
A Knowledge-Based Expert System for Scheduling of Airborne Astronomical Observations

P.R. Nachtsheim, W.B. Gevarter, J.C. Stutz, and
C.P. Banda

(NASA-TM-88194) A KNOWLEDGE-BASED EXPERT
SYSTEM FOR SCHEDULING OF AIRBORNE
ASTRONOMICAL OBSERVATIONS (NASA) 17 p

N87-19972

CSCL 09B

G3/62 Unclass
43548

December 1985



National Aeronautics and
Space Administration

A Knowledge-Based Expert System for Scheduling of Airborne Astronomical Observations

P. R. Nachtsheim,
W. B. Gevarter, Ames Research Center, Moffett Field, California
J. C. Stutz,
C. P. Banda, Informatics General Corporation, Palo Alto, California

December 1985



National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

A KNOWLEDGE-BASED EXPERT SYSTEM FOR SCHEDULING OF AIRBORNE ASTRONOMICAL OBSERVATIONS

P. R. Nachtsheim and W. B. Gevarter
Ames Research Center

J. C. Stutz and C. P. Banda
Informatics General Corporation

KAOS (Kuiper Airborne Observatory Scheduler) is a knowledge-based expert system developed at NASA Ames Research Center to assist in route planning of a C-141 flying astronomical observatory. This program determines a sequence of flight legs that enables sequential observations of a set of heavenly bodies derived from a list of desirable objects. The possible flight legs are constrained by problems of observability, avoiding flyovers of warning and restricted military zones, and running out of fuel. A significant contribution of the KAOS program is that it couples computational capability with a reasoning system.

INTRODUCTION

NASA Ames has a flying astronomical observatory that consists of a C-141 aircraft with a one-meter telescope mounted within the aircraft. The telescope is constrained in azimuth to point directly to the left of the aircraft (270 degrees from the aircraft heading) and to be limited to a range of elevations of 35 to 72 degrees. The main mission of this Kuiper Astronomical Observatory (KAO) is to support infrared astronomical research by making observations at altitudes greater than 37,000 feet--above most of the atmosphere's infrared absorbing water vapor. KAO missions are allocated to research groups on the basis of anticipated scientific value. In order to maximize such value, every effort is made to make efficient use of the available observation time (approximately six and one half hours of the total flight duration capability of seven and one half hours). However, out of the many possible observation sequences that can be constructed from a list of desirable objects to observe for a particular mission, only a few satisfy all the constraints. Constraints and goals include the following:

- (1) Observed objects must be visible in the elevation window (set by user--usually 35 to 72 degrees).
- (2) Flyovers of prohibited military and other zones must be avoided.
- (3) Deadlegs (during which no observation takes place) should be minimized.

- (4) The number of useful observations should be maximized.
- (5) The aircraft must reach its destination (either home base or another base) within the allotted flight time.

A typical flight plan showing the observation flight legs (after climbout from Moffett Field, California) is shown in Figure 1. Note that the flight route clears the prohibited flyover areas and terminates in the vicinity of Moffett Field. (Observe the smooth curvature of each observation leg.) Since the telescope is fixed, aircraft headings during an observation flight leg are completely determined by the location of the aircraft relative to the observation object.

To aid in the planning of such flight routes, two conventional computer programs have been developed over a period of ten years.⁽¹⁾ One program (WINDO) has been used to determine, for a given aircraft location and date, the times (if any) that desired objects will be visible in the elevation window. The other program (KNAV) is used interactively to generate a

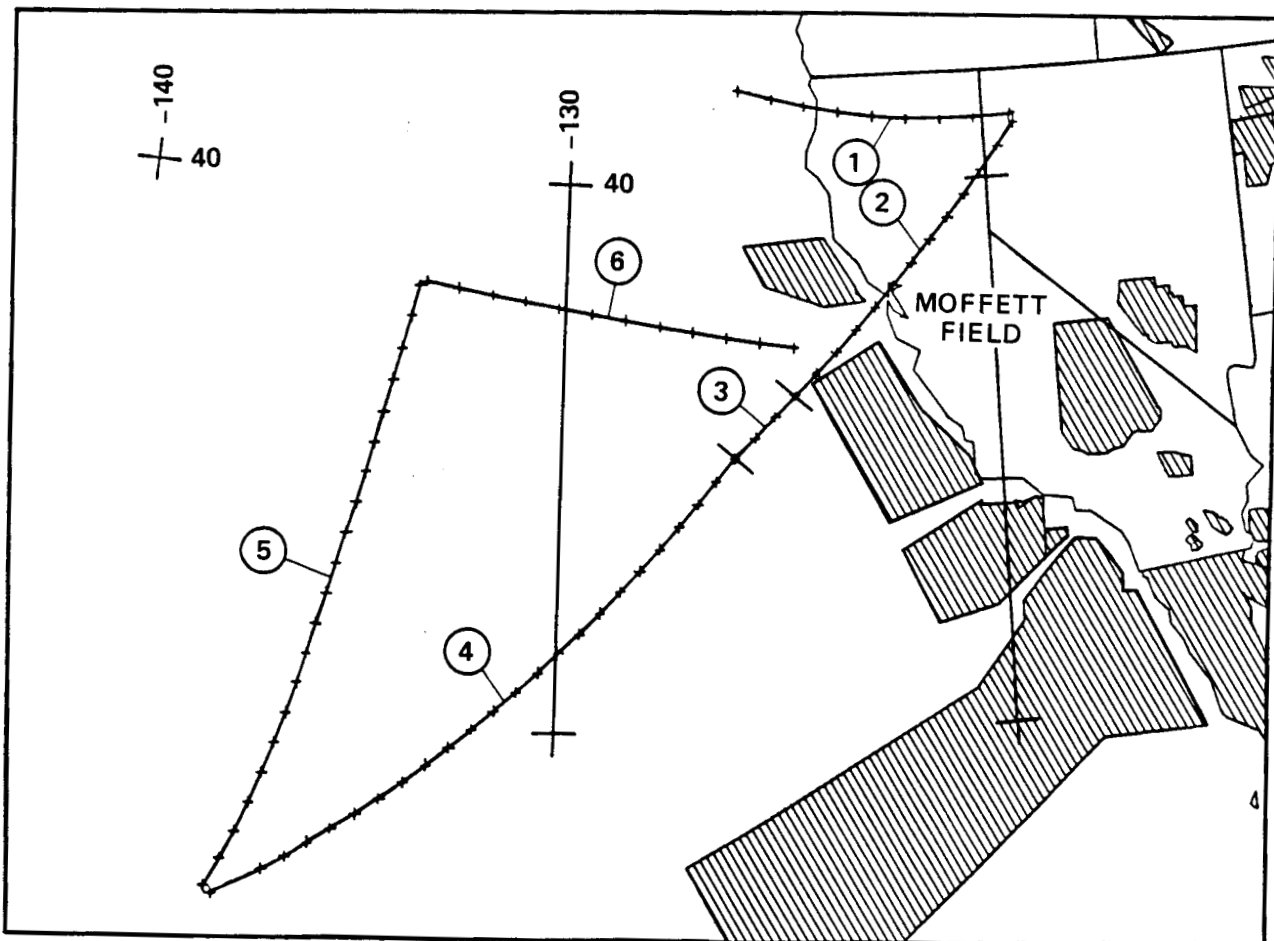


Figure 1
A KAO flight plan showing prohibited overflight areas (diagonal lines) and the six observation legs following climbout from Moffett Field, CA.

sequence of flight legs given a starting location, starting time, and an input sequence of the objects to be observed. Unfortunately KNAV is not designed to automatically satisfy any of the imposed constraints.

APPLICATION OF EXPERT SYSTEMS TO AIRBORNE OBSERVATIONS

Development of an expert system scheduler to facilitate automation of certain aspects of flight planning was deemed practical.⁽²⁾ The expert system is one of the many fields in artificial intelligence that attempts to solve nonanalytical complex problems that require specialized knowledge that most people do not possess. A few such expert systems currently being used to successfully solve practical problems in narrow domains include MYCIN⁽³⁾, MACSYMA⁽⁴⁾, DENDRAL⁽⁵⁾, and RI⁽⁶⁾. A scheduler was developed using expert system techniques which addressed the problem of automatic generation of flight plans for airborne observations. It was named KAOS (Kuiper Airborne Observatory Scheduler).

Expert systems are not procedural programs. The characteristic that most typifies expert systems is that the solution is based on heuristic rules rather than on a deterministic algorithm. A heuristic rule is a procedural tip ("rule-of-thumb") or incomplete method for performing some task.

The expert system derives its knowledge from experts in the field and attempts to "capture" this knowledge in the form of rules in order to emulate the cognitive processes. Since the knowledge in these systems comes from humans, it is difficult to build an expert system which performs better than the person or persons from whom the knowledge was extracted. However, if a substantial amount of knowledge is captured from the experts, this type of system allows many practical problems to be solved by unskilled persons.

The essential features of an expert system are the data base, rule base, and inference engine.

The data base for KAOS consists of the catalog of objects and ephemeris data currently in use at Ames for flight planning. The geographical locations of warning and restricted zones are also in the data base, as well as a consolidated list of flight departure points. Data may be added or deleted by the user.

The rule base contains the rules under which the system operates. These rules are typically of the "if...then..." form, sometimes called situation-action. The situation ("if" portion) of the rule is a condition or state of the problem. The expert system determines if the condition or state is satisfied. Whenever the situation of the rule is satisfied, the rule "fires" and the specified action ("then" portion) takes place. If the situation of a rule is not satisfied, then no action takes place. An example of a situation-action rule in KAOS is:

IF object is not in viewing window

THEN do not schedule object for this observation-leg

The data extracted from the data base along with computations performed on that data typically make up the situation portion of the rule, but another rule's action may also create the situation for a subsequent rule to fire.

The basic mode of operation of KAOS is to generate and test; that is, a leg is calculated for the observation of an object and then is tested. Typically an observation is rejected if (1) the object is outside of the window, (2) the leg overflies a restricted zone, (3) the leg overflies a warning zone, or (4) the completion of a leg leads to a point out of range. These rules can be relaxed by the user, and other rules can be added. In a sense, the system can be "trained." Rules (2), (3), and (4) may be relaxed by the user at run-time. However, rule (1) is always in effect; therefore all the objects in the flight plan will be in the elevation window specified by the user.

The inference engine is a computer program. It is the mechanism by which the rules and data are utilized to reach a conclusion. The inference engine of an expert system looks for "matches" of the situation of a rule to be satisfied by the data from the data base or by the action portion of another rule. Thus, rules may induce a certain dependency on one another. Rules can be "backward-chained" to attempt to satisfy a goal or "forward-chained" to update the world state by adding all facts logically derivable from a given new fact.

Most inference engines employ a pseudo-English-like structure such as prefix predicate calculus. Hence, programming languages such as Lisp and PROLOG which can manipulate such structures are convenient and are frequently used to develop inference engines. The inference engine employed in KAOS was developed at Stanford University. It is written in Lisp and is named MRS, Meta-Level-Reasoning-System.⁽⁷⁾

CAPABILITIES OF KAOS

KAOS can generate flight plans by conducting a search of objects that satisfy the specified constraints. The user inputs a list of objects and the viewing duration for each object. In addition to the rules mentioned previously, additional capabilities are employed to facilitate flight planning.

Additional capabilities include the option of the user to specify the object viewing order, the ability to develop flight plans backward or forward in time, computation and display of object trend information (whether an object is rising, setting, or in transit), and an explanation of why an observation was rejected, e.g.:

O REJECTED: INTERSECTION WITH ZONE R-4806/7/8/9/
O REJECTED: OBJECT OUT OF WINDOW
O REJECTED: OUT-OF-FUEL DISTANCE 16, RANGE 4.7

The user can control the extent of automation desired by selecting various options and selective relaxation of various rules. For example, the user can select the starting time and location or specify a range of starting times and starting latitudes. In the second case KAOS will determine the starting values. There is no loss of generality in omitting starting longitude since changing longitude is equivalent to changing time.

A user's guide is available which describes how to use KAOS.⁽⁸⁾

SUGGESTED STRATEGIES FOR USING KAOS

The current version of KAOS is intended to be used in conjunction with WINDO and KNAV. KAOS is envisioned as an interactive system, which could be used as a training tool for beginning planners as well as a flight planning aide by more experienced ones. Perhaps its greatest strength is that it can generate plans automatically which satisfy specified constraints. However, it is still in the development stage and does not contain enough heuristics to truly mimic an experienced human flight planner. This can lead to very large search spaces since many unpromising plans are pursued for too long a time period. Thus the person running KAOS needs to employ various strategies to reduce the search space and to enable KAOS to find a successful plan.

KAOS can generate all possible flight plans for a limited number of objects, e.g., four or five, in a reasonable length of time. However, when the number of objects exceeds six, it takes an excessive amount of time to generate a single flight plan. The problem is that for a large number of objects the number of possibilities is enormous. Therefore, a good strategy to reduce the search space is to employ "divide and conquer"; that is, to split the flight plan into sections, finding a partial plan for a subset of the objects which does not necessarily end at Moffett Field, and then from this point, to construct a flight plan with the remainder (or a second subset) of the objects.

Another serious problem for KAOS is finding a path through the rather narrow corridors between the warning zones off the coast. Here a good heuristic is to select a point within a corridor and to run KAOS backward to develop a plan from the corridor point to Moffett. The next procedure is to run KAOS forward from the corridor point to complete the plan.

To correct the problem of the plan ending too far north or south of Moffett, the user can alter the viewing order of objects in order to obtain a rough balance of rising and setting objects. For example, since viewing of setting objects drives the aircraft northward, if the final position

ends up too far south of Moffett, one could drive the flight north to a better ending location by finding an object, currently viewed while rising, which could be viewed later in the plan as a setting object.

The flight plan shown in Figure 1 was prepared following the instructions in the user's guide. This flight plan is to be compared with the flight plan shown in Figure 2 which was prepared by a navigator at Ames Research Center and was the flight plan actually flown. It is noteworthy that the navigator opted to overfly a warning zone in order to arrive at a flight plan. (The aircraft may enter a warning zone if FAA approval is obtained.) Aside from the different warning zone behavior, the two plans are similar. The sequence of objects viewed is the same for both. The solid lines shown in the second plan are legs during ascent and descent when no viewing is performed.

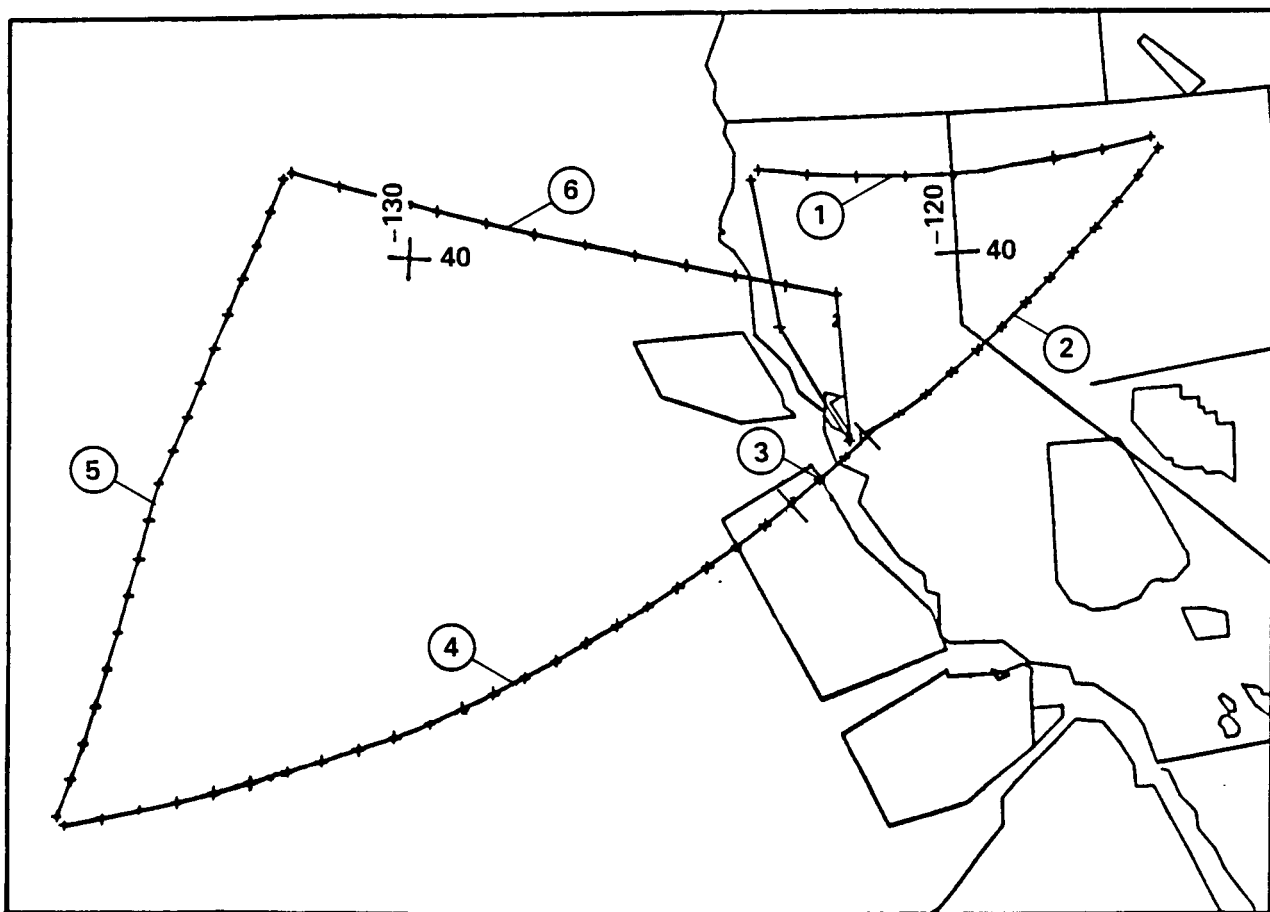


Figure 2
A KAO flight plan prepared by a navigator. (Solid lines represent climb-out and return to Moffett.)

IMPLEMENTATION OF ASTRONOMICAL COMPUTATIONS AND EXPERT SYSTEMS FOR AIRBORNE OBSERVATIONS

KAOS addressed the problem of the highly constrained moving observer represented by the KAO aircraft. It was implemented in MRS in a Franz Lisp⁽⁹⁾ background and in a Eunice environment on a VAX 11/780, and was later transferred to a Symbolics-3600 Lisp machine. KAOS uses MRS's facility for making nonmonotonic assumptions to link a sequence of observation legs into a flight plan.

It is apparent that in order for KAOS to conduct the search for a successful flight plan it needs accurate quantitative modeling of the locations of the heavenly bodies and the aircraft. This need was satisfied by coupling the symbolic reasoning program, MRS, with the needed codes from the WINDO and KNAV modules. These were implemented when the modules were written in FORTRAN and later when they were rewritten in Lisp. The ability of the implementation of Franz Lisp to handle "foreign function" permitted coupling to occur.

The MRS language incorporates mechanisms for representing propositions as function calls. Specific predicates may be declared to use one of several representations using the "REPN" meta-relation. The standard representations allow function evaluation for truth or value, using quoted or evaluated arguments. MRS variables in the function argument list are instantiated with their current bindings before evaluation. It is up to the user to assure such bindings exist at evaluation time. MRS also provides a general predicate "is" for making Lisp function calls to obtain a binding. When forward chaining, the "runnable" predicate can be used to evaluate a function for side effects. These standard mechanisms were supplemented by providing further generic function evaluation predicates based on the Lisp "eval" and "apply" functions. Versions of these functions were written for any of the representations that were needed.

These function calls were implemented using the MRS "LISP" relation. In MRS (lisp <sym> <op>) means that <op> is the Lisp function used to compute the function denoted by the symbol <sym>. The Lisp <op> may be any Lisp function including user-defined functions. For example, if we denote pattern matching variables as symbols prefixed by a dollar sign (\$) as is usual, then the following query in the syntax of MRS

```
(and (= $x USA) (GNP $x $y))
```

will bind \$y to "gross-national-product" of the USA provided that the statement (lisp GNP gross-national-product) is in the data base where gross-national-product is a user-defined Lisp function that performs the calculation. In addition, the "REPN" statement that was mentioned previously must also be in the data base. Utilization of these MRS relations provided easy access to all Lisp facilities and through Franz Lisp to other languages.

DEVELOPMENT OF KAOS

It is instructive to trace the development of KAOS since this illustrates the application of artificial intelligence concepts to the solution of nonanalytical complex problems.⁽¹⁰⁾ The concept of partial programming was actually used to develop KAOS.⁽¹¹⁾ For the problem at hand, a partial program can be defined as any set of constraints of the potential actions of the C-141 flying astronomical observatory. In partial programs, more than one action can be executed at each point in time. The main advantage of partial programming for this application was that it facilitated incremental program development.

The requirements of the flight plan are dictated by the astronomer. He specifies the set of objects and the time of observation on each object. The rules invoked in the program to impose the constraints were described previously as well as the fact that the rules can be relaxed.

The program was developed incrementally by adding one rule at a time. Legs were calculated for the observations of the objects and then were tested. Computations were performed for a selected object which provided for each leg, a start-time, finish-time, and latitude and longitude for each way-point along the leg. Initially, the only requirement imposed was that the legs be contiguous and that no object be viewed more than once.

The next step was to address the observability of the objects; for this reason, the description of a leg was augmented to include the elevation of the selected object. All the computations performed to describe a leg were performed outside of MRS by utilizing the WINDOW and KNAV modules described earlier. In artificial intelligence jargon the results of these computations represent the state of the "world."

The basic rule which served as the generator of the generate-and-test method included the information about the legs. The situation portion of the rule contained the leg information, and this information was coupled to the symbolic reasoning as described previously by MRS mechanism for representing propositions as function calls. The action portion of the rule stated it was permissible to assume an observation of the selected object on this leg unless some test rule failed. Each assumable leg was screened by all the test rules. For example, if the elevation of the object was outside the observation window, then that observation-leg would be rejected.

The flight-plan in the course of automatic-planning could shrink as well as grow. This is referred to as nonmonotonic reasoning. The MRS residue⁽¹²⁾ facility is used to satisfy the goal of developing the flight plan. The residue facility is ideal for producing automatic plans such as this one in that it takes the required observations and a set of values describing the assumptions the program is allowed to make as input. This process is a departure from the proving of traditional theorems in that assumptions can

The previously described rules were added incrementally. Obviously, the added rules must be orthogonal to the earlier rules in the sense that rules do not contradict each other.

The diagram is a technical map or chart. It features a large circular area on the left side, with a smaller circle inside it. A line with cross-ticks runs diagonally from the bottom left towards the top right, passing through the circles. Another line with cross-ticks runs horizontally across the middle. A third line with cross-ticks runs vertically on the right side. There are six numbered points (1-6) in circles. Point 1 is at the top right, near the intersection of the horizontal and vertical lines. Point 2 is just below point 1. Point 3 is on the diagonal line, inside the smaller circle. Point 4 is on the diagonal line, outside the circles. Point 5 is on the diagonal line, further down and to the left. Point 6 is on the horizontal line, near the vertical line. A coordinate system is shown with labels 140, 130, and 40. A small inset map in the top right corner shows the location of the main map area within a larger geographical context.

9

location of the geographic point which marks the completion of a proposed leg as center, a circle is drawn with its radius equal to the range at that point. If the destination, usually Moffett, does not lie within the circle, the proposed leg is rejected.

The representation of the warning and restricted zone was essential. KAOS does not have the ability of human planners to visually recognize (from a graphic representation) overflights of restricted and warning zones. Two methods were developed to represent and check for warning and restricted zones. The user may choose one or the other (or none) at run-time.

The first method represented the terrain as a two-dimensional array. The latitude and longitude of a curvilinear square represented the rows and column of the array. The information stored in the value cell of the array indicated the presence or absence of a restricted or warning zone. This method provided a quick although crude representation of the terrain. Figure 4 shows this method of representation using "squares" with one degree to a side.

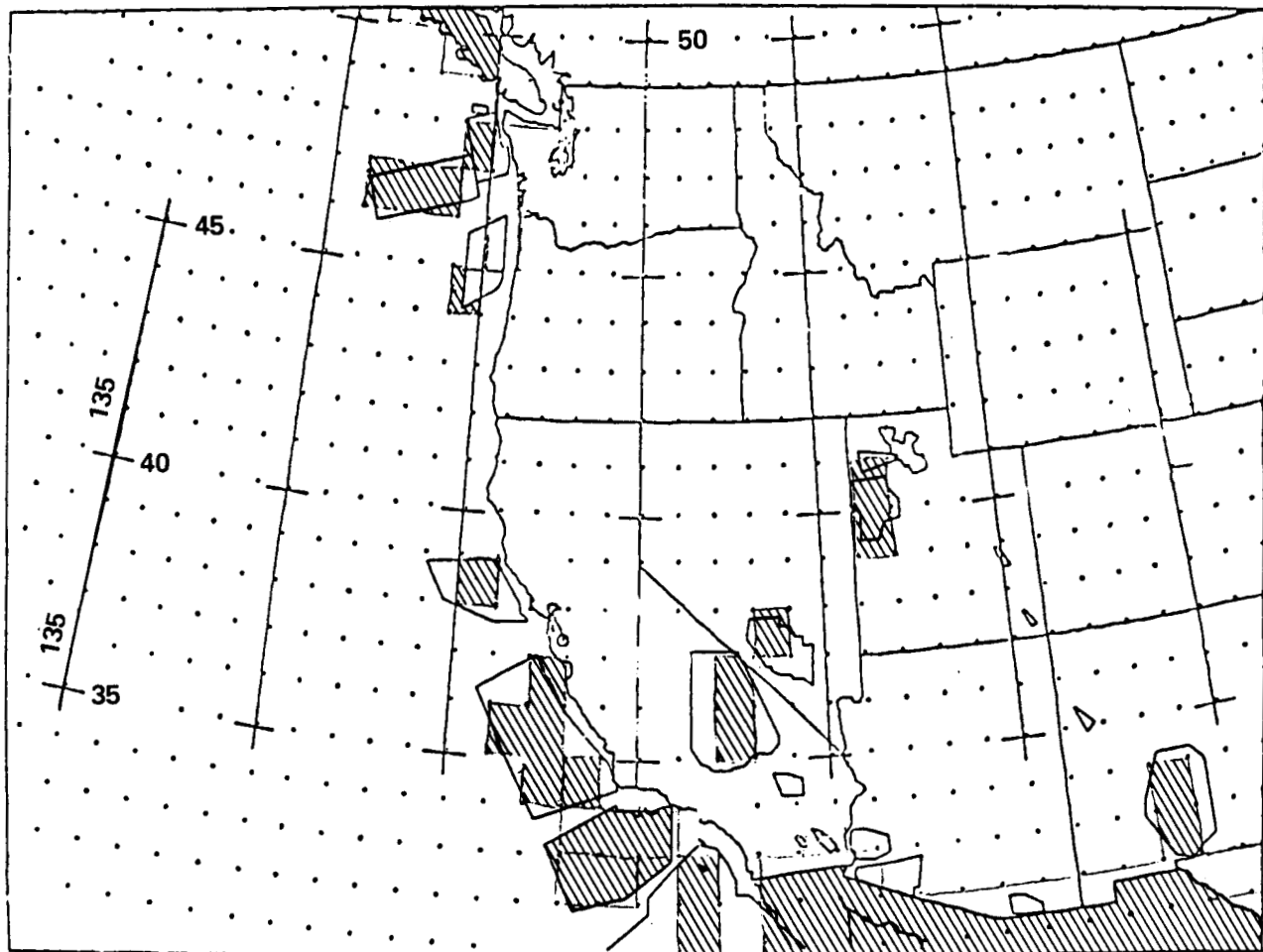


Figure 4
Representation of terrain using squares.

The second method utilized the data-base of KNAV which represented the boundaries of the zones as line segments. This data-base of zones can be represented graphically as shown in Figure 1. KAOS uses this zone information as follows. The rectangle enclosing a zone (see Fig. 5) is used as a filter; that is, if a leg does not pass through the enclosing rectangle, it could not pass through the zone. If the leg does pass through the rectangle, further computations are made to see if the leg intersects any of the straight line segments which compose the zone boundaries. The line segment method of detecting forbidden zones is more accurate but is not as fast as is the array-lookup method.

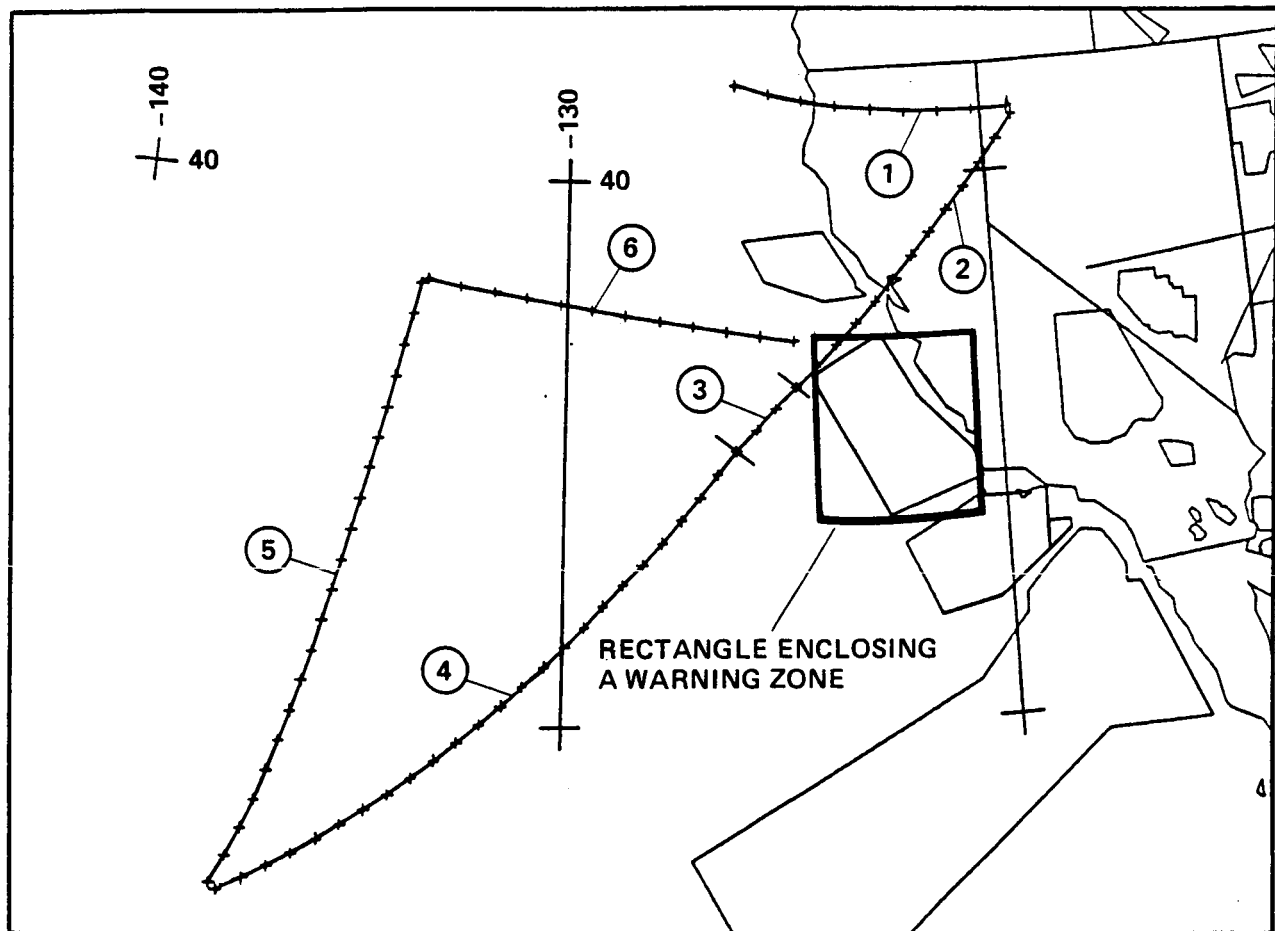


Figure 5
Representation of terrain using zone boundaries.

CONCLUSIONS AND DISCUSSIONS

A knowledge-based expert system scheduler (KAOS) was developed which offers high level performance in a complex environment. It serves as a valuable aid to flight planners. One of its significant capabilities is that it

couples symbolic reasoning with extensive numerical computations. MRS's mechanism for representing symbolic propositions as function calls is well suited to deal with quantitative models involving symbolic reasoning. KAOS has made use of this MRS feature to couple symbolic reasoning with conventional mathematical algorithms to provide a basis for generation of flight plans for airborne astronomical observations.

An interactive version of KAOS was developed to enable the user to provide guidance in the search for a flight plan. Some strategies to guide the search are described in the user's manual. Further development is required to incorporate these heuristics and more into a completely autonomous version.

KAOS has received limited acceptance from astronomers. It has aided in development of flight plans for Drs. Martin Cohen (UCB), James Houck (Cornell), and Harley Thronson (University of Wyoming). In addition, it has been used to develop flight plans for several astronomers at Ames Research Center.

KAOS is now being used for the generation of potential feasible high value flight route plans for such missions as the observation of Halley's Comet. In addition, the KAOS technology is being transferred to space observation of astronomical observations. Scheduling of observations for SIRTf (Space Infra-Red Telescope Facility) is being developed.

REFERENCES

- (1) Erickson, E.F. and Matthews, S., WINDO: A computer program for planning astronomical observations (NASA Technical Memorandum, TM X-73,156, 1976).
- (2) Gevarter, W.B., Nachtsheim, P.R., Stutz, J.C., and Banda, C.P., An expert system for the planning and scheduling of astronomical observations (IEEE International Conference on Robotics and Automation, March 1985).
- (3) Buchanan, B.G. and Shortliffe, E.H., Rule-based expert programs: the MYCIN experiments of the Stanford Heuristic Programming Project (Addison-Wesley, Reading, MA, 1984).
- (4) MIT Mathlab Group, A brief overview of MACSYMA (Laboratory for Computer Science, Cambridge, MA, 1982).
- (5) Lindsay, R., Buchanan, B.G., Feigenbaum, E.A., and Lederberg, J., Applications of artificial intelligence for chemical inference: the DENDRAL Project (McGraw-Hill, New York, 1980).

- (6) McDermott, J., R1: a rule-based configurer of computer systems (Artificial Intelligence, Vol. 19, No. 1, 1982).
- (7) Genesereth, M.R., Greiner, R., Grinberg, M.R., and Smith, D.E., The MRS dictionary (Department of Computer Science Memo HPP-80-24, Stanford University, Stanford, CA, Dec. 1980).
- (8) Banda, C.P., Stutz, J.C., Nachtsheim, P.R., and Gevarter, W.B., "User's guide for KAOS (Kuiper Astronomical Observatory Scheduler), (Informatics General Corp. Technical Note 84-7104-925-45, Nov. 1984).
- (9) Foderaro, J.K. and Sklower, K.L., The FRANZ LISP manual, (University of California, Berkeley, 1981).
- (10) Winston, P.H., Artificial intelligence (Addison-Wesley, Reading, MA, 1984).
- (11) Genesereth, M.R., Partial programs (Department of Computer Science Memo HPP84-1, Stanford University, Stanford, CA, Jan. 1984).
- (12) Finger, J.J., Residue-A deductive approach to design (Department of Computer Science Memo HPP-83-46, Stanford University, Stanford, CA, Dec. 1983).
- (13) Hill, J.C., Utegaard, T.F., and Riordan, G., Dutton's navigation and piloting (United States Naval Institute, Annapolis, MD, 1967).

1. Report No. NASA TM-88194	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle A KNOWLEDGE-BASED EXPERT SYSTEM FOR SCHEDULING OF AIRBORNE ASTRONOMICAL OBSERVATIONS		5. Report Date December 1985	
		6. Performing Organization Code A-86054	
7. Author(s) P. R. Nachtsheim and W. B. Gevarter, and J. C. Stutz and C. P. Banda (Informatics General Corporation, Sunnyvale, CA)		8. Performing Organization Report No.	
		10. Work Unit No.	
9. Performing Organization Name and Address Ames Research Center Moffett Field, CA 94035		11. Contract or Grant No.	
		13. Type of Report and Period Covered Technical Memorandum	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code 506-54-61	
15. Supplementary Notes Point of contact: P. Nachtsheim, Ames Research Center, M/S 244-7, Moffett Field, CA 94035 (415) 694-6526 or FTS 464-6526			
16. Abstract KAOS (Kuiper Airborne Observatory Scheduler) is a knowledge-based expert system developed at NASA Ames Research Center to assist in route planning of a C-141 flying astronomical observatory. This program determines a sequence of flight legs that enables sequential observations of a set of heavenly bodies derived from a list of desirable objects. The possible flight legs are constrained by problems of observability, avoiding flyovers of warning and restricted military zones, and running out of fuel. A significant contribution of the KAOS program is that it couples computational capability with a reasoning system.			
17. Key Words (Suggested by Author(s)) Expert systems Computer science Artificial intelligence		18. Distribution Statement Unlimited Subject category - 62	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 16	22. Price*