conclusion of another rule is true, and the conclusion is not already known, the inference engine will execute the rules which lead to that conclusion.

The process will continue until a conclusion can be reached for the desired rule or until all rules with that as a conclusion are exhausted in which case the program ends with the statement no conclusion can be reached.

Rule conditions are executed sequentially until a false statement is found. Due to this it is important to place all conditions before actions, if you only want the actions accomplished if the rule is true. ie. Part of a rule can be executed but the then part of the rule not executed due to one condition in the rule not being met.

For example:

```
RULE Partial rule execution
IF Forget all data
  AND DISPLAY all data forgotten
  AND Confirm forget all data
THEN Action IS forget all data
  AND FORGET all data
```

If the choice to forget all data is made, the all data forgotten display will be shown before the confirmation to forget the data is asked. The

display will be shown regardless of the answer to the confirmation question.

When there are several groups of rules which need to be examined only at specific times during the KB execution, use one common goal with steps to select the applicable rules. This procedure is demonstrated in the sample KB provided called PACKING.PRL.

This situation occurs when a menu is used, especially with sub menus, where the CYCLE command is used to get back to the menu for another choice.

Forward chaining is accomplished by having a group of rules all have the same THEN conclusion. Since the inference engine tries to satisfy rules with a THEN conclusion which matches the current goal all rules with a common conclusion will be considered. The goal must be defined as MULTI to prevent the inference engine from stopping once the goal is satisfied the first time. This is demonstrated in the sample knowledge base called PACKING.PRL.

The ELSE feature must be used with caution, especially when using a forward chaining strategy. The ELSE conclusions will always be executed if the THEN part of the rule cannot be met. This means the first ELSE found in a series of rules with the same THEN conclusion will be executed if all the rules above it in the knowledge base are false.

The OR function did not operate properly at all times. The case where it failed during testing had an OR condition which was true but search was continued to satisfy the second condition of the OR as if the first condition was false. To correct this problem the rule was broken into two separate rules, one to consider each OR case. This worked correctly. The problem may stem from there being several AND conditions after the OR conditions. The rule which had to be changed can be examined in the program code in Appendix B. It is now the two rules titled SRM: Schedule Resources out of schedule time and SRM: Schedule Resources all requirements scheduled. The OR conditions were Out Of Schedule Time OR All Requirements Scheduled.

Commands

The STOP command stops the inference engine from looking for another attribute of an object which has been designated as having multiple attributes with multi or exhaustive.

FORGET will not work to forget goals in the KB unless it is used in the Command section of the KB in conjunction with the CYCLE command. When the FORGET command is used in a rule to forget a goal the goal is set to false, not reset to an unknown status. (Corrected in version 1.3 (36)).

MULTI and EXHAUSTIVE only apply to object-attribute pairs, and in some cases simple facts.

The ASK command will always ask for the value of the

fact and replace a previous value if one exists.

Compiler

Any text following an ! anywhere in the text is treated as a comment by the compiler and will not be used in the program code. This includes ! within text displays. Which means an ! cannot be used except as a comment marker. (Corrected in version 1.3 (36)).

Rules, text, etc. are not ended until another Reserved Word is found by the compiler. This means if a comment line has the ! deleted it becomes part of the last Reserved Word command used. (Corrected in version 1.3 (36)).

ie. Each Reserved Word marks the beginning of a block of text to be used by that Reserved Word. The block is ended by the next Reserved Word. All text between Reserved Words is considered part of the command syntax with lines of text following ! excluded.

Words known to start new blocks are:

RULE
DISPLAY
TEXT
EXPAND
TITLE

Blank lines are not significant in RULE blocks but they are significant in any other block, and will affect the information displayed.

y is considered a reserved word in some cases.

ie. answer = y will not compile in some cases.

(Corrected in version 1.3 (36)).

The compiler locks-up at different times when an error is encountered during compiling. This appears as an inability to accept any commands from the keyboard requiring a reboot of the system. Since the program automatically saves the file before compiling the only thing lost is the time to reboot. (Corrected in version 1.3 (36)).

The same phrase which describes an object attribute cannot be used as a DISPLAY name. ie. If "Print Screen" is a possible attribute of the object "Action to perform", a display defined by "DISPLAY Print Screen" will cause a compile error. (Corrected in version 1.3 (36)).

Every DISPLAY text defined must be called somewhere in the rules portion of the program or a compile error will occur.

Compiling is good for execution speed, but makes development slower due to the compile time required after each change is made. To reduce this time use small test files to debug a rule or small group of rules which work together. Once the rules are working properly include them in the main program. This can significantly reduce the time spent waiting on compilation of a large program when making many small changes trying to debug a rule operation.

I had trouble using the Include "$" command to include partial rule bases during compiling. I did this to try to allow partial editing of the rule base without having to

load the entire file each time into the editor. I got an error "END EXPECTED" after the second set of rules was included. I don't know what the problem was, but I went back to using the entire file each time. I have had no problems using include for text only files, as with the DISPLAY command. (Corrected in version 1.3 (36)).

Facts and Variables

Strings can only have one value at at time in the KB. Only Objects can have multiple values.

One Object cannot be assigned the value of another object. ie. Object 1 := Object 2 is not a valid operation.

A single attribute of an object cannot be forgotten, only all attributes of an object.

It is not possible to change the number of variables during program operation in Insight 2+. The number of facts to be used must all be in the program at the time it is compiled. This means you must have designated names for all the facts you plan to use, even if they will not all be used every time the program is run. ie. You want to be able to handle up to 20 different constellations with up to 60 satellites in each constellation. You must designate fact names for 1200 entities at compile time even if you may only need 10 of them for small runs. This can be a major problem for programs which do not require the same number of facts for each program run.

The way around this problem is to use the data base

interface of I2+ to store the information. Then use DBPAS programs to access this information. The file access is limited by the disk access speed, but it is a workable solution if a non constant number of variable values is required.

To allow a string or numeric fact to be only entered once during program operation even if CYCLE is used, use the following procedure.

To obtain the first value check to see if the fact is not equal to some value it can never equal. I2+ will ask for the value of the fact the first time the rule is encountered to make the comparison. Since the fact can never equal the value, the statement is always true. From then on when the rule is executed the value for the fact which is already known will be used to make the comparison and no input will be requested. If a new value is desired for the fact just use the FORGET command to forget the value and when the rule is executed again I2+ will ask for the value of the fact.

```
ie. RULE Enter Name
    IF Name <> dummy
      AND ...
    THEN Action
```

Files

Insight 2+ requires the file name to be used for program output be specified in the program code. There is no provision to allow selection of the file name during

program execution. This may possible using an external program which is called by Insight 2+, however the external program must do all the file operations and pass the desired information back to Insight 2+.

All DB operations must be accomplished by a call to a DBPAS program which executes the required operations and passes the desired information to the KB. There are sample programs given but the actual program used must be written to match your requirements.

General

A working knowledge of PASCAL is essential if any data base access is to be used. All data base access is accomplished through DBPAS programs which are a subset of PASCAL. Good examples are provided in the sample programs, but they must be modified to meet the needs of the program being designed.

Be sure to make the CONFIG.SYS file modifications called for in the READ.ME file or you may get a RUN-TIME ERROR F0 during compiling using include files. I don't know the cause but that was the fix I found.

The only way to stop the I2+ program if it gets into an infinite loop is to reboot the computer. (Corrected in version 1.3 (36)).

The program has locked-up several times during program execution. Not nearly as many times as during compiling. I

have no idea what the cause is. (Corrected in version 1.3 (36)).

The program locked-up during the create Knowledge Tree option with a disk assess error message. (Corrected in version 1.3 (36)).

The program locked-up during create Knowledge Tree option when it listed a possible infinite recursion. (Corrected in version 1.3 (36)).

DBPAS

The maximum variable length in DBPAS is 16 characters.

The command to allow sending information to a printer is not implemented in DBPAS. The only way to print a DB file during program operation is to pass the information back to the KB and use the print command within the PRL commands. This is much more difficult, especially if you need to print all the records in a file since there is no looping structure built into Insight 2+ PRL commands. (Implemented in version 1.3 (36)).

> An alternative is to use another language such as Turbo Pascal to do the printing with an external CALL to the program. The only problem with this is these programs are not designed to interact with the data base files.

The var command must be present with at least one variable defined for a DBPAS program to compile correctly.

When trying to assign a value to a string variable with length 1 in DBPAS a blank must be appended to the value due to a bug in the program. ie. string := 'n '; will assign the value n to string. This is not required for string variables with a defined length longer than 1. (Corrected in version 1.3 (36)).

DBPAS boolean parameters passed back to I2+ must be defined as Simple Facts, not Strings. To assign values use true and false in the DBPAS code.

If parameters are not being passed properly to a DBPAS program, try changing the order in which the parameters are passed. (Corrected in version 1.3 (36)).

ie. DBPAS program defined by:

```
program test (receive Rec_num : integer;
                      Mark_val: string(1));
```

always showed the Rec_num parameter being 0, no matter what was passed, changing the definition to:

```
program test (receive Mark_val : string (1);
                      Rec_num : real);
```

worked just fine.

When passing parameters back to Insight 2+ from an external program, always pass back the proper number of parameters using proper format. When accessing data base files, this may necessitate passing dummy parameters if the end of the data base is reached. Not returning the values for some of the parameters after having already returned the

proper parameters previously may return the previous parameters, but this cannot be relied on. A mismatched parameter value error after a routine has operated properly in the past, may be due to this problem. Sample programs showing how to handle accessing nonexistent data base records is shown in the DBPAS code in Appendix B in the three Get Data Base information files.

Appendix E: LRS File Output Sample

The following is a printout of the information stored by LRS during the execution of Test F. This gives a sample of the information available in the SCHEDULE.DAT file created during LRS execution. The information is formatted the same way is would appear on a printer using the "Display Information on the printer" option in LRS.

Page breaks are not provided by LRS. The text is broken at convenient points for better reading in this appendix.

```
**********************************************************************************

Product created by Fred on 25 Nov

Test F

**********************************************************************************
|                                                                                |
|  ****************************************************************************  |
|  |                                                                          |  |
|  |                                                                          |  |
|  |                                L R S                                     |  |
|  |                                                                          |  |
|  |                  Launch   Resource  Scheduling                           |  |
|  |                                                                          |  |
|  |                       A  Planning  Tool                                  |  |
|  |                                                                          |  |
|  |                            by                                            |  |
|  |                                                                          |  |
|  |                       Major Fred H. Koch                                 |  |
|  |                                                                          |  |
|  |                                                                          |  |
|  ****************************************************************************  |
|                                                                                |
**********************************************************************************

**********************************************************************************
                              Program Assumptions

    This program uses all deterministic computations.  This means if a
    vehicle launches it always successfully places its payload into orbit.
    If a satellite takes two weeks checkout time it will always take exactly
    two weeks to checkout.  This reasoning was used to simplify the program
```

operation.

All specific parameters for vehicles, launch pads, and satellites are entered by the user. This information is used by the program to provide a possible matching of launch resources to launch requirements.

The schedule provided is not necessarily the optimum schedule. It is a feasible schedule. One which meets all the constraints established by the user entered information and the scheduling rules of the program.

The Starting Day of the Schedule is 1.

The Ending Day of the Schedule is 8.

Satellite Sat 3 missed launch due to no vehicle available
by the ending schedule day. The satellite was available on
day 1.

Satellite sat 1 missed launch due to no vehicle available
by the ending schedule day. The satellite was available on
day 3.

Satellite sat 2 launched on Shuttle 1 from pad 2
on Schedule Day 4.

Satellite Sat 4 launched on Alv 2 from Pad 3
on Schedule Day 5.

The following Satellites were not launched during the schedule period.

sat 1 with desired launch day of 3 and NLT
launch day of 9 and priority of 9.00.

Sat 3 with desired launch day of 1 and NLT
launch day of 5 and priority of 1.00.

The following Vehicles were available at the end of the schedule.

Shuttle 2 available on day 3 was available
at the end of the schedule.

Alv 1 available on day 1 was available
at the end of the schedule.

pad 1 able to launch shuttle available on day 7.

Pad 4 able to launch alv available on day 4.

Bibliography

1. Aderhold, Major Dave, Telephone interview. U.S. Space Command/J3X, Colorado Springs, CO, 16 April 1986.

2. Aderhold, Major Dave. U.S. Space Command J3X. Personal Interview. Peterson AFB, CO, 13 August 1986.

3. Arbabi, Mansur "Range Scheduling Automation," IBM Technical Directions, 10(3):57-62 (1984)

4. Boarnet, Marlon "Resource Planning & Management System (RPMS)." Video Tape by NASA/Johnson Space Center. Available from Major Steve Cross, AFIT Computer Science Department. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, viewed 2 October 1986.

5. Brownston, Lee and others. Programming Expert Systems in OPS5. Reading MA: Addison-Wesley Publishing Company Inc., 1985.

6. Cheeseman, Peter C. and William T. Park. Modeling and Planning Robotic Manufacturing. Contract N00014-83-C-0649. SRI International, Menlo Park CA, March 1985 (AD-A161 014).

7. Forgy, Charles L. OPS5 User's Manual. CMU-CS-81-135. Pittsburg: Carnegie-Mellon University, 1981.

8. Fox, Mark S. Constraint-Directed Search: A Case Study of Job-Shop Scheduling. PhD dissertation. Computer Science Department, Carnegie-Mellon University, Pittsburgh PA 15213, 13 December 1983 (AD-A138 307)

9. Fox, Mark S. The Intelligent Management System: An Overview: Interim Report, Intelligent Systems Laboratory, The Robotics Institute, Carnegie-Mellon University, Pittsburgh PA 15213, 7 December 1982 (AD-A126 345)

10. Fox, B.R. and K.G. Kempf "Complexity, Uncertainty and Opportunistic Scheduling," Proceedings of IEEE 1985 Second Conference on Artificial Intelligence Applications. 487-492. IEEE Computer Society Press, Washington D.C., 1985

11. Fox, Mark S. and Stephen F. Smith "ISIS - A Knowledge-Based System for Factory Scheduling," Expert Systems, The International Journal of Knowledge Engineering, 1: 25-49 (July 1984). (AD-A160 320)

12. Fox, Mark S. and Stephen F. Smith. Constraint-Based Scheduling in an Intelligent Logistics Support System: An Artificial Intelligence Approach: Annual Report, 15 March 1983--14 March 1984. AFOSR Contract F49620-82-K-0017. Intelligent Systems Laboratory, Robotics Institute, Carnegie-Mellon University, Pittsburgh PA 15213, 21 July 1983 (AD-A145 613)

13. Fox, Mark S. and Stephen F. Smith Constraint-Based Scheduling in an Intelligent Logistics Support System: An Artificial Intelligence Approach: Annual Report, 15 March 1984--14 March 1985. AFOSR Contract F49620-82-K-0017. Intelligent Systems Laboratory, Robotics Institute, Carnegie-Mellon University, Pittsburgh PA 15213, 15 July 1985 (AD-A159 043)

14. Gadsden, J.A. "An Expert System for Evaluating Electronic Warfare Tasking Plans for the Royal Navy," Proceedings of the IEEE 1984 Computer Society Conference on Artificial Intelligence Applications. 86-91. IEEE Computer Society Press, Silver Spring MD, 1984

15. Gates, Kermit H. and Toini Figgins Decision Aids for Target Aggregation Command and Control Warfare Strategy Planning Aid (CTA) - Functional Description. Contract F30602-81-C-0263. PAR Technology Corporation, New Hartford NY, May 1984 (AD-B086 773).

16. Gilmore, John F. and others "Knowledge-Based Route Planning Through Natural Terrain," Proceedings of SPIE Applications of Artificial Intelligence II (1985). 548:128-136. SPIE - The International Society for Optical Engineering, Bellingham WA, 1985

17. Goldstein, Ira P. and R. Bruce Roberts "Nudge, A Knowledge-Based Scheduling Program," Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI). 257-263. Department of Computer Science, Carnegie-Mellon University, Pittsburgh PA, 1977

18. Harmon, Paul and David King. Expert Systems. New York: John Wiley & Sons, Inc., 1985.

19. Ho, William P.C. "Intelligent Computer-Aided Design by Modeling Chip Layout as a Meta-Planning Problem," Proceedings of SPIE Applications of Artificial Intelligence III (1986). 635:628-635. SPIE - The International Society for Optical Engineering, Bellingham WA, 1986

20. Insight 2+ Reference Manual. Version 1.0. Level Five Research, Inc., Indialantic FL, 1986

21. Insight 2+ Technical Information Summary. Level Five Research, Inc., Indialantic FL, no date

22. Kaste, Richard C. A Manual for The Artillery Consultant "Battle". Technical Report BRL-TR-2717. US Army Ballistic Research Laboratory, Aberdeen Proving Ground MD, March 1986 (AD-B100 149)

23. Keirsey, David M. and others "Multilevel Path Planning for Autonomous Vehicles," Proceedings of SPIE Applications of Artificial Intelligence (1984). 485:133-137. SPIE - The International Society for Optical Engineering, Bellingham WA, 1984

24. Kohn, W. and others "Automatic Procedures Generator for Orbital Rendezvous Maneuver," Proceedings of SPIE Space Station Automation (1985). 580:40-52. SPIE - The International Society for Optical Engineering, Bellingham WA, 1985

25. Kornell, Jim "A VAX Tuning Expert Built Using Automated Knowledge Acquisition," Proceedings of the IEEE 1984 Computer Society Conference on Artificial Intelligence Applications. 38-41. IEEE Computer Society Press, Silver Spring MD, 1984

26. Levine, Andrew P. "ESP: An Expert System for Computer Performance Management," Proceedings of SPIE Applications of Artificial Intelligence III (1986). 635:90-96. SPIE - The International Society for Optical Engineering, Bellingham WA, 1986

27. Nachtsheim, P.R. and others "A Knowledge-Based Expert System for Scheduling of Airborne Astronomical Observations," NASA Report TM-88194

28. Newman, P.A. and K.G. Kempf "Opportunistic Scheduling for Robotic Machine Tending," Proceedings of IEEE 1985 Second Conference on Artificial Intelligence Applications. 168-175. IEEE Computer Society Press, Washington D.C., 1985

29. North American Aerospace Defense Command, Aerospace Defense Command, Space Command. SPACECOMNIBUS. N/A/S Pamphlet 11-3. Peterson Air Force Base CO, 1 August 1985.

30. Nygard, Kendall E., "Allocate - A Load Planning System for LOGAIR." Address to AFIT students. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 28 July 1986

31. Orciuch, Ed "ISA: Intelligent Scheduling Assistant," Proceedings of the IEEE 1984 Computer Society Conference on Artificial Intelligence Applications. 314-320. IEEE Computer Society Press, Silver Spring MD, 1984

32. Pease, Marshall C. ACS.1: An Experimental Management Tool, Office of Naval Research Contract N00014-71-C-0210. Stanford Research Institute (SRI), Menlo Park CA 94025, November 1976 (AD-A037 311)

33. Pritsker, A. Alan B. Simulation and SLAM II (Second Edition). West Lafayette IN: Systems Publishing Corporation, 1984.

34. Robinson, Ann and David Wilkins Man-Machine Cooperation for Action Planning: Final Report, Office of Naval Research Contract N00014-80-C-0300. Artificial Intelligence Center, Computer Science and Technology Division, SRI International, 333 Ravenswood Avenue, Menlo Park CA 94025, November 1982 (AD-A124 243)

35. Rockmore, A. Joseph, and others Decision Aids for Target Aggregation Route Planning Aid (RPA) - Functional Description. Contract F30602-81-C-0263. PAR Technology Corporation, New Hartford NY, May 1984 (AD-B086 774).

36. Seiler, Karl. Telephone Interview. Level Five Research, Inc., Indialantic FL, 24 November 1986

37. Semeco, Antonio C. and others "GENSCHED - A Real World Hierarchical Planning Knowledge-Based System," Proceedings of SPIE Applications of Artificial Intelligence III (1986). 635:250-256. SPIE - The International Society for Optical Engineering, Bellingham WA, 1986

38. Stockbridge, Capt Samuel E. Autonomous Vehicle Mission Planning Using AI Techniques. MS thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1985 (AD-A163 956).

39. Tate, A. and A.M. Whiter "Planning with Multiple Resource Constraints and an application to a Naval Planning Problem," Proceedings of the IEEE 1984 Computer Society Conference on Artificial Intelligence Applications. 314-320. IEEE Computer Society Press, Silver Spring MD, 1984

40. Whalen, Powell J. and Theodore F. Skowronski. "DOSS - An Expert System for Large Scale Design," Proceedings of SPIE Applications of Artificial Intelligence III (1986). 635:70-75. SPIE - The International Society for Optical Engineering, Bellingham WA, 1986

VITA

Major Fred H. Koch was born on 10 March 1952, the son of Harvey W. and Betty M. Koch, in Cincinnati, Ohio. He graduated from Anderson Senior High in Cincinnati, Ohio in 1970. He was appointed an Air Force ROTC four year scholarship and attended Miami University in Oxford, Ohio. He graduated from Miami University with a Bachelor of Arts Degree in Aeronautics and Mathematics in 1974. He received his commission in June 1974 and served as a Transportation Officer at Columbus AFB, Mississippi. In January 1976 he enter Undergraduate Navigator Training at Mather AFB, California. He graduated from Undergraduate Navigator Training in 1976 receiving an assignment to KC-135s. He served as a navigation instructor at K.I. Sawyer AFB, Michigan in the 46th Air Refueling Squadron until 1981. In 1981 he became a research navigator in the 4950th Test Wing at Wright-Patterson AFB, Ohio. He upgraded to navigation instructor and designed the navigation procedures for the EC-18B aircraft and was the first navigator to fly in the navigator position in the EC-18B. In 1985 he entered the School of Engineering, Air Force Institute of Technology under the Graduate of Space Operations program.

Permanent Address: C/O Roy W. Proctor
1167 Shangrila Dr.
Cincinnati, OH 45230

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b DECLASSIFICATION DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GSO/ENS/86D-4	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION: School of Engineering; 6b. OFFICE SYMBOL (If applicable): AFIT/ENS	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code): Air Force Institute of Technology, Wright-Patterson AFB, OH 45433-6583	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: US Space Command; 8b. OFFICE SYMBOL (If applicable): J3X	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State and ZIP Code): Peterson AFB, CO 80914-5001	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): Fred H. Koch, Major, USAF	
13a. TYPE OF REPORT: MS Thesis; 13b. TIME COVERED: FROM _____ TO _____; 14 DATE OF REPORT (Yr., Mo., Day): 1986 December; 15. PAGE COUNT: 214	
16 SUPPLEMENTARY NOTATION	

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB GR	Artificial Intelligence Scheduling, Planning, Launching
06	04		
12	01		

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

Title: LRS: A KNOWLEDGE BASED APPROACH TO LAUNCH RESOURCE SCHEDULING

Thesis Advisor: Gregory S. Parnell, Lt Col, USAF
Asst Professor of Ops Research

Approved for public release: IAW AFR 190-1
[illegible] 8 April 87
[illegible] W.OLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (AU)
Wright-Patterson AFB OH 45433

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION	
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS ☐	UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL: Gregory S. Parnell, Lt Col	22b TELEPHONE NUMBER (Include Area Code): 513-255-2549	22c OFFICE SYMBOL: AFIT/ENS

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE

This work describes a knowledge based prototype launch resource scheduling tool called LRS. The tool was requested by US Space Command (J3X) for use by planners to evaluate future space launch resource requirements.

A knowledge based approach to the launch resource scheduling problem is used due to its knowledge intensive nature. The LRS prototype is implemented using a knowledge system tool called Insight 2+ on an IBM PC compatible microcomputer. LRS uses menus and explanation screens to make it simple to operate.

The code is written in IF THEN ELSE production rule format making it easy to understand and easy to update as policy and needs change. LRS uses dBase II format files for storing the required information on launch vehicle and launch pad resources, and satellite launch requirements. LRS matches launch resources to launch requirements. A launch schedule is produced which shows how the resources meet the launch requirements. At completion of the matching process LRS provides a list of unsatisfied requirements and available resources. This list may be used by the planner to determine how well the planned launch resources meet the estimated launch requirements.

The necessary enhancements and improvements to convert the LRS prototype into an operational system are identified.

END

5-87

DTIC