N89-19852

AN INTELLIGENT INTERFACE FOR SATELLITE OPERATIONS: YOUR ORBIT DETERMINATION ASSISTANT (YODA)

Anne Schur
GE/RCA Advanced Technology Laboratories
Route 38, Moorestown Corporate Center
Moorestown, New Jersey 08057

ABSTRACT

An intelligent interface is often characterized by the ability to adapt evaluation criteria as the environment and user goals change. Some factors that impact these adaptations are redefinition of task goals and, hence, user requirements; time criticality; and system status. To implement adaptations affected by these factors, a new set of capabilities must be incorporated into the human-computer interface design. These capabilities include: 1) dynamic update and removal of control states based on user inputs, 2) generation and removal of logical dependencies as change occurs, 3) uniform and smooth interfacing to numerous processes, databases, and expert systems, and 4) unobtrusive on-line assistance to users of varied skill levels. This paper discusses how these concepts were applied and incorporated into a human-computer interface using artificial intelligence techniques to create a prototype expert system, YODA (Your Orbit Determination Assistant). YODA is a "smart" interface that supports in real time orbit analysts who must determine the location of a satellite during the station acquisition phase of a mission. The paper also describes the integration of four knowledge sources required to support the orbit determination assistant: orbital mechanics, spacecraft specifications, characteristics of the mission support software, and orbit analyst experience. This initial effort is continuing with expansion of YODA's capabilities, including evaluation of results of the orbit determination task.

INTRODUCTION

Current satellite mission support tasks at GE utilize off-line mission support software known as SOCS (Spacecraft Orbit Control System). SOCS stores information about the satellite and contains the algorithms used for nine major analysis tasks; e.g., orbit determination and ephemeris propagation. These algorithms operate on parameter values supplied by the orbit analyst (O/A). Entries are made via a terminal, with no input prompts provided. It is the responsibility of the O/A to check all inputs and consult a set of manuals for input requirements, including appropriate format and values. Once all values have been entered, the O/A issues a command telling SOCS to receive the entered values and perform the selected analyses. When an analysis is complete, SOCS returns an output via hardcopy. The O/A must interpret the results to determine their validity.

These practices are time consuming and require the O/A to have experience in orbital mechanics, a sound knowledge of the spacecraft characteristics, and knowledge of both the SOCS software and data entry procedures. Compounding these difficulties is the trend for future satellite mission support tasks to serve an increased number of satellites of increased complexity, despite the counter trend toward satellite autonomy. As the total mission support workload increases, personnel will be in shorter supply, and therefore, often lacking in experience. It will be difficult for these personnel to respond to and perform critical tasks in a timely manner using current practices. To alleviate this situation, GE took steps to simplify ground support operations by creating an intelligent human-computer interface. The objective of this effort was to reduce both workload and required experience level needed to perform orbit determination.

An intelligent interface is often characterized by the ability to adapt evaluation criteria as the environment and user goals change. Satellite orbit determination is a reoccurring mission support task in which the criteria used to locate satellites changes with environmental changes; e.g., variations in spacecraft characteristics, orbit and mission phase. The adaptable interface developed for this task was named YODA* (Your Orbit Determination Assistant). YODA is an expert system prototype that assists orbit analysts who must determine the location of a satellite under severe time constraints, who have little realtime satellite operations experience, or who may perform the orbit determination task infrequently.

*YODA was implemented on a Texas Instrument Explorer using ART (Automation Reasoning Tool, developed by Inference). Common LISP was used to reformat the input orbit determination values into a form that was readable by the SOCS software.)

YODA has the ability to check all inputs, adapt its evaluation criteria, dynamically update and remove control states based on user inputs, and provide unobtrusive on-line assistance to users of various skill levels. The following discussion describes how these capabilities were incorporated into the human-computer interface.

USER OPERATIONAL REQUIREMENTS

For YODA to be a success it was critical for it to be accepted by the user community. Therefore, priority was given to meeting user operational requirements. Table I lists these requirements, together with the features implemented in YODA to meet the specified needs.

Collectively, these features impart intelligence to the user interface. Knowledge incorporated into the human-computer interface includes the ability to sense/make inferences about: appropriate default values, advice needed, value constraints for sanity checks, and the relationships between items as a function of the situation.

TABLE I. USER REQUIREMENTS AND ASSOCIATED YODA FEATURES

REQUIREMENT	YODA FEATURE
Perform only orbit determination tasks	• Minimizes required user knowledge of computer systems, SOCS and data input procedures • Provides default values, and limit and sanity check criteria
Easily accommodate user change of mind	• Adapts with mind change
Serve multiple users • Reduce need to seek expert advice	• Provides relevant help easily
Allocate control to user versus system	• Enables multipath access • Performs any task in any order
Provide feedback • what to do • result of input	• Assesses value of user input and procedure choice based on current situation • Meaningful input prompts, error messages, and advice based on current situation • Advice when required
Provide knowledge, "drowned in information but starved for knowledge"	• Cognitive model of tasks • Maintains relationships between information items
System expandable to on/off-line training Grow system to accommodate more spacecraft and SOCS modules	• Track where the user is in the task (future) • Standardized operations • Underlying code readable, repeatable, updatable, without requiring close support from AI expert

APPROACH

Knowledge Acquisition

Knowledge acquisition was performed using subject matter experts from two technical domains: SOC software and orbit determination, because current practices require the O/A to have knowledge of both these domains. For example, communicating a value for a particular element requires the O/A to input SOCS values indicating the particular record, the specific memory location, and the associated target element value. Table II shows the domain knowledge needed to make an entry. The information actually input to SOCS is highlighted. Figure 1 shows a completed orbit determination input with the examples provided in Table II highlighted.

To meet user requirements, a detailed task analysis was included as part of the knowledge acquisition process. This approach enabled an operational concept to be formulated that specifically met the needs of the O/A in the context of all mission support phases (see Figure 2). Additionally, the O/A would be provided with a means to create and maintain a cognitive model of the orbit determination task and would no longer be required to know SOCS software.

TABLE II. EXAMPLES OF ORBIT ANALYST KNOWLEDGE REQUIRED TO INPUT ORBIT DETERMINATION INFORMATION

DOMAIN KNOWLEDGE					
Orbit Determination		SOCS Software			
Input Type	Item	Perm-Up Data Type	SOCS Blocks	Memory Location	Value
Bounds	Element 1	PE	SREALS	354	1000000
Coordinate System	Keplerian				6
Spacecraft	SPACENET				A

Knowledge Acquisition Results

The task analysis revealed the orbit determination function to be comprised of six distinct operational tasks. Two of the tasks were perceived by the O/A to contain optional subtasks; one subtask, for example, is adding special modifications which enable effects, such as solar pressure, drag, time bias, and range bias to be considered during the orbit determination computation performed by the SOCS software. Also, the task analysis provided the framework upon which to associate all information items and establish the relationships between identified items. Each item upon which the analyst would perform some action required that the following kinds of information be associated with it:

```
► SEMIDIAMETERDEMO-1ASODO        15        0
  TAB105TAB104TAB103TAB102TAB101TAB100TAB099TAB098TAB097TAB096TAB095TAB094TAB093
  TAB092TAB091
  PE,SREALS,371,50
  PE,SREALS,370,50
  PE,SREALS,369,50
  PE,SREALS,368,1000000
  PE,SREALS,367,1000000
  PE,SREALS,366,1000000
  PE,SREALS,359,100.0
  PE,SREALS,358,100.0
  PE,SREALS,357,100.0
  PE,SREALS,356,123
  PE,SREALS,355,0.05
► PE,SREALS,354,1000000
  PE,SREALS,49,0.9
  0,2,122388,233445
► 4021,6,2345565543,0.72988194
  0,456,567,458
  0,3,0,0
  ,,,
  ,,,
  ,,,
  ,,,
  ,,,
  ,0,2,2
  end
```


Figure 1. Currently Used Orbit Determination Input Display

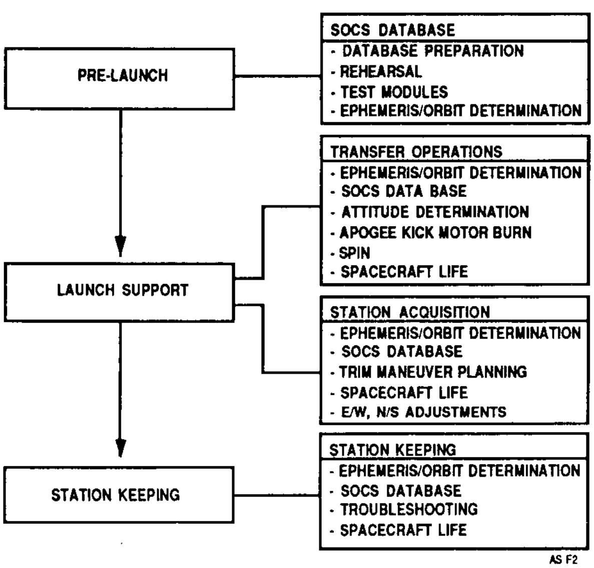

Figure 2. Mission Support Phases for Satellite Operations

- criteria to access information
- criteria to remove or exchange information
- input messages
- error messages
- general help (task level; e.g., initial guess, using YODA)
- specific help (item level; e.g., coordinate system selection)
- evaluation criteria in respect to the set-up environment
- where to transfer information, if applicable
- SOCS values.

Figure 3 is a simplified diagram of these relationships. Tasks are indicated as rectangles, with the information associated with each task and the relationships between each information item depicted hierarchically. Thus, if an information item occurring above another is removed, then the information items below it will also be removed and/or exchanged for appropriate new information.

SOFTWARE DESCRIPTION

Overall System

Figure 4 shows a system function flow. The O/A enters values, which YODA checks for validity. When all inputs have been made, the analyst indicates that they may be sent to SOCS for orbit determination computation. This action calls a routine which converts all the values specified by the analyst into a form that can be read by the SOCS software (Figure 1). The SOCS software then computes the location of the satellite and provides the results in hardcopy. It is up to the analyst to validate the orbit determination results. (An expert system to perform this latter task is now being developed. The outputs of this expert system will be fed into YODA in the form of recommendations to improve the orbit determination result.)

Software Architecture

The interface architecture is composed of objects and rules. The objects are represented in two separate knowledge-bases: a task knowledge-base and a value knowledge-base. The task knowledge-base is comprised of those items (constants) which will always be part of the task despite user initiated (rule activated) changes to those objects: i.e., their values. This knowl-

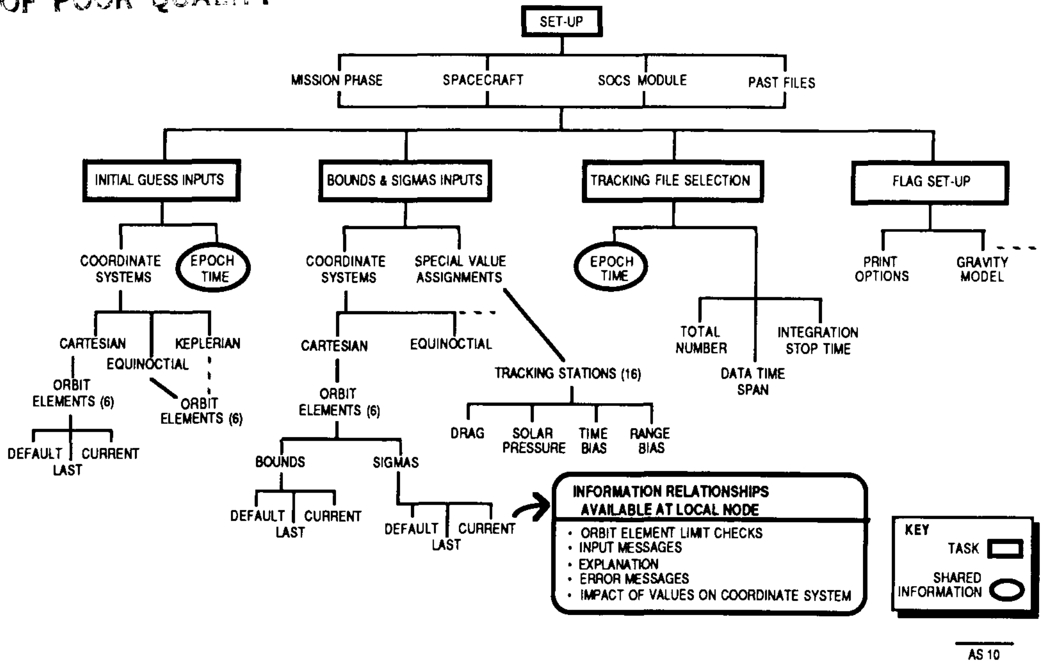

Figure 3. Simplified Diagram Showing Information Relationships Between Individual Items and Orbit Determination Tasks

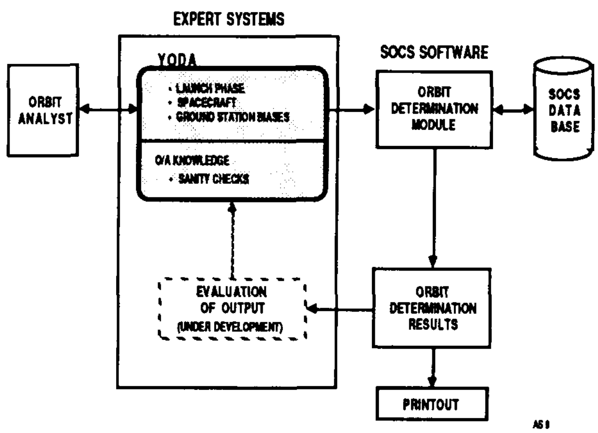

Figure 4. System Level Function Flow

edge-base plays a critical role in achieving the criterion of providing a cognitive model of the orbit determination process to the analyst. Figure 5 shows an example from the analyst's perspective. The display/window is represented as an object and corresponds to a specific orbit determination task. The icons/objects within the window represent the items needed by the analyst to perform the task. Figure 3 represents the way in which the task knowledge-base is structured. The value knowledge-base contains the values which the objects in the constant knowledge-base can adopt, and the evaluation criteria used to assess the validity of any values input made by the analyst (see Figure 6). Specific input messages, error messages, and recommendations are also housed in this knowledge-base. A third knowledge-base contains the general help facilities. Table III compares, in summary

TABLE III. COMPARED OBJECT CHARACTERISTICS OF THE TASK AND VALUE KNOWLEDGE-BASES

TASK KNOWLEDGE-BASE OBJECTS	VALUE KNOWLEDGE-BASE OBJECTS
• Directly manipulatable by analyst, empty input slots	• Analyst unable to manipulate these objects directly
• Associated SOCS function values	—
• Control state identifiers	—
• Links to associated defaults, evaluation criteria, and help information housed within other knowledge-base	• Corresponding links for default and evaluation criteria
—	• Default values, evaluation criteria, input and error messages, all of which correspond to appropriate mission phase, spacecraft, coordinate system, etc.

form, the differences between the objects of the task knowledge-base and the value knowledge-base.

The rules embody the knowledge of the interface. Each rule refers to the state of some relevant knowledge-base. In accordance with a change of state initiated by the O/A the rule can adapt the interface to the appropriate context. For example, using an extreme case, if an analyst changes his/her mind about which spacecraft upon which to perform an orbit determination, YODA can update and/or remove appropriate control

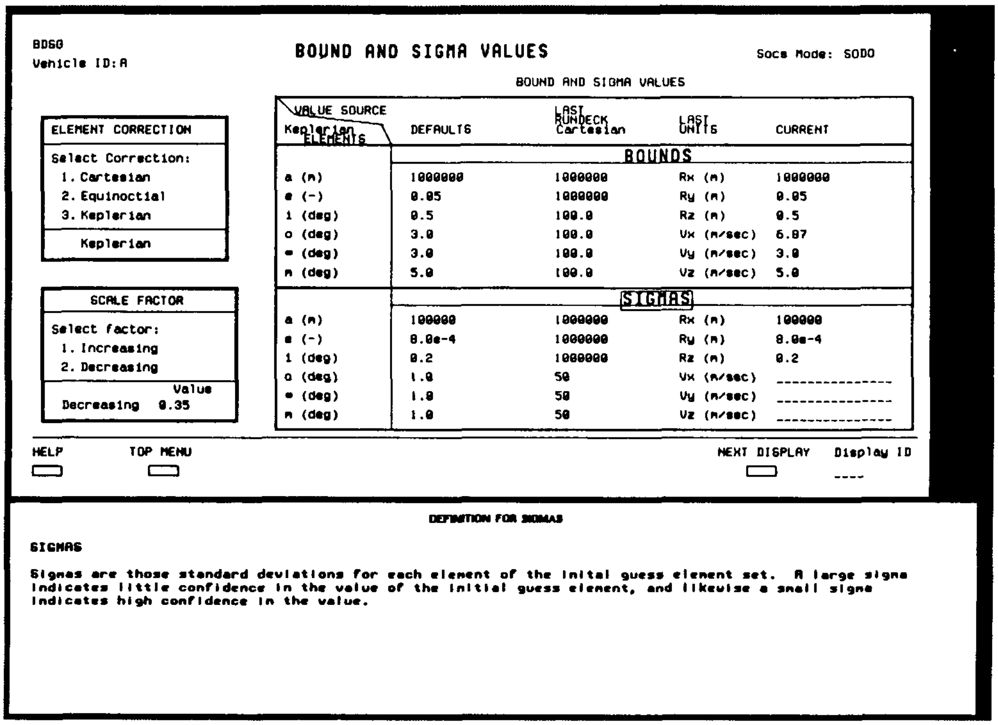

Figure 5. Graphical Input Display Used to Support Bound and Sigma Value Entry Task

```
(DEFSCHEMA KEPLERIAN-BOUND-DEFAULTS-FOR SPACENET
             "default values for Keplerian bounds of SPACENET"
  (vehicle-id A)
  (use correction-type)
  (class bounds)
  (coordinate-system keplerian)
  (element  (el1  1000000))
  (element  (el2  1.05))
  (element  (el3  0.5))
  (element  (el4  3.0))
  (element  (el5  3.0))
  (element  (el6  5.0)))
```

Figure 6. Example of an Object in the Knowledge-Base

states and generate and/or remove logical dependencies as the change occurs. All element values (defaults, last, current) associated with the 'old' vehicle will be removed from the fact base. Default and last file values associated with the 'new' spacecraft in respect to the already specified coordinate system, and the constraints used for last file selection (mission phase, SOCS software module) will be displayed. The analyst will be expected to input new current values. At a local level, changing a coordinate system will result in the O/A seeing an exchange of default values and the removal of current values. Evaluation criteria and user advice will also be appropriately removed and/or exchanged, but in a manner that is transparent to the analyst. Figure 7 shows the information that is presented to the analyst when asking about the Cartesian coordinate system. To obtain this information the analyst pointed to the word "Cartesian". The rules also maintain the appropriate values required to execute SOCS functions. This bookkeeping task is also transparent to the user.

User Interaction with YODA

The dialogue between the analyst and YODA consists of direct manipulation. This allows the analyst to control the sequence of events and therefore have the freedom to take any action in any order. An expert could fill in orbit determination values working within a single display (the summary display). A less experienced analyst can walk through each task in a top down sequence, while a more experienced analyst can perform the tasks nonsequentially. At all times the analyst is provided with commands that are relevant to the task. To enable the value knowledge-base to be easily updated, schemata were written in English (Figure 6). To change the values, the current values have to be deleted and new ones put in their place. The person performing this update would not have to modify the rule base.

SUMMARY

YODA is a knowledge entry program developed to allow orbit analysts to efficiently use an external data source. This is accomplished by enabling orbit determination evaluation criteria to be correctly specified by an O/A so that orbit determination computations can be executed by a software system. The O/A works within a visual programming environment. Graphical forms allow specification of knowledge using a "fill in the blanks" approach. For example, orbit element values are represented in a schema language as simple icons. Whenever one of these icons is selected with a mouse, a special environment is entered, providing inferential knowledge needed to evaluate or provide advice about the parameter value in respect to the situation. The O/A is at all times in control of the input process. This is achieved by interactive displays that support the analyst's cognitive model of the task in a natural idiom. Feedback to the O/A is immediate and is presented in a manner which conveys the actions taken by the system on the inputs supplied by the analyst.

The developed prototype was successfully demonstrated. It is now being ported into a delivery system to enable YODA's use in GE's mission control room during daily operations and satellite launches.

FUTURE

The current system is limited to creating inputs for entry into the SOCS software. The output from the orbit determination computation is evaluated by the analyst. An effort is currently under way to automate this task, using an expert system that will inform the O/A of the results of the output evaluation and recommend changes that could be made on the initial inputs. It is also planned to expand this system to an online job-performance aid, where the effect of the orbit parameter values input by the O/A can be seen dynamically and in comparison to both the known defaults and last parameter value sets for the specified mission phase, spacecraft, and coordinate system.

REFERENCES

ASC Off-line Software User's Guide, RCA Corporation, Astro-Electronics Division, Princeton, NJ, Document Number: UG-OV8-2606915, 1986.

Bayman, P., and Mayer, R.E., "Instructional Manipulation of Users' Mental Model for Electronic Calculators," **International Journal of Man-Machine Studies**, 20, 189-199, 1984.

Chandrasekaran, B., "Expert Systems: Matching Techniques to Tasks," **Artificial Intelligence in Business**, Editor Reitman, W. Norwood NJ:Ablex, 1984.

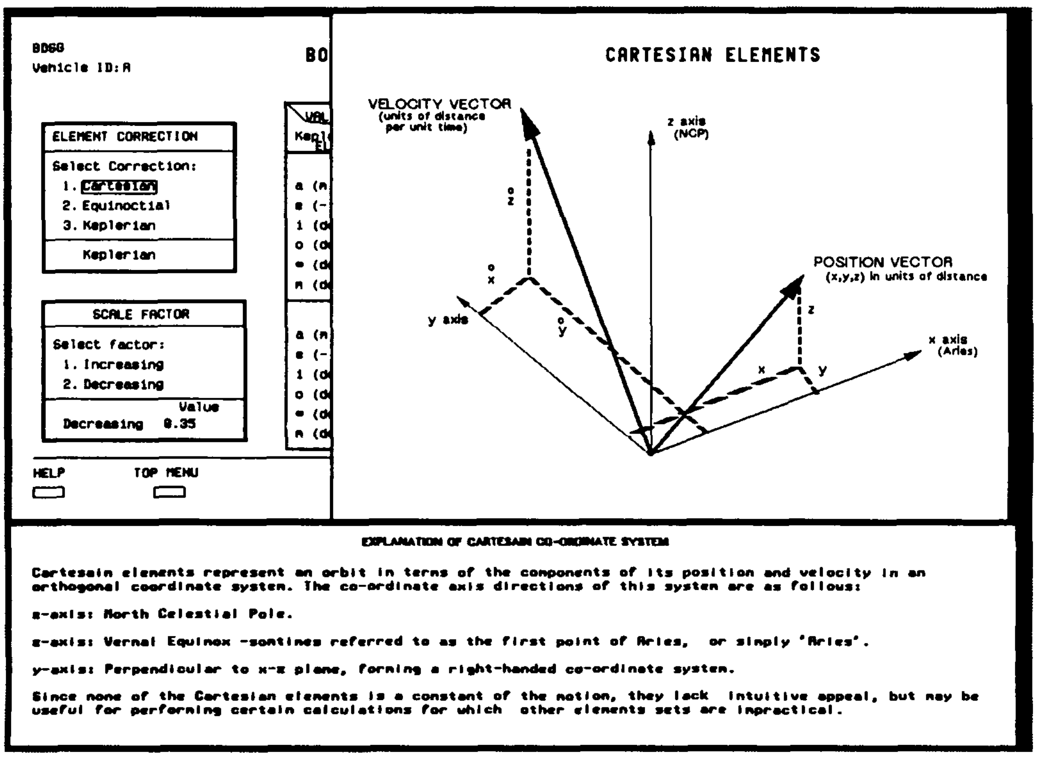

Figure 7. Explanation Display Provided When Requested by Analyst Pointing to "Cartesian"

Charniak, E.C., "Toward a Model of Children's Story Comprehension," MIT AI Laboratory, TR-266, Cambridge, MA, 1972.

Hendler, J.A., (Ed), *Expert Systems: The User Interface*, Norwood NJ:Ablex, 1987.

Norman, D.A., *Cognitive Engineering. In User Centered Design*, Editors: Norman, D.A., and Draper, S.W., Hillsdale NJ:Erlbaum, 31-65, 1986.

Rasmussen, J., *Information Processing and Human Machine Interaction*, Elsevier Science Publishing Co, North-Holland Series, 1986.

Shneiderman, B., "Direct Manipulation: A Step Beyond Programming Languages," *Computer*, 16 (89), 57-69, 1983.

SOCS Database Manual, RCA Corporation, Astro-Electronics Division, Princeton, NJ, Document Number: IS-AV7-2606915, 1987.

Swigger, K. M., "An Intelligent Tutoring System for Generation of Ground Tracks," **Proceedings of the American Association for Artificial Intelligence**, 72-76, 1987.